بسم الله الرحمن الرحيم

جامعة السودان المفتوحة

تقنيات الإنترنت(2)

رمز المقرر ورقمه: حسب 2038

إعداد المادة العلمية:أ.د.السيد محمود الربيعي

د. إبراهيم فتحي معوض

التحكيم العلمي : د. أحمد صلاح الدين عبد الله
التصميم التعليمي : إيهاب عبد الحي
التدقيق اللغوي : أ. الهِدَيْ عبدالله محمد
التصميم الفني : أماني الأمين مبارك

منشورات جامعة السودان المفتوحة، الطبعة الأولي2007
جميع الحقوق محفوظة لجامعة السودان المفتوحة، لا يجوز إعادة إنتاج أي جزء من هذا المقرر، وبأي وجه من الوجوه، إلا بعد الموافقة

مقدمة المقرر

الحمد لله بجميع محامده كلها ماعلمت منها وما لم أعلم، عدد خلقه كلهم ماعلمت منهم وما لم أعلم، والصلاة والسلام علي السلطان الأعظم والنائب العام الاكرم، باب الرحمة المفتوح عند إنغلاق جميع الأبواب في وجوه المذنبين، وقبس الرجاء الملموح عند سيطرة اليأس في قلوب المعذبين، الباسط يد الإحسان لكل تآئب ليأخذ بيمينه إلى الصف الأول، والمستغفر للغافلين من أمته لينقذهم بجاهه من الهبوط إلى الدرك الأسفل. فطوبي لكل منتمٍ إليه. وياقرةعين المكثرين من الصلاة والسلام عليه

عزيزى الدارس، مرحبا بك إلى هذا المقرر.

ولقد اهتم المطورون بتطوير وتحديث صفحات الإنترنت وإضافة كل جديد لها، حيث ظهر العديد من لغات البرمجة التي يتم دمجها في مواقع وصفحات الإنترنت لتضيف تحكما وتفاعلا أفضل للمستخدمين. وقد أدى احتدام المنافسة بين الشركات إلى ظهور العديد من أدوات تطوير الإنترنت ولغات البرمجة.

ويهدف هذا الكتاب إلى تعليم الطالب كيفية تصميم وتطوير صفحات الويب بنوعيها باستخدام حزم البرامج الجاهزة، وكذلك بدراسة الأنواع المختلفة للغات برمجة الويب الفعالة لمعرفة الفروق الجوهرية بين هذه اللغات. وفيه يتم التركيز على شرح ووصف تعليمات ودوال لغة الجافا سكربت (JavaScript) ومجال استخدامها، ومميزات هذه اللغة وعيوبها مع شرح أمثلة تطبيقية لها.

كما سيتعرض الكتاب لشرح تفصيلي لتعليمات ومكتبة لغة VBScript التي تمثل اللغة الافتراضية لتطوير تطبيقات تعمل لدى خادم الويب (Web Server) ، ثم يتعرض الكتاب لتقنية صفحات الخادم الفعالة (ASP) (Active Server Pages) وكذلك إلى التعامل مع قواعد البيانات باستخدام (ADO) التي تمكن المبرمج من تطوير المواقع الفعالة (Dynamic Web Sites)، أو ما تسمى بتطبيقات الإنترنت (Web-based Applications) مع شرح أمثلة تطبيقية لها.

محتويات المقرر

الوحدة	رقم الصفحة
المقدمة	أ
الوحدة الأولى: المفاهيم الرئيسية لإطار عمل تشغيل صفحات الويب	4
الوحدة الثانية: تطوير مواقع ثابتة المحتوى Microsoft Front Page	29
الوحدة الثالثة: البرمجة لدى العميل (Client-Side Programming) الجزء الأول	71
الوحدة الرابعة: البرمجة لدى العميل (Client-Side Programming): الجزء الثاني	95
الوحدة الخامسة: البرمجة لدى الخادم لغة VBScript	131
الوحدة السادسة: البرمجة لدى الخادم	173
الوحدة السابعة: التعامل مع قواعد البيانات	215


الوحدة الأولى
المفاهيم الرئيسية لإطار عمل
تشغيل صفحات الويب

محتويات الوحدة

الموضوع	رقم الصفحة
المقدمة	4
تمهيد	4
أهداف الوحدة	5
1. الأنواع الرئيسية لمواقع الإنترنت	6
1.1 المواقع ثابتة المحتوى Static Web Sites	6
1.1.1 الإنترنت ثم إعادة نشرها مرة ثانية على خادم الويب	8
2.1.1 عيوب المواقع ثابتة المحتوى	8
2.1 المواقع متغيرة المحتوى Dynamic Web Sites	10
3.2 تلافى عيوب المواقع الثابتة المحتوى	11
2. التصور الكامل لعناصر إطار عمل تشغيل صفحات للويب	13
1.2 عميل الويب Web Client	14
3.2 خادم الويب Web Server	16
3.3 بروتوكول HTTP	17
3.4 قواعد البيانات (Databases)	18
3.5 لغة HTML	18
3.6 مكتبة كائنات ASP	19
3.7 لغة JavaScript	20
3.8 لغة VBScript	20
3.9 المعالجة لدى العميل Client-Side Processing	20
3.10 المعالجة لدى الخادم Server-Side Processing	21
3.11 مكتبة DHTML	21

الخلاصة	23
لمحة مسبقة	23
إجابات التدريبات	24
مسرد المصطلحات	26
المراجع	27

المقدمة

تمهيد

عزيزي الدارس، مرحبا بك في الوحدة الأولى من مقرر "تقنيات وبرمجة الإنترنت"
بما أن الهدف الرئيس لهذا الكتاب هو تطوير تطبيقات الويب التي نطلق عليها اسم مواقع متغيرة المحتوى (Dynamic Web Sites)، هذه الوحدة تقدم أولاً شرحاً مفصلاً عن الأنواع الرئيسية لمواقع الإنترنت ومكوناتها، حيث تنقسم مواقع الإنترنت طبقا لأسلوب إنشاء محتواها إلى مواقع ثابتة المحتوى (Static Web Sites)، ومواقع متغيرة المحتوى (Dynamic Web Sites)، ثم تعرض مميزات وعيوب كل منهما بشئ من التفصيل.

كما فضلنا أن نضع تصوراً كاملاً وواضحاً لمكونات إطار عمل تشغيل صفحات الويب لكي نعطى القارئ فكرة مختصرة عن مكونات هذا الإطار، حيث تكمن صعوبة تطوير المواقع متغيرة المحتوى (Dynamic Web Sites) في وجود العديد من المفاهيم والتقنيات التي تحتاج إلى توضيح، ومن ثم رأينا أن نقدم في هذه الوحدة شرحاً وافياً عن عميل الويب (Web Client)، و خادم الويب (Web Server)، وبروتوكول HTTP، ولغة HTML، وكل من مكتبة كائنات ASP، ومكتبة DHTML وكل من لغة JavaScript، ولغة VBScript، والمعالجة لدى العميل (Client-Side Processing)، والمعالجة لدى الخادم (Server-Side Processing).

عزيزي الدارس، تكمن أهمية هذه الوحدة في كونها وحدة ابتدائية تأسيسية تنبني عليها بقية الوحدات، ولذا لابد أن نوليها اهتماماً خاصاً وذلك بإجابة التدريبات المصاحبة لها، وأسئلة التقويم الذاتي الواردة في ثناياها لانها ستساعدك في فهم ماورد فيها من مفاهيم.

أهلا بك مرة أخرى إلى هذه الوحدة عسى أن تنتفع بها وأن تفيد منها، وأن تساعدنا في نقدها وتطويرها.

أهداف الوحدة



عزيزي الدارس، بعد فراغك من دراسة هذه الوحدة ينبغي أن تكون قادراً على أن:

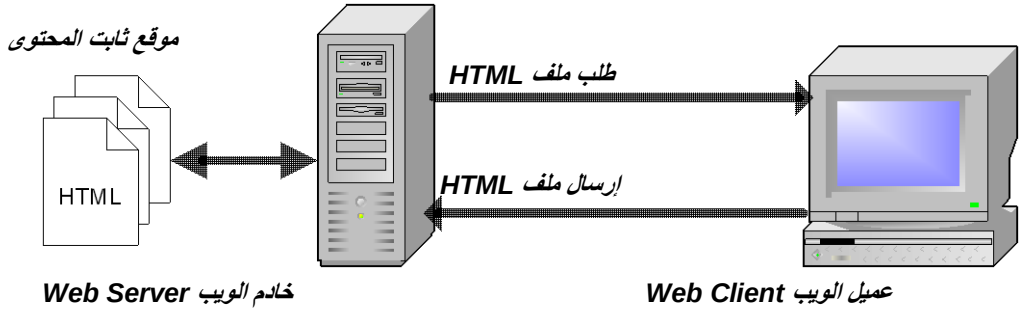
- تشرح أنواع مواقع الإنترنت.
- تعرف وظائف أنواع مواقع الويب.
- تشرح إطار عمل الويب.
- تعرف الصعوبات التي تواجه تطوير الويب.
- تبين وظائف المكتبات المختلفة للويب.
- تعدد أنواع البرمجة المستخدمة في الويب .

1. الأنواع الرئيسية لمواقع الإنترنت

عزيزي الدارس، تنقسم مواقع الإنترنت (Web Sites) إلى نوعين رئيسين: مواقع ثابتة المحتوى (Static Web Sites) ومواقع متغيرة المحتوى (Dynamic Web Sites) تتكون المواقع ثابتة المحتوى من العديد من ملفات HTML فقط (بالإضافة إلى ملفات أخرى مثل ملفات صور، صوت... الخ) حيث تحتوي هذه الملفات على لغة HTML التي تفسر بواسطة متصفح الإنترنت (Internet Explorer) أو ما يسمى بعميل الويب (Web Client) لكي تظهر أمام المستخدم بشكل جيد ومنسق، أما المواقع متغيرة المحتوى فتتكون من نوع آخر من أنواع الملفات (مثل ملفات ASP أو ملفات JSP) التي يتم معالجتها أولاً على خادم الويب (Web Server) عند طلبها من عميل الويب (Web Client)، وإرسال ناتج المعالجة وسنتحدث بشيء من التفصيل عن كل من هذين النوعين.

1.1 المواقع ثابتة المحتوى Static Web Sites

ولقد أطلق على هذه المواقع اسم المواقع ثابتة المحتوى (Static Web Sites) لأن محتوياتها لا تتغير مع الوقت، بمعنى إذا تصفحنا هذه المواقع آلاف المرات ستظهر لنا بنفس المحتويات دون تغيير، حيث تتكون هذه المواقع من ملفات تحتوي على شفرة HTML، وهي اللغة المسؤولة عن ظهور العناصر (مثل النصوص، الجداول، الصور... إلخ) داخل صفحات الويب بشكل منسق ومحدد، ولذلك عندما يتم استدعاء أحد هذه الملفات من خادم الويب وذلك بالضغط على رابطة ما (Hyperlink) مثلا، يتم إرسال هذا الملف كما هو دون تغيير في محتوياته من خادم الويب (Web Server) إلى عميل الويب (Web Client) يوضح الشكل رقم 1-1 أحد المواقع الثابتة وسيناريو تفاعل عميل الويب مع خادم الويب لاسترجاع أحد ملفات HTML.



شكل 1-1 : تفاعل عميل الويب مع خادم الويب لاسترجاع ملفات المواقع ثابتة المحتوى

ويمكن تشبيه المواقع ثابتة المحتوى بالكتاب (أو المجلة) الذي إذا قرأناه عدداً من المرات سنجد محتوياته من نصوص وصور ثابتة لا تتغير إلا إذا تم إصدار نسخة جديدة منه (أي إلا إذا تم تعديله وطباعته ونشره مرة ثانية)، ويوضح الشكل رقم (2-1) أحد أمثلة صفحات HTML التي تحتوي على نصوص وصور وعناصر متحركة.



شكل 2- 1 أحد أمثلة صفحات HTML

لاحظ أن ثبات المحتوى لا يعني عدم احتواء هذه المواقع على عناصر متحركة مثل لقطات الفيديو (Video Clips) أو أفلام الفلاش (Flash Movies) وغيرهما، حيث تحتوي هذه المواقع على عناصر عديدة مثل النص، الصور الثابتة (Static Images)، الصور المتحركة (Animated Images)، وملفات الصوت (Audio Files)... إلخ، ولكن تذكر أن هذه العناصر تظهر في كل مرة للمستخدم كما ظهرت له أول مرة دون تغيير إلا إذا تم إعادة تصميم محتوياتها بواسطة مصممي المواقع.

1.1.1 الإنترنت ثم إعادة نشرها مرة ثانية على خادم الويب

مميزات المواقع ثابتة المحتوى

تتميز هذه المواقع بسهولة تصميمها وقصر مدة التطوير اللازمة لتطويرها حيث إن المطلوب من مصممي هذه المواقع أن يتعلم فقط أحد أدوات تطوير مواقع الإنترنت مثل برنامج Microsoft Front Page الذي يمكن المصمم من تطوير مواقع ثابتة المحتوى بسهولة ويسر وفى خلال فترة زمنية قصيرة، ولذلك نجد أن عدد مطوري هذه المواقع كثيرون وتجد أيضاً أن تكلفة إنشاء مثل هذه المواقع رخيص لأن الأمر يتطلب تعلم أحد برامج تطوير مواقع الإنترنت فقط (أي لا يتطلب تطوير برامج).

أسئلة تقويم ذاتي



1.عرف المواقع ثابتة المحتوى (Static Web Sites).
2. ما مميزات وعيوب المواقع ثابتة المحتوى(Static Web Sites)؟

2.1.1 عيوب المواقع ثابتة المحتوى

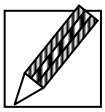
أما عن عيوب هذه المواقع، فإنها عديدة، حيث إن اسم هذه المواقع (مواقع ثابتة المحتوى) يعكس ضعفها، فعلى سبيل المثال بفرض أن أحد المكتبات التجارية أرادت أن تنشئ موقعاً لها على الإنترنت لكي تعرض محتويات هذه المكتبة من كتب ومجلات، إلخ، فإذا قرر الشخص المسؤول تصميم موقع ثابت المحتوى – وذلك لقلة تكلفة تطويره– فإنه

سوف يواجه مشاكل إضافة وصيانة محتوى هذا الموقع بعد نشره لأول مرة، ومن أمثلة ذلك:

- إذا أراد مسؤول المكتبة أن يحذف بعض الكتب من الموقع لسبب ما (عدم توافرها لديهم)، فإنه يجب أن يتم إلغاء بيانات هذه الكتب يدوياً من ملفات HTML التي تحتوى على الكتب المراد حذفها، ثم إعادة نشرها على الإنترنت مرة ثانية.
- إذا أراد مسؤول المكتبة إضافة كتب جديدة إلى الموقع، فإنه يجب أن يتم إضافة بيانات هذه الكتب يدوياً إلى ملفات HTML موجودة (أو إنشاء ملفات HTML جديدة لهذا الغرض) ثم إعادة نشرها على الإنترنت.
- إذا أراد مسؤول المكتبة تغيير بيانات بعض الكتب (مثل سعر البيع أو الكمية المتوفرة)، فإنه يجب أن يتم تعديل بيانات هذه الكتب يدوياً داخل ملفات HTML التي تحتوي على الكتب المراد تعديل بياناتها، ثم إعادة نشرها على الإنترنت.

باختصار فإنه يجب على مسؤول المكتبة تعيين فريق عمل من مطوري مواقع الإنترنت ثابتة المحتوى لتعديل وصيانة محتويات هذا الموقع، ولذلك سيصبح ثمن صيانة الموقع ثابت المحتوى يساوي أضعاف تكلفة تطوير موقع متغير المحتوى.

تدريب (1)



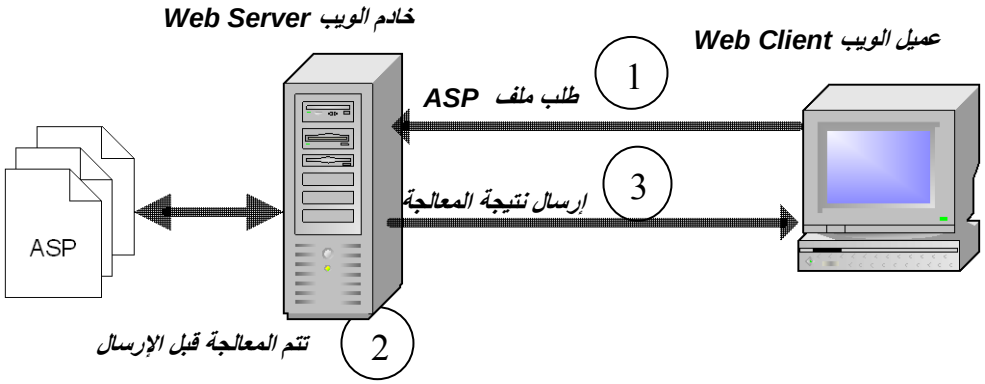
1. وضح بالرسم خطوات سيناريو تفاعل عميل الويب مع خادم الويب لاسترجاع محتويات المواقع ثابتة المحتوى.

2.1 المواقع متغيرة المحتوى Dynamic Web Sites

عزيزي الدارس، على عكس المواقع ثابتة المحتوى (Static Web Sites)، فإن المواقع متغيرة المحتوى (Dynamic Web Sites) تتكون من نوع آخر من أنواع الملفات التي تحتوي على شفرة ASP (Active Server Pages) التي يتم معالجتها لدى خادم الويب (Web Server) قبل أن ترسل إلى عميل الويب (Web Client)، حيث يتم إرسال ناتج التنفيذ فقط وهو عبارة عن نصوص أو شفرة HTML التي تظهر للمستخدم وكأن مصدرها هو ملف HTML وليس ملف ASP. يوضح شكل رقم 3-1 خطوات سيناريو تفاعل عميل الويب مع خادم الويب لاسترجاع محتويات المواقع متغيرة المحتوى، ويتم تنفيذ هذا السيناريو على ثلاث خطوات:

- يطلب عميل الويب من خادم الويب إرسال ملف ASP معين.
- يعالج خادم الويب شفرة ASP للملف المطلوب قبل إرسال ناتج التنفيذ.

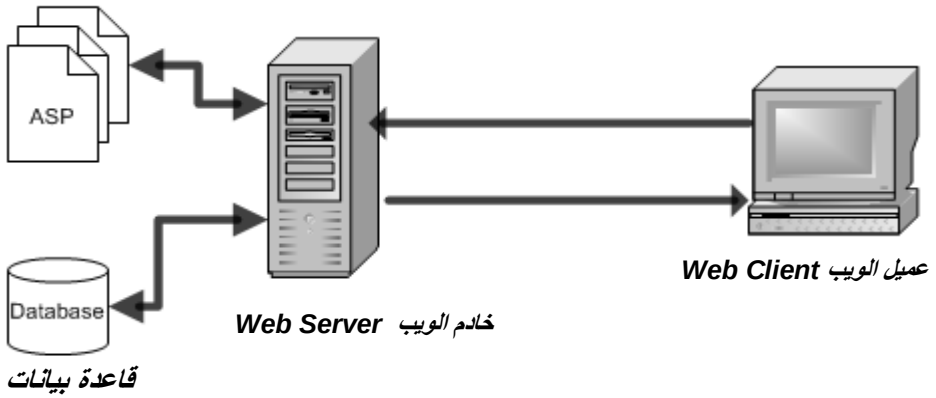
1) إرسال نتيجة المعالجة إلى عميل الويب.



شكل 3-1 : سيناريو تفاعل عميل الويب مع خادم الويب لاسترجاع محتويات المواقع متغيرة المحتوى

ولكن كيف تتغير محتويات ملفات الموقع؟ كما ذكرنا سابقاً فإن شفرة ASP المطورة داخل ملفات ASP يتم معالجتها بواسطة خادم الويب قبل إرسالها إلى عميل الويب حيث يمكن اعتبار كل ملف من النوع ASP برنامجاً فرعياً مستقلاً يمكن أن يستقبل بيانات ويعالج هذه البيانات على خادم الويب، ثم يرسل النتائج إلى عميل الويب، ومن ثم يمكن تخزين البيانات المراد أن تظهر للمستخدم بشكل ديناميكي داخل قاعدة البيانات المتصلة بخادم الويب.

(مصدر بيانات صفحات الموقع) كما هو واضح في شكل رقم 4-1.



شكل 4-1 : اتصال خادم الويب بقاعدة بيانات المواقع متغيرة المحتوى

3.1 تلافى عيوب المواقع الثابتة المحتوى

بالرجوع إلى المثال السابق (تصميم موقع ثابت المحتوى لمكتبة)، لاحظنا وجود العديد من المشاكل التي تواجه مسؤول المكتبة، التي تؤدي إلى زيادة تكلفة صيانة الموقع، ولكي نتغلب على هذه المشاكل (وإضافة مميزات عديدة أخرى سيتم ذكرها لاحقا) يجب تصميم موقع متغير المحتوى (Dynamic Web Site) متصلاً بقاعدة البيانات التي ستخزن بيانات منتجات المكتبة (الكتب، المؤلفين، الأسعار... إلخ)، ومن الملاحظ

أن قاعدة البيانات تمثل مصدر بيانات الموقع، وبالتالي إذا تم إضافة أو تعديل بيانات هذه القاعدة، سيتغير محتوى الموقع بناء على هذا التعديل.

يوضح الشكل رقم 4-1 مكونات بيئة تشغيل مواقع متغيرة المحتوى، حيث يتكون الموقع من مجموعة ملفات ASP (بالإضافة إلى ملفات مساعدة أخرى مثل ملفات الصور، فيديو، صوت... إلخ) والتي يمكن تقسيمها إلى قسمين بناء على نوع المستخدم:

1- ملفات موقع المستخدم النهائي

تستدعى هذه الملفات بواسطة المستخدم النهائي للاستعلام عن معلومات من قاعدة البيانات مثل بيانات الكتب (النوع، الاسم، السعر... الخ) بغرض الإطلاع عليها، كما يمكن للمستخدم أيضا أن يقوم ببعض المهام الأخرى مثل تسجيل/تعديل بياناته أو إضافة تعليقاته.

2- ملفات موظفي المكتبة (صيانة قاعدة البيانات)

تُسْتَدْعَى هذه الملفات بواسطة أحد موظفي المكتبة فقط بعد إدخال اسم مستخدم وكلمة مرور صحيحة، حيث لا يحق لأحد غير موظفي المكتبة استدعاء هذه الملفات، حيث تمكن هذه الملفات موظفي المكتبة من صيانة بيانات قاعدة البيانات من حذف أو إضافة، أو تعديل... إلخ، كما يمكن لموظف المكتبة الإطلاع على بيانات المستخدم النهائي، واسترجاع آرائه، وهكذا.

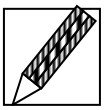
عزيزي الدارس، يجب أن تلاحظ أنه من خلال الاتصال الوثيق بين هذين النوعين من ملفات ASP: ملفات تعدل البيانات بواسطة موظفي المكتبة، وملفات تظهر البيانات من قاعدة البيانات بعد أحدث تعديل، تصبح محتويات هذا الموقع متغيرة دائما بناء على متطلبات الموقع، وهذا يحل جميع المشاكل التي تم ذكرها آنفا عند تصميم المواقع ثابتة المحتوى.

أسئلة تقويم ذاتي



1) عرف المواقع المتغيرة المحتوى(Dynamic Web Sites).
2) ما مميزات المواقع متغيرة المحتوى؟

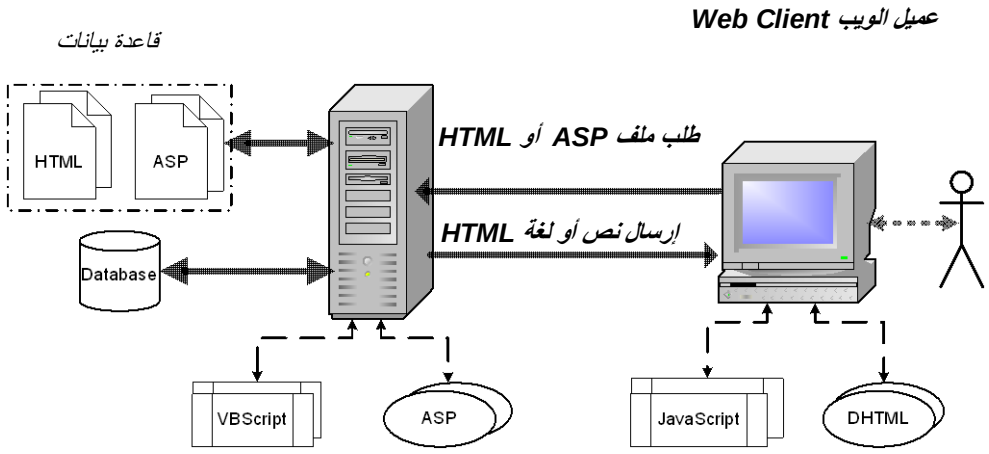
تدريب (2)



اشرح (مستخدما الرسم) فكرة تحقق ديناميكية المواقع متغيرة المحتوى، وذلك بالاستعانة بقاعدة بيانات متصلة بخادم الويب.

2. التصور الكامل لعناصر إطار عمل تشغيل صفحات الويب

عزيزي الدارس، بعد فترة من التفكير الدقيق وجدنا أنه من المفضل أن نضع تصوراً كاملاً وواضحاً لمكونات إطار عمل تشغيل صفحات الويب لكي نعطى القارئ فكرة مختصرة عن مكونات هذا الإطار، حيث إن مكمن صعوبة تطوير المواقع متغيرة المحتوى يأتي من وجود العديد من المفاهيم والتقنيات التي ربما تربك مطوري هذه المواقع. يوضح شكل رقم 5-1 مكونات إطار عمل صفحات الويب التي يجب على القارئ أن يفهم جيداً هدف كل مكون. وفيما يلي سيتم إعطاء نبذه مختصرة عن كل مكون من مكونات هذا الإطار، ثم يتم شرح هذه المكونات بالتفصيل داخل صفحات هذا المقرر.



شكل 5-1 : التصور الكامل لعناصر إطار عمل تشغيل صفحات الويب

1.2 عميل الويب Web Client

عزيزي الدارس، عندما تقرر أن تتصفح صفحات الإنترنت وتستسقي معلومات هامة من مواقع مختلفة أو تستخدم أحد مواقع محركات البحث، فأنت تستخدم برنامجاً يسمى عميل الويب (Web Client) أو متصفح الإنترنت (Internet Browser)، والذي لديه المقدرة على أن يتصل ببرنامج آخر يسمى خادم الويب (Web Server)، ويتحاور معه من خلال بروتوكول يسمى HTTP، ولذلك أحيانا نطلق على برنامج عميل الويب اسم عميل HTTP (HTTP Client) نسبة إلى بروتوكول HTTP ، ومن أشهر أمثلة هذه البرامج برنامج Internet Explorer الذي طورته شركة مايكروسوفت حيث يمكنك الحصول على أحدث إصدارات هذا البرنامج من الموقع www.microsoft.com/ie .

يجب أن يتوفر في برنامج عميل الويب مجموعة من القدرات التي تمكنه من تنفيذ عدة مهام يمكن حصرها في التالي:

1- يجب أن يكون لدى برنامج عميل الويب القدرة على تفسير لغة HTML والتي تستخدم لظهور المعلومات والعناصر من نص وصورة وجداول وصوت وملفات حركة بشكل منسق داخل صفحة الويب.

يجب أن يكون لدى عميل الويب القدرة أيضاً على معالجة أوامر لغة برمجة نصية (Scripting Language) مثل: لغة JavaScript، ولغة VBScript، حيث يحتوي عميل الويب على كل من محركات تنفيذ هذه اللغات (Script Engine)، مع العلم أن معظم برامج عميل الويب لديها القدرة على تنفيذ أوامر لغة JavaScript (محور دراسة الوحدة الثالثة من هذا الكتاب). تستخدم هذه اللغات لتنفيذ بعض الأوامر لدى عميل الويب وهي تسمى البرمجة لدى عميل الويب (Client-Side Programming)، مثل تطوير برامج لدى عميل الويب للتحقق من إدخال بيانات صحيحة قبل إرسالها لخادم الويب، كما يحتوي عميل الويب أيضا على مكتبات هذه اللغات والتي تتكون من دوال (Functions) معرفة مسبقا بواسطة مطوري اللغة، حيث يتمكن المبرمج من استدعاء هذه الدوال داخل البرامج التي ستعالج لدى عميل الويب.

2- يحتوي عميل الويب أيضاً على مكتبة لغة HTML الديناميكية (Library DHTML) والتي تتكون من مجموعة من الكائنات (Objects) التي تحتوي على دوال وصفات هذا الكائن، توفر مكتبة لغة HTML الديناميكية للمبرمج إمكانية تغيير شكل عناصر صفحة HTML بعد ظهورها أمام المستخدم، وذلك نتيجة تنفيذ برامج فرعية يتم استدعاؤها عند حدوث أحداث معينة، فمثلاً يمكنك أن تظهر صورة مكان أخرى، وذلك عندما يشير المستخدم بمؤشر الفأرة فوق الصورة الأولى (أي صدور الحدث On Mouse Over) حيث يتم استدعاء برنامج فرعي مكتوب بأحد اللغات النصية ولغة DHTML لفعل ذلك.

2.2 خادم الويب Web Server

عزيزي الدارس، أما عن البرنامج الذي يستضيف موقع الإنترنت ويرسل لغة HTML لعميل الويب (سواء أكانت مخزنة في ملفات HTML أو نتيجة تنفيذ ملفات ASP) لتظهر أمامك في شكل معلومات منسقة فيسمى خادم الويب (Web Server)، أو خادم HTTP (HTTP Server) نسبة إلى بروتوكول HTTP ، وسمي خادم الويب بالخادم لأنه يعمل 24 ساعة منتظراً أي طلب يأتي من أحد برامج عملاء الويب.

ومن أمثلة خادم الويب: برنامج IIS (Internet Information Server) الذي يعمل تحت بيئة نظام تشغيل Windows NT ، والأجيال الأعلى مثل نظام تشغيل Windows 2000 ، و نظام تشغيل Windows XP، وبرنامج خادم الويب الشخصي PWS (Personal Web Server) الذي يعمل تحت بيئة نظم التشغيل الأقل صدارة والتي لم تعد تستخدم بكثرة مثل نظم التشغيل Windows 98, 89, Me.

ويتيح لك برنامج خادم الويب (Web Server) أن تقوم بتطوير موقع الويب وتجربته على جهازك الشخصي قبل نقله إلى جهاز الشركة التي ستتعاقد معها لتوفير حضورك على الويب حيث يتيح لك إمكانية استضافة وإدارة مواقع الويب. يجب أن يتوفر لدى برنامج خادم الويب عدة قدرات نلخص أهمها في النقاط التالية:

1- استضافة مواقع إنترنت وانتظار طلبات إرسال ملفات من عملاء الويب، ثم تنفيذ الإجراء المناسب، وذلك بناء على نوع الملف المطلوب، حيث يحتوى موقع الويب على عدة أنواع من الملفات. وفيما يلي نلخص ردود فعل خادم الويب طبقا لنوع الملف المطلوب:

- ملفات HTML: عندما يطلب أحد هذه الملفات من خادم الويب يتم إرساله كما هو دون إجراء معالجة.
- ملفات ASP: عندما يطلب أحد هذه الملفات من خادم الويب يتم معالجة شفرة ASP المتواجدة داخل هذا الملف، ثم إرسال ناتج المعالجة وبقية محتوى الملف.

- ملفات أخرى (مثل ملفات الصوت، الصور... إلخ): عندما يطلب أحد هذه الملفات من خادم الويب يتم إرسالها كما هي دون إجراء معالجة.

2- تنفيذ أوامر لغات برمجة نصية (Scripting Language) مثل لغة JavaScript ولغة VBScript، حيث يحتوي خادم الويب على محرك تنفيذ كل من هاتين اللغتين (Script Engine)، مع العلم أن لغة VBScript، تمثل اللغة الافتراضية لكتابة برامج المعالجة لدى الخادم (محور دراسة الوحدة الخامسة). تستخدم هذه اللغة لتنفيذ بعض البرامج لدى خادم الويب والتي تسمى البرمجة لدى خادم الويب (Server-Side Programming)، مثل: تطوير برامج اتصال بقاعدة بيانات وتنفيذ استعلام معين على البيانات وإصدار نتيجة الاستعلام بلغة HTML ليتم إرسالها وعرضها داخل جدول لعميل الويب، ويحتوي أيضاً خادم الويب أيضاً على مكتبات هذه اللغات (دوال معرفة مسبقة من قبل مطوري اللغة) والتي تمكن المبرمج من استدعاء دوال جاهزة.

3- احتواء على مكتبة لغة ASP وهى اختصار (Active Server Pages) والتي تتكون من مجموعة من الكائنات (Objects) حيث يحتوي كل كائن على دوال وصفات لهذا الكائن، والتي تمكن المبرمج من تنفيذ عدة مهام مثل استقبال بيانات من المستخدم من خلال نموذج، وإرسال نتائج المعالجة، الخ.

3.2 بروتوكول HTTP

عزيزي الدارس، يستخدم بروتوكول HTTP (اختصار لجملة Hypertext Transfer Protocol) لإجراء حوار بين كل من عميل الويب (Web Client) وخادم الويب (Web Server)، حيث يطلب عميل الويب من خادم الويب إرسال محتويات ملف (مستند HTML أو ملف ASP أو ملف آخر) عندئذ يستجيب خادم الويب باتخاذ الإجراء المناسب بناء على نوع الملف المطلوب كما أوضحنا سابقا.

ويجب أن ننوه إلى أن برتوكول HTTP هو أحد برتوكولات خدمات شبكة الإنترنت الذي يعنى بخدمة الإبحار والتجول داخل صفحات الويب (Web Surfing) ،

كما توجد برتوكولات أخرى مثل برتوكول FTP (اختصار لجملة File Protocol Transfer) الذي يستخدم لنقل الملفات من وإلى خادم الويب (File Uploading & Downloading)، حيث يخصص لكل برتوكول منفذ خاص (Port) لا يتكرر مع برتوكول آخر، مع العلم أن رقم المنفذ الخاص لبرتوكول HTTP يساوي 80 ، ولذلك عندما يتصل عميل الويب بخادم الويب لابد أن يستعمل كلاً من عنوان خادم الويب ورقم منفذه (80).

4.2 قواعد البيانات (Databases)

إن القدرة على التعامل مع قواعد البيانات تعد من أهم خصائص تقنية ASP،حيث يمكن استخدامها لجعل المعلومات المحفوظة في قاعدة البيانات متاحة للمستخدمين الذين يزورون موقع الويب، واستخدام قاعدة البيانات لحفظ بيانات الموقع واستخدام ملفات ASPللوصول إلى هذه البيانات تمثل طريقة فعالة لعرض المعلومات المحدثة بشكل دوري، تتيح ملفات ASP أيضاً للمستخدمين صيانة البيانات الموجودة داخل قاعدة البيانات. فعلى سبيل المثال يمكنك استخدام ملفات ASP لإضافة أو تعديل أو حذف السجلات في قاعدة البيانات.

5.2 لغة HTML

تمثل لغة HTML (اختصار لجملة Hypertext Markup Language) إحدى لغات الكمبيوتر التي تستخدم لإنشاء صفحات الويب (مستندات HTML) وعرضها للمستخدم مستخدما برامج عميل الويب، حيث تتكون لغة HTML من نصوص وعلامات (Tags)، ويمكن إنشاء مستندات HTML مستخدما أي برنامج بسيط لتحرير النصوص مثل برنامج المفكرة (Notepad) الذي يأتي ضمن نظم التشغيل Windows.

ولأن صفحات الويب تستخدم صيغة موحدة وقياسية، فإن الفائدة الكبرى من استخدام صفحات الويب لعرض المعلومات هي أن هذه المستندات يمكن عرضها على أي جهاز حاسب يحتوي على متصفح ويب بما في ذلك أجهزة الكمبيوتر التي تعمل بنظام تشغيل ليونكس، أو نظام الويندوز أو نظام ماكنتوش، ومن الممكن عرض هذه الصفحات

أيضاً على أجهزة الكمبيوتر المحمولة باليد وعلى بعض أجهزة الهاتف المحمول كذلك. ولأن مستندات HTML تحتوي على نصوص فقط، فإنها مثالية لنقل المعلومات على الشبكات البطيئة مثل شبكة الإنترنت.

أسئلة تقويم ذاتي



1) عرف عميل الويب مع ذكر أمثلة له.
2) عرف خادم الويب مع ذكر أمثلة له.
3) ما المقصود بالعبارة التالية "المعالجة لدى العميل؟

6.2 مكتبة كائنات ASP

عزيزي الدارس، تحتوي مكتبة ASP على خمسة كائنات رئيسية (الكائن Response والكائن Request والكائن Server والكائن Session والكائن Application)، يقوم الكائن Response بإرسال المعلومات من خادم الويب إلى عميل الويب، ويقوم الكائن Request بقراءة المعلومات التي يرسلها عميل الويب إلى خادم الويب، ويقوم الكائن Server بإنشاء الكائنات ويوفر إمكانية الوصول إلى معلومات حول عميل الويب، ويقوم الكائن Session بحفظ معلومات الجلسة الخاصة لكل مستخدم على حدة أثناء تجوله داخل موقع الويب، ويقوم الكائن Application بحفظ ومشاركة المعلومات لاستخدامها داخل صفحات الموقع بين جميع مستخدمي الموقع، مع العلم أن كل واحد من هذه الكائنات الخمسة يحتوي على دوال وصفات يمكن أن يستخدمها المبرمج كأدوات تطوير صفحات ASP.

وهناك إصدارات عديدة من تقنية Active Server Pages ، ولكن الإصدار 3 هو أحدث هذه الإصدارات، مع العلم أن هذا الإصدار متوفر مع الإصدار 5 من برنامج خادم الويب IIS ، أما الإصدار 2 من Active Server Pages فهو متوفر مع برنامج

خادم الويب Personal Web Server 4.5 . وكما هو الحال مع معظم البرامج الأخرى، فإن الإصدار يوفر 3 خصائص أكثر من الإصدارات التي تسبقه.

7.2 لغة JavaScript

عزيزي الدارس، إن لغة JavaScript تمثل إحدى اللغات النصية التي اشتهر استخدامها مع لغة HTML لإنشاء صفحات ويب تفاعلية، حيث يمكن أن يحتوى ملف HTML على كل من أوامر JavaScript وشفرة HTML ، ليقوم متصفح الويب بتفسير وعرض شفرة HTML وتنفيذ أوامر JavaScript عند وقوع أحداث معينة مثل الضغط على الزر الأيسر للفأرة. سوف نتعرض لهذه اللغة بالتفصيل في الوحدة الثالثة.

8.2 لغة VBScript

لغة VBScript (اختصار للجملة Visual Basic Scripting) تمثل إحدى اللغات النصية والتي تعتبر اللغة الافتراضية لتطوير صفحات المواقع متغيرة المحتوى (صفحات ASP) والتي يتم معالجتها لدى خادم الويب (Server-Side Processing). تم تطوير لغة VBScript بواسطة شركة مايكروسوفت وهي مبنية على لغة فيجوال بيسك الشهيرة. سوف نتعرض لهذه اللغة بالتفصيل في الوحدة الخامسة.

9.2 المعالجة لدى العميل Client-Side Processing

يقوم متصفح الويب بتفسير وعرض شفرة لغة HTML المستخدمة لإنشاء صفحة الويب، وعندما يتم تضمين أوامر JavaScript في صفحة الويب لمعالجتها لدى عميل الويب، فإن المتصفح سيقوم بمعالجتها، ولأن المعالجة تتم على كمبيوتر المستخدم، فإن هذه المعالجة تسمى المعالجة لدى العميل (Client-Side Processing)، وتتميز المعالجة من جانب العميل بأن خادم الويب لن يضطر إلى تخصيص أي موارد لمعالجة هذه الأوامر، والتي تتصف ببعض العيوب التي يمكن تلخيصها كالتالي:

1. يستطيع المستخدم أن يرى هذه الأوامر عن طريق إظهار شفرة الملف وبالتالي لا يمكن كتابة أوامر سرية يتم تنفيذها لدى العميل.
2. لا يمكن كتابة أوامر تتعامل مع قاعدة بيانات والتي يتم تنفيذها لدى العميل، حيث إن مكان قاعدة البيانات هو خادم الويب.
3. تتطلب بعض الوقت الإضافي لكي يتم نقل رموز HTML وأوامر لغة JavaScript إلى متصفح الويب حيث تتم معالجتها.

10.2 المعالجة لدى الخادم Server-Side Processing

لقد تم تطوير تقنية Active Server Pages من قبل شركة مايكروسوفت والتي تستخدم لإنشاء صفحات ويب ديناميكية وقوية، حيث يتم معالجة هذه الصفحات على خادم الويب، ثم إرسال نتائج المعالجة إلى عميل الويب ،ولذلك تسمى معالجة لدى الخادم (Server Side Processing). وتتميز المعالجة من جانب الخادم بتلافي جميع عيوب المعالجة لدى العميل، ولكن سيضطر خادم الويب إلى تخصيص موارد لمعالجة هذه الأوامر.

يتم استخدام لغة VBScript لتطوير برامج صفحات ASP مستفيدا من كل من مكتبة لغة VBScript ذاتها ومكتبة تقنية ASP، ولأن هذه البرامج سوف تنفذ لدى خادم الويب، فإن متصفحات الويب لا يوجد لها دور في ذلك ومن ثم يمكن استخدام أي متصفح مع مواقع ملفات ASP.

11.2 مكتبة DHTML

قدمت شركة Microsoft مكتبة DHTML اختصار لجملة (Dynamic HTML) والتي تتكون من عدة كائنات (Objects) والتي تستخدم لتغيير ظهور عناصر صفحة HTML بعد عرضها للمستخدم.

عندما يستقبل عميل الويب صفحة HTML من خادم الويب، يقوم بتحويل كل عنصر من عناصر هذه الصفحة إلى كائن (Object)، ولا ننسى أن كل كائن يتسم بعدة صفات

(Attributes) ويمكن أن ينفذ عدة دوال (Methods) حيث يستطيع المبرمج من خلال لغة VBScript أن يكتب أوامر تغير هذه الصفات، أو يستدعي هذه الدوال ليغير عناصر صفحة HTML بعد عرضها للمستخدم، حيث سيتم استدعاء هذه الأوامر كنتيجة لصدور أحداث (Events) مثل حركة الماوس فوق أحد العناصر أو ضغط الزر الأيسر للماوس... الخ.

أسئلة تقويم ذاتي



1. ما المقصود بالمصطلح ASP ؟
2. ما المقصود بالعبارة التالية "المعالجة لدى الخادم"؟
3. ما المقصود بالمصطلح DHTML ؟

الخلاصة

في هذه الوحدة عزيزي الدارس، قدمنا شرحاً مفصلاً عن الأنواع الرئيسية لمواقع الإنترنت ومكوناتها، حيث تنقسم مواقع الإنترنت طبقا لأسلوب إنشاء محتواها إلى مواقع ثابتة المحتوى (Static Web Sites) ومواقع متغيرة المحتوى (Dynamic Web Sites)،

قدمنا تصوراً كاملاً وواضحاً لمكونات إطار عمل تشغيل صفحات الويب لكي نعطي للقارئ فكرة مختصرة عن مكونات هذا الإطار، حيث تكمن صعوبة تطوير المواقع متغيرة المحتوى (Dynamic Web Sites) في وجود العديد من المفاهيم والتقنيات التي تحتاج إلى توضيح، ومن ثم رأينا أن نقدم في هذه الوحدة شرحاً وافياً عن كل من عميل الويب (Web Client) و خادم الويب (Web Server) وبروتوكول HTTP ولغة HTML ، وكل من مكتبة كائنات ASP ومكتبة DHTML ، وكل من لغة JavaScript ولغة VBScript ، والمعالجة لدى العميل (Client-Side Processing)، والمعالجة لدى الخادم (Server-Side Processing)

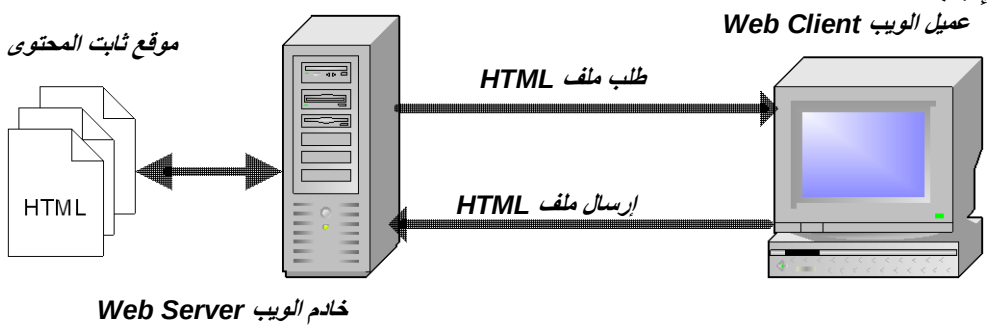
لمحة مسبقة عن الوحدة التالية

عزيزي الدارس، في وحدتنا التالية سنتناول برنامج **Microsoft Front** Page لترى الطريقة التي ستظهر بها الصفحة داخل مستعرض الويب أثناء مرحلة التصميم، ومن ثم سوف نستخدم هذا البرنامج خلال هذه الوحدة لتصميم صفحات الويب من خلال شرح عناصر لغة HTML بإعطاء نبذة عن العنصر، ثم توضيح شفرة HTML المناظرة له، ثم عرض شرح تفصيلي عن كيفية إنشاء هذا العنصر مستخدمين برنامج Microsoft Front Page 2000 لانتشار وجوده بين مستخدمي الحاسب نتمنى أن تستمتع بدراسة هذا الجزء من رحلة تطوير صفحات الويب.

إجابات التدريبات

تدريب (1)

الإجابة

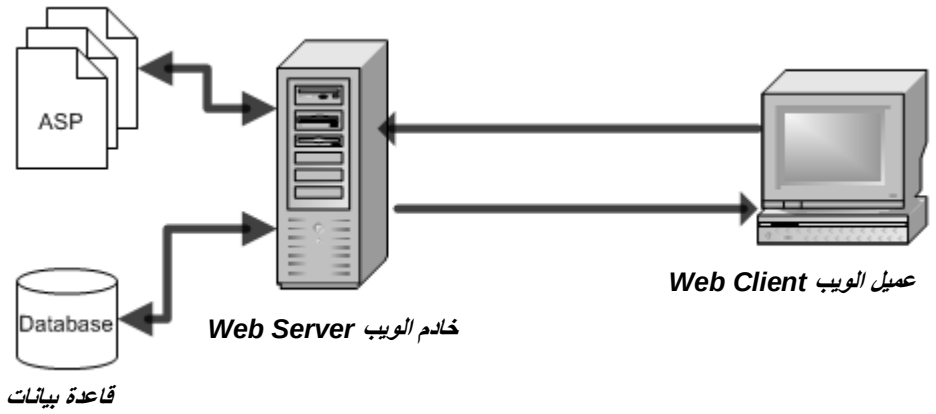


1. عميل الويب يرسل لخادم الويب يطلب ملف HTML.
2. يستجيب خادم الويب ويرسل الملف لعميل الويب.

تدريب (2)

الإجابة:

يمكن تخزين البيانات المراد ظهورها للمستخدم بشكل ديناميكي داخل قاعدة بيانات متصلة بخادم الويب، وبالتالي تكون هذه القاعدة هي مصدر البيانات لصفحات ASP كما هو مبين بالرسم.



3- اذكر باختصار القدرات التي يجب أن تتوفر لدى عميل الويب لكي يتمكن من تنفيذ مهامه؟.

الإجابة:

- يجب أن يكون لدى برنامج عميل الويب القدرة على فهم وتفسير لغة HTML.
- قدرة تنفيذ أوامر لغة برمجة نصية (Scripting Language) مثل لغة JavaScript ولغة VBScript، حيث يحتوي عميل الويب على محرك تنفيذ هذه اللغات.
- يحتوى عميل الويب على مكتبة لغة HTML الديناميكية (DHTML library) والتي تتكون من مجموعة من الكائنات Objects حيث يتمكن المبرمج من تغيير شكل عناصر صفحة HTML بعد ظهورها أمام المستخدم.

مسرد المصطلحات

❖ **مواقع ثابتة المحتوى (Static Web Sites)**

أطلق على هذه المواقع اسم المواقع ثابتة المحتوى، لأن محتوياتها لا تتغير مع الوقت، بمعنى أننا إذا تصفحنا هذه المواقع آلاف المرات ستظهر لنا بنفس المحتويات دون تغيير، حيث تتكون هذه المواقع من ملفات تحتوي على شفرة HTML

❖ **مواقع متغيرة المحتوي (Dynamic Web Sites)**،

على عكس المواقع ثابتة المحتوى (Static Web Sites) ، فإن المواقع متغيرة المحتوى (Dynamic Web Sites) تتكون من نوع آخر من أنواع الملفات والتي تحتوي على شفرة Active Server Pages ASP)

HTML

وهي اللغة المسؤولة عن ظهور العناصر (مثل النصوص، الجداول، الصور... إلخ) داخل صفحات الويب.

المراجع

[1] عبدالكريم العقيل، "المخطط الأزرق لبناء صفحات ويب ديناميكية"، مكتبة جرير، الرياض، 2001م.

[2] د. أحمد أحمد شعبان دسوقى، أ. د. السيد محمود الربيعي، "تقنيات وبرمجة الإنترنت" مكتبة الرشد، الرياض، 2004م.

[3] د. عبد القادر عبد الله الفنتوخ، "الإنترنت للمستخدم العربي"، مكتبة العبيكان، الرياض، 2001م.

[4] د. إبراهيم فتحي معوض، "أساسيات برمجة الإنترنت"، معهد العالمية للحاسب والتقنية، 2005م.


الوحدة الثانية
تطوير مواقع ثابتة المحتوى
Microsoft Front Page

محتويات الوحدة

الموضوع	رقم الصفحة
المقدمة	31
تمهيد	31
أهداف الوحدة	32
1. إصدارات لغة HTML	33
2. إنشاء موقع جديد (New Web Site)	33
3. إضافة صفحة ويب جديدة إلى الموقع الحالي	35
4. التركيب الأساسي لمستند HTML	36
5. تصميم خلفية صفحات الويب	38
6. إضافة Theme إلى خلفية الصفحة	39
7. رابطة الاستدعاء Hyperlink	40
8. زر استدعاء ذو تأثير خاص (Hover Button)	42
9. الجداول (Table)	43
11. الصور (Images)	48
10. النصوص المتحركة (Marquees)	51
11. التعامل مع الإطارات (Frames)	52
12. النماذج الحوارية (Forms)	55
الخلاصة	67
لمحة مسبقة عن الوحدة التالية	67
إجابة التدريبات	68
المراجع	69

المقدمة

تمهيد

عزيزي الدارس، مرحبا بك في الوحدة الثانية من مقرر "تقنيات وبرمجة الإنترنت"

لقد ذكرنا في الوحدة الأولى من هذه الكتاب أن لغة HTML تستخدم في إنشاء صفحات ويب ثابتة المحتوى (Static Web Pages)، ولذلك كان يجب علينا أن نخصص هذه الوحدة لشرح عناصر هذه اللغة (مثل الجداول والصور والإطارات) وكيفية تصميم مواقع ثابتة المحتوى مستخدمين برنامج Microsoft Front Page، هذا بالإضافة إلى شرح تفصيلي لكيفية تصميم النماذج الحوارية (Forms) والتي تستخدم لاستقبال بيانات من المستخدم. من الممكن عزيزي الدارس أن تستخدم أحد برامج تحرير النصوص البسيطة (مثل برنامج المفكرة الذي يأتي مع نظام تشغيل الويندوز) لكي تقوم بإنشاء وتعديل وحفظ صفحات الويب، ولكن لا تستطيع مثل تلك البرامج التعرف على العلامات (Tags) الخاصة بلغة HTML أو إدراجها تلقائياً في النص، لذلك يجب عليك إدراج علامات HTML بنفسك لكي تحدد كيفية ظهور عناصر صفحة الويب عند عرضها داخل الويب (Web Browser).

وهناك أيضاً العديد من البرامج المتخصصة في إنشاء صفحات الويب، مثل: برامج تحرير HTML، وبرامج التحرير المرئي (Visual Editors). وهي برامج مصممة خصيصاً لمساعدتك في إنشاء صفحات الويب، ومن أشهر هذه البرامج برنامج Microsoft Front Page.

عزيزي الدارس، أهلا بك مرة أخرى إلى هذه الوحدة وعسى أن تنتفع بها وأن تفيد منها، وأن تساعدنا في نقدها وتطويرها.

أهداف الوحدة



عزيزي الدارس، بعد فراغك من دراسة هذه الوحدة ينبغي أن تكون قادراً على أن:

- تتعرف على كيفية إنشاء موقع جديدة.
- تستطيع إضافة صفحة ويب جديدة إلى الموقع الحالي.
- تُصَمَّمُ خلفية صفحات الويب .
- تضيف الصور إلى صفحة الويب .
- تتعامل مع الإطارات في صفحة الويب .
- تدرج نماذج الحوار في صفحة الويب .

1. إصدارات HTML

لقد تم إصدار عدة إصدارات من لغة HTML، حيث يتم تطويرها باستمرار، ويتم طرح إصدار جديد منها كل عامين تقريباً، ويوفر كل إصدار جديد خصائص جديدة تعطي المستخدمين مزيداً من التحكم عند إنشاء صفحات الويب. تقوم رابطة الويب World Wide Web Consortium (تختصر W3C) بتنظيم إصدارات HTML ، ويمكنك عزيزي القارئ أن تستعرض الكثير من المعلومات عن لغة HTML وعن إصدارتها من خلال الموقع www.w3c.org.

2. إنشاء موقع جديد (New Web Site)

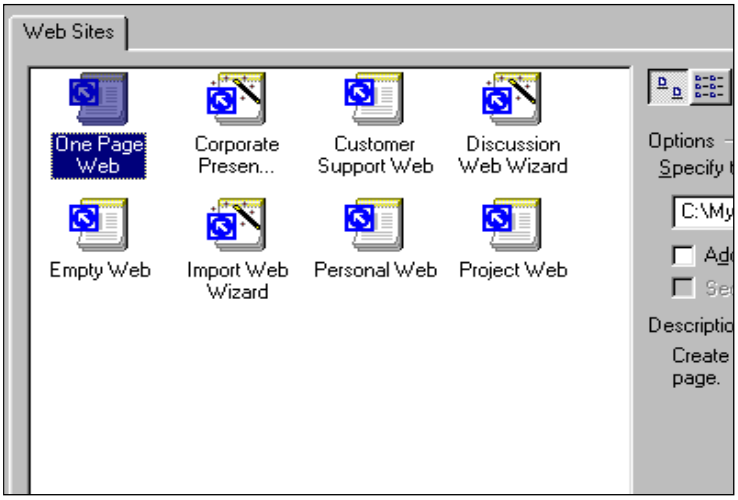
عزيزي الدارس، يتكون موقع الويب ثابت المحتوى (Static Web Site) من مجموعة من ملفات HTML وأنواع أخرى من الملفات مثل ملفات الصور والصوت. ولكي تُنشئ موقعاً جديداً مستخدما برنامج Microsoft Front Page الذي يمثل الحاوية لصفحات الويب التي تريد إدراجها، اتبع الخطوات التالية:

الخطوة 2 اذهب إلى القائمة File ثم اختر منها الأمر New ثم اختر الأمر Web كما هو واضح في الشكل رقم (1)



شكل 1-2: اختيار الأمر Web لإنشاء موقع جديد

الخطوة 2 عند تنفيذ الأمر السابق، سوف يظهر لك الصندوق الحواري الموضح في شكل 2-2، لاحظ أنه يوجد اختيارات مختلفة داخل الصندوق الحواري التالي والتي تمثل أنواع المواقع المختلفة التي يمكن تصميمها، ومن أهم هذه الاختيارات الاختيار One Page Web لبناء موقعك بنفسك (صفحة تلو الأخرى)، ولاختيار Personnel Web لتصميم موقعٍ شخصيّ، ولتكملة خطوات إنشاء موقع جديد، فضلا قم باختيار الاختيار الأول.



شكل 2-2: اختيار أحد أنواع مواقع الويب (Web Sites)

الخطوة 3 ـ ن الموقع

على القرص للموقع.

لخطوة 4 سو (Folde،

انقر نقراً مزدوجًا على هذه الصفحة لكي تقوم بفتحها في موقعك، ولاحظ أنه يجب إبقاء هذه الصفحة بنفس الاسم مع عدم تغييره.

3. إضافة صفحة ويب جديدة إلى الموقع الحالي

ولكي تنشأ صفحة ويب جديدة إلى الموقع الحالي، اتبع الخطوات التالية:

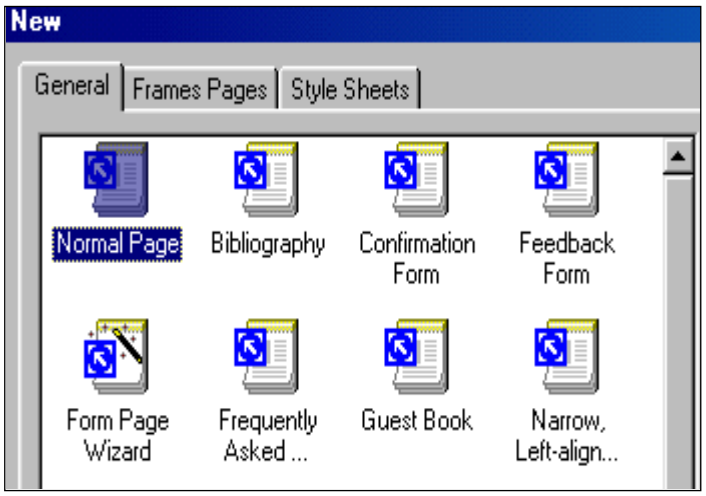
الخطوة 1. اذهب إلى القائمة File ثم اختر منها الأمر New، ثم اختر الأمر Page كما هو واضح في الشكل رقم 3-2.



شكل 3-2: اختيار الأمر Page لإنشاء موقع جديد

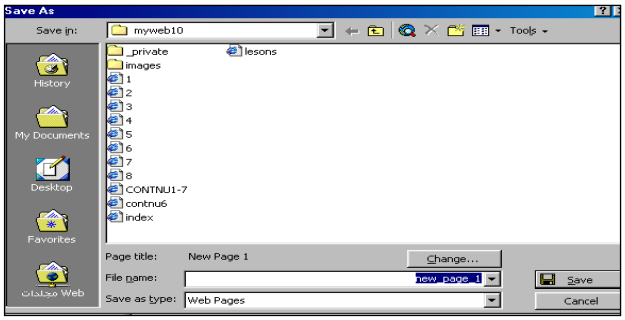
الخطوة 2. عند تنفيذ هذا الأمر سوف يظهر لك الصندوق الحواري الموضح في شكل 4-2، اختر الخيار الأول ثم انقر على الزر Ok (إذا كنت تريد أن تنشئ صفحة فارغة)، عندئذ ستظهر لك الصفحة الجديدة داخل الموقع.

الخطوة 3.



شكل 4-2: اختيار أحد أنواع صفحات الويب (Web Page)

الخطوة 4. أما إذا أردت حفظ الصفحة الجديدة (أو تغيير اسم صفحة موجودة) فما عليك إلا أن تختار القائمة File ثم الاختيار Save As، عندئذ سوف يظهر لك الصندوق الحواري الموضح في شكل 5-2



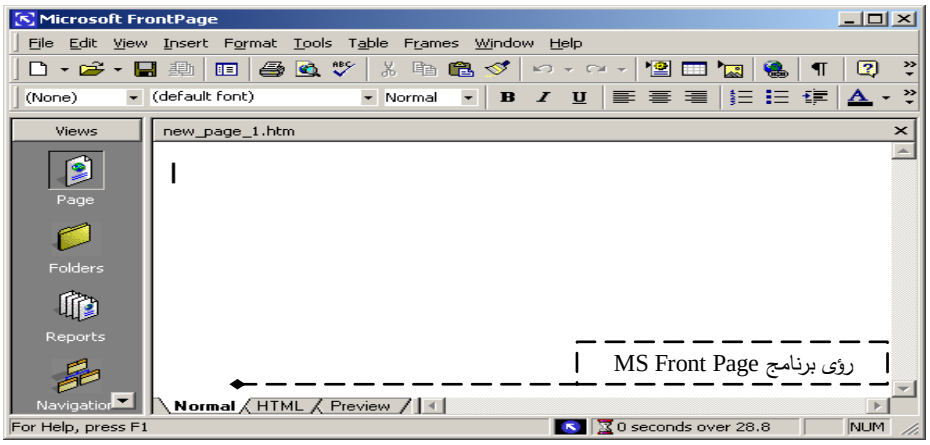
الخطوة 5. غير اسم الصفحة للاسم الذي تريده ثم انقر على الزر Save.

4. التركيب الأساسي لمستند HTML

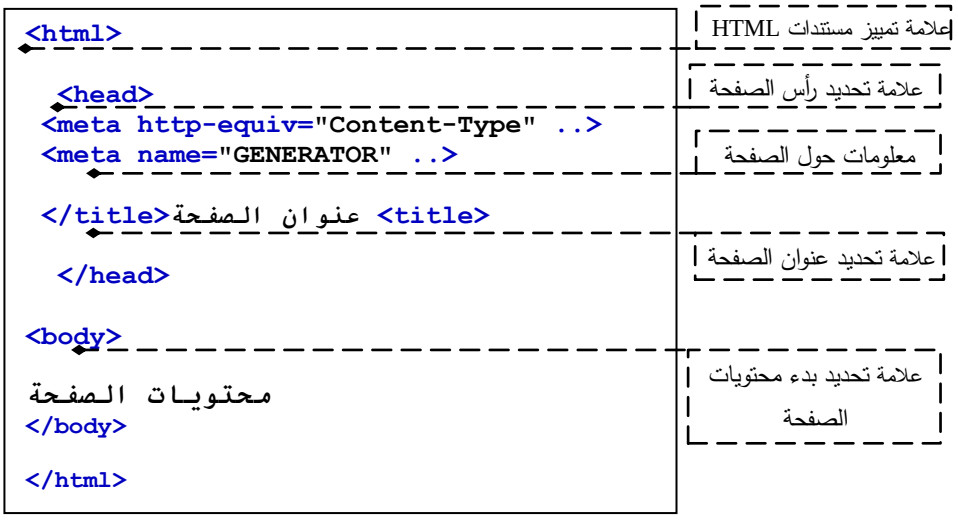
عزيزي الدارس، عند إنشاء ملف HTML جديد مستخدماً برنامج MS FRONT PAGE تظهر لك الصفحة بيضاء داخل رؤية Normal (لاحظ أنه يوجد ثلاث رؤى داخل البرنامج: رؤية Normal ورؤية HTML ورؤية Preview كما هو واضح في شكل رقم 2-6 ، وعند الضغط على رؤية HTML تظهر لك شفرة HTML المناظرة للتصميم الحالي، ويوضح شكل رقم 2-7 التركيب الأساسي لمستند HTML، حيث يحتوي الشكل على وصف رسومي مختصر لأهم العلامات (Tags) الخاصة بلغة HTML.

تقوم العلامات (Tags) بإخبار مستعرض الويب بهيكل وتصميم المستند حيث تختص كل علامة بوظيفة محددة حيث يتم إحاطة كل علامة بقوسي الزاوية < > ، مع العلم أن معظم العلامات تتكون في الحقيقة من علامتين: علامة بداية، وعلامة نهاية حيث تؤثر العلامة في النص المحصور بين علامتي البداية والنهاية. إن بعض علامات HTML تحتوي على سمات (Attributes) داخل علامة البداية والتي يمكن استخدامها

للتحكم في خيارات محتوى العلامة، فعلى سبيل المثال، تحتوي علامة <FONT> (التي تستخدم لتغيير خط النص) على سمة COLOR التي يمكن استخدامها لتغيير لون النص المحصور بين كل من علامة البداية والنهاية.



شكل 2-6 : نافذة برنامج MS Front Page

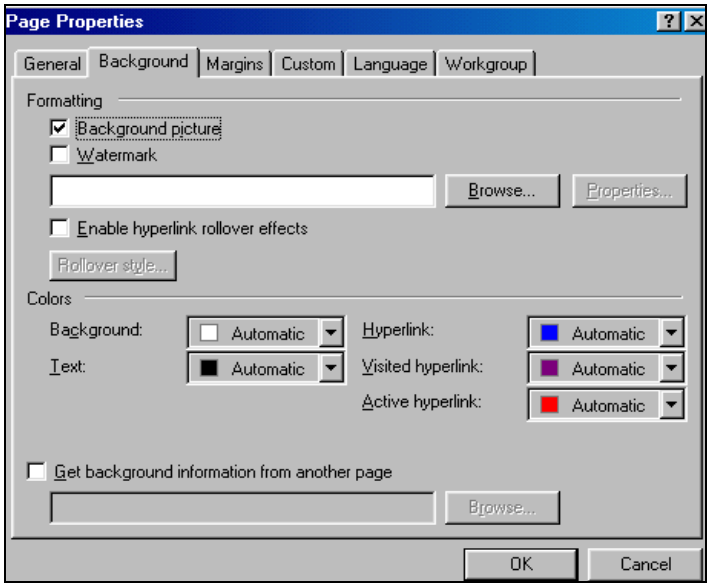


شكل 2-7: التركيب الأساسي لمستند HTML

5. تصميم خلفية صفحات الويب

عزيزي الدارس تظهر خلفية صفحات الويب عند بداية إنشائها باللون الأبيض، ولتصميم خلفية مناسبة لصفحات موقع الويب الخاص بك اتبع الخطوات التالية:

الخطوة 1. اضغط بزر الماوس الأيمن في أي مكان فارغ داخل صفحة الويب التي تريد أن تغير خلفيتها، ثم قم باختيار الأمر Page Properties ، عندئذ سيظهر الصندوق الحواري الموضح في شكل 8-2.



شكل 8-2 : صندوق تحديد صفات صفحات الويب

الخطوة 2. لاحظ أن الصندوق الحواري الموضح في شكل 8-2 يحتوى على عدة اختيارات تمكنك من تغيير خصائص خلفية صفحة كما سنرى خلال الخطوات التالية.

الخطوة 3. إذا كنت تريد أن تضع صورة كخلفية للصفحة، اضغط الاختيار Background Picture ثم اضغط على الزر Browse لإظهار

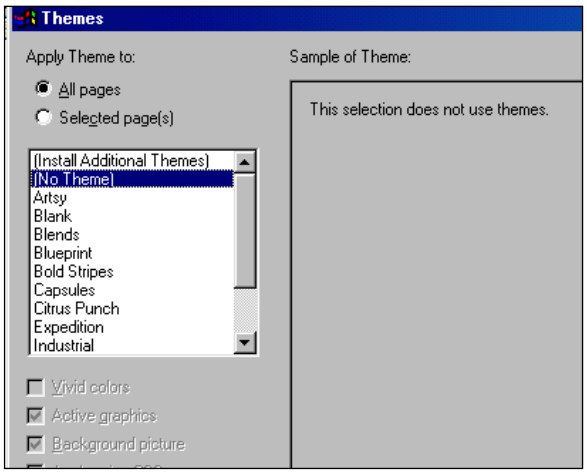
صندوق اختيار ملف الذي يمكنك من اختيار ملف الصورة المراد وضعها كخلفية للصفحة، وإذا أردت أن تجعل الصورة ثابتة (أي علامة مائية) اخترْ الاختيار Watermark.

الخطوة 4. كما يمكنك اختيار لون الخلفية (غير اللون الأبيض الافتراضي)، وذلك من خلال الاختيار Background الذي يظهر أسفل الجزء Color لاختيار اللون المناسب.

6. إضافة Theme إلى خلفية الصفحة

لكي تضفي منظرا جميلا (Theme) لخلفية صفحات الويب، يمكنك استخدام الخلفيات الجاهزة الآتية مع برنامج MS Front Page وذلك مستخدما إحدى الطريقتين التاليتين.

الطريقة الأولى: انقر بالزر الأيمن للماوس في أي مكان فارغ داخل صفحة الويب المراد تغيير خلفيتها، ثم اخترْ الأمر Theme، عندئذ سوف يظهر لك الصندوق الحواري الموضح في شكل 9–2 الذي يوضح أن الاختيار الافتراضي للصفحة الحالية هو No Theme(أي عدم احتواء الصفحة الحالية على خلفيات جاهزة)، والآن يمكنك اختيار تصميم جاهز (Theme) من قائمة التصميمات الجاهزة، حيث يظهر لك شكل التصميم الذي تختاره في الجزء الأيمن من الصندوق الحواري، وإذا أردت تطبيق هذا التصميم (Theme) على جميع صفحات الموقع، اخترْ الاختيار All Pages، أما إذا أردت تطبيق هذا التصميم (Theme) على صفحة معينة/صفحات محددة، اخترْ الاختيار Selected Page(s).



شكل 9-2 نافذة اختيار الخلفيات الجاهزة (Themes)

الطريقة الثانية: افتح قائمة Format ثم قم باختيار الأمر Theme، سيظهر لك الصندوق الحواري الموضح في شكل 9–2، وعندئذ يمكنك اختيار خلفية لصفحات الموقع مستخدما نفس خطوات الطريقة الأولى.

7. رابطة الاستدعاء (Hyperlink)

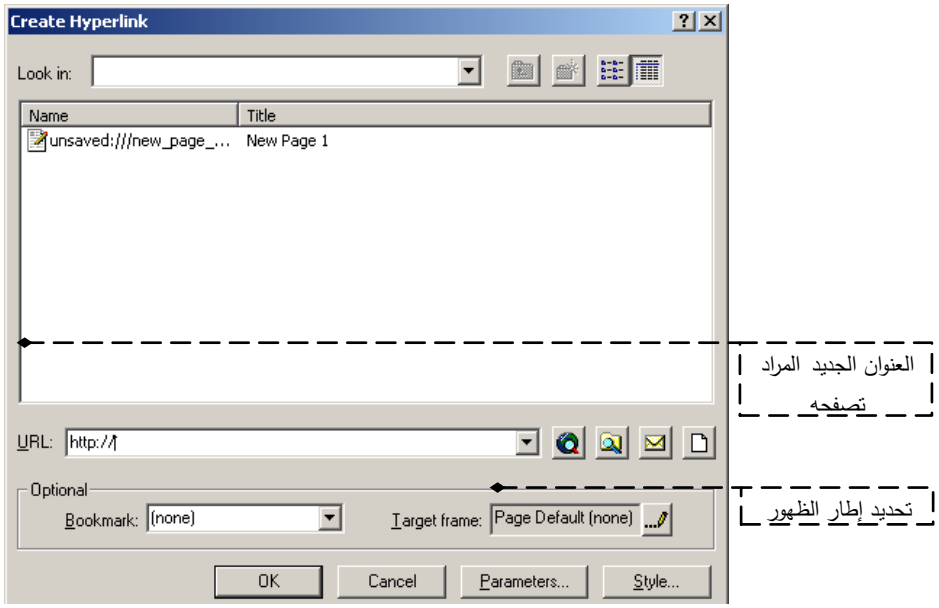
عزيزي الدارس، عندما تريد أن تنتقل من صفحة ويب إلى صفحة ويب أخرى داخل نفس موقع الإنترنت أو حتى عندما تريد أن تنتقل من موقع إلى موقع آخر، فإنه يجب عليك أن تفعل شيئاً من أثنين: كتابة العنوان الجديد الذي تريد أن تتصفحه داخل قائمة العنوان أعلى نافذة المتصفح، أو تضغط على رابطة استدعاء لاستدعاء الصفحة الجديدة وظهورها في الإطار المحدد للظهور.

ولكي توفر إمكانية استدعاء صفحات ويب (أو مواقع ويب) أخرى من

داخل أحد صفحات موقعك وإظهارها داخل متصفح الويب، يجب أن تضيف رابطة استدعاء مع نص أو مع صورة، وفيما يلي سنشرح خطوات تصميم رابطة استدعاء وشفرة HTML المناظرة لها.

خطوات التصميم

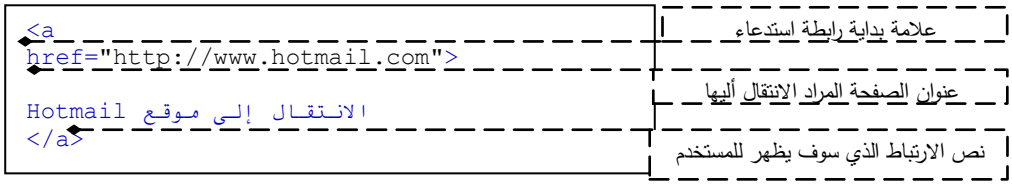
يتم اختيار النص (أو الصورة) المراد إلحاق رابطة استدعاء (Hyperlink) به ، ثم اختيار قائمة Insert، ثم اختيار الأمر Hyperlink، عندئذ ستظهر نافذة حوارية تمكنك من كتابة العنوان الجديد وتحديد إطار الظهور (نافذة جديدة، الإطار الحالي النشط أو إطار آخر داخل النافذة الحالية) كما هو واضح في شكل 10-2.



شكل 10-2 : وصف مكونات نافذة إنشاء رابطة الاستدعاء

شفرة HTML المناظرة

يوضح شكل رقم 11-2 شفرة HTML المقابلة لإنشاء رابطة استدعاء، حيث يحتوي الشكل على وصف رسومي مختصر لأهم علامات (Tags) الخاصة بلغة HTML.

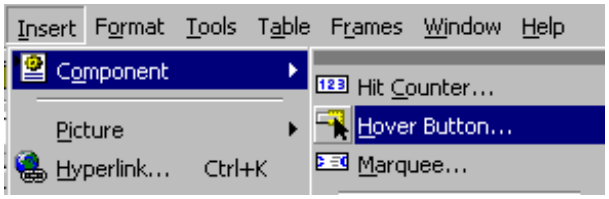


شكل 11- 2: وصف رسومي لأهم علامات HTML لإنشاء رابطة استدعاء

8. زر استدعاء ذو تأثير خاص (Hover Button)

كما يمكنك أن تضيف أزراراً ذات تأثير شيق (Hover Buttons) لصفحات موقعك لتوفير إمكانية التنقل بين صفحات الموقع المختلفة، وتتميز هذه الأزرار بإصدار تأثيرات جميلة عند الضغط أو الوقوف عليها كأن يضيء الزر عند تمرير الماوس عليه، ولإدراج هذه الأزرار، اتبع الخطوات التالية:

الخطوة 1. قم باختيار القائمة Insert ثم اختر الأمر Component، ثم اضغط على الأمر Hover Button كما هو واضح في الشكل 12-2

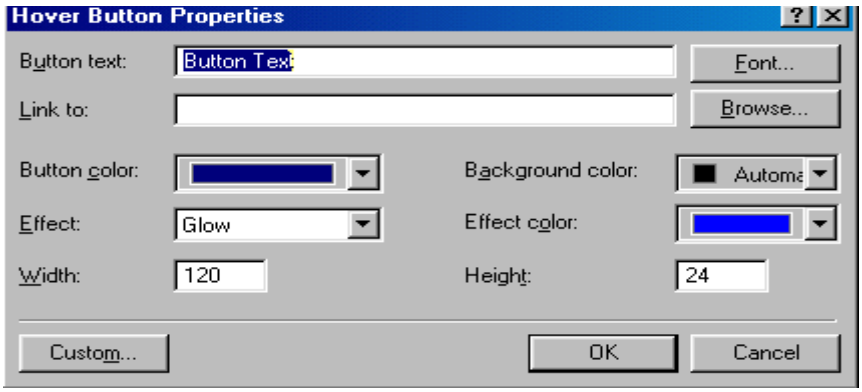


شكل 12-2 : اختيار الأمر Hover Button

الخطوة 2. عندئذ سيظهر لك الصندوق الحواري الموضح في شكل 13-2، لاحظ أنه يوجد عدة اختيارات داخل هذا الصندوق، وفيما يلي شرح مختصر عن كل اختيار:

- يستخدم الاختيار Button Text لكتابة النص الذي يمثل عنوان الزر.

- يستخدم الاختيار Link To لكتابة عنوان الصفحة/الموقع الذي تريد أن تنتقل إليه عند الضغط على هذا الزر، يمكنك الاستعانة بالزر Browse لاستعراض صفحات موقعك.
-



شكل 13-2 : خصائص الزر Hover Button

يستخدم الاختيار Button Color لتغيير لون الزر.

- يستخدم الاختيار Effect لتغيير نوع التأثير.
- يستخدم الاختيار Effect Color لتغيير لون التأثير.
- الاختيارات Wide/Height لتغيير ارتفاع/عرض الزر.

9. الجداول Tables

عزيزي الدارس من أهم العناصر التي تدرج بكثرة داخل صفحات الويب هي الجداول، حيث يوجد بها عدة استخدامات، فعلى سبيل المثال يستخدم الجدول لعرض بيانات جدوليه مثل عرض الجدول الدراسي، أو جدول رحلات داخل موقع لشركة سياحة...إلخ، وبالإضافة إلى ذلك، يستخدم الجدول كأداة تنسيق، حيث يمكن تنسيق عناصر مختلفة من نص، وصور، وأفلام فيديو داخل الجدول، حيث يمكن تقسيم خلايا

الصفوف بأبعاد مختلفة، ويمكن أن تحتوي الصفوف على عدد خلايا مختلفة، وهكذا. وفي هذه الحالة يمكن إخفاء حدود الجدول، وبالتالي تظهر العناصر منسقة دون ظهور حدود الجدول للمستخدم.

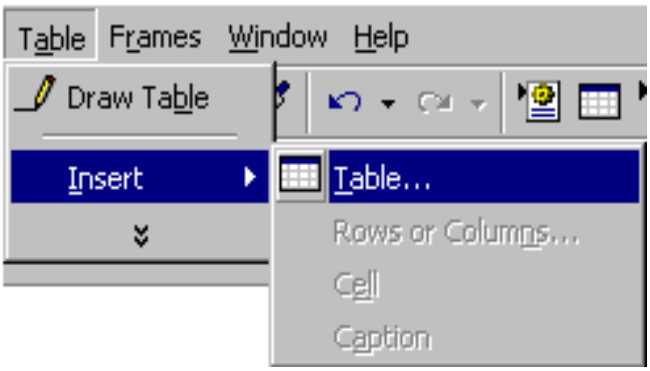
ومن الأمثلة المشهورة لاستخدام الجداول في تنسيق العناصر، عرض منتجات داخل موقع تجاري (Electronic Catalog)، حيث يتم عرض صورة المنتج، ووصف المنتج، وسعر المنتج، وزر إضافة المنتج إلى سلة المشتريات (Shopping Cart) كما هو واضح في شكل رقم 14-2.



شكل 14-2 : استخدام الجدول كأداة تنسيق منتجات داخل موقع تجاري

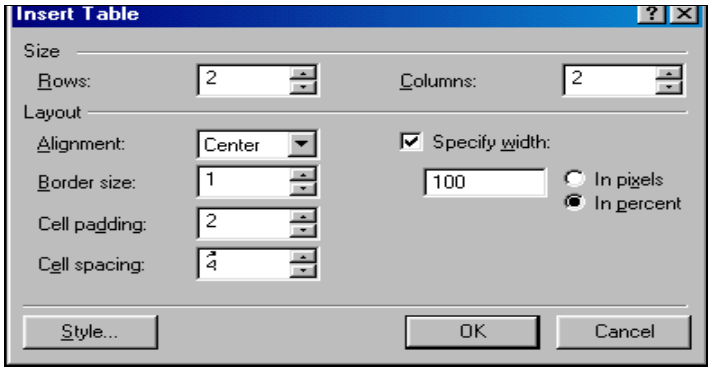
ولكي ندرج جدولاً داخل إحدى صفحات الويب أو تعدل صفات جدول، اتبع الخطوات التالية:

الخطوة 1. افتح القائمة الفرعية Table ، ثم اختر الأمر Insert ، ثم اختر الأمر Table كما هو واضح في الشكل 15-2.



شكل 2-15 :اختيار الأمر Table لإدراج جدول

الخطوة 2. عندئذ سيظهر الصندوق الحواري الموضح في شكل 2-16 لاحظ أنه يوجد عدة اختيارات في الصندوق الحواري لتغيير كل من: عدد الصفوف (Rows)، وعدد الأعمدة (Columns)، ومكان الجدول (يسار أو يمين أو وسط) وسمك حدود الجدول (Border Size) والمسافة بين خلايا الجدول (Cell Spacing)، والمسافة الداخلية بين محتويات الخلية وحدود الخلية (Cell Padding) وبعد ضبط قيم هذه الصفحات بناء على احتياجات الجدول الذي تريد أن تدرجه، قم بالضغط على الزر OK.



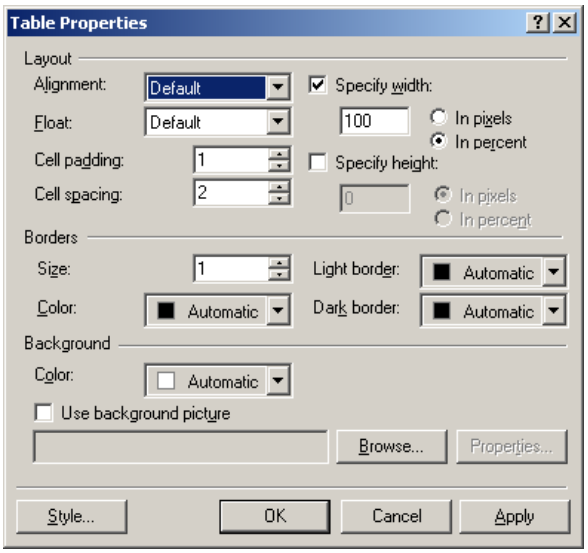
شكل 2-16 : نافذة خصائص الجدول

الخطوة 3. والآن بالطبع تستطيع الكتابة داخل خلايا الجدول المدرج وتنسيق هذه النصوص، أو إدراج صور أو إدراج روابط استدعاء كما تعلمنا في الدروس السابقة.

خطوات تغيير تصميم الجدول

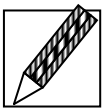
كما يمكننا تغيير تصميم الجدول بعد إدراجه داخل صفحة الويب عن طريق تغيير سمات الجدول ، ولتوضيح ذلك اتبع الخطوات التالية:

الخطوة 1. ضع مؤشر الفأرة داخل الجدول المراد تغيير تصميمه، ثم قم باختيار القائمة Table، ثم اخترْ القائمة Properties، ثم الأمر Table، حيث تظهر النافذة المبينة في شكل رقم 17–2 الذي تمكنك من تغيير سمات الجدول.



شكل 17-2 : شاشة سمات الجدول

تدريب (1)

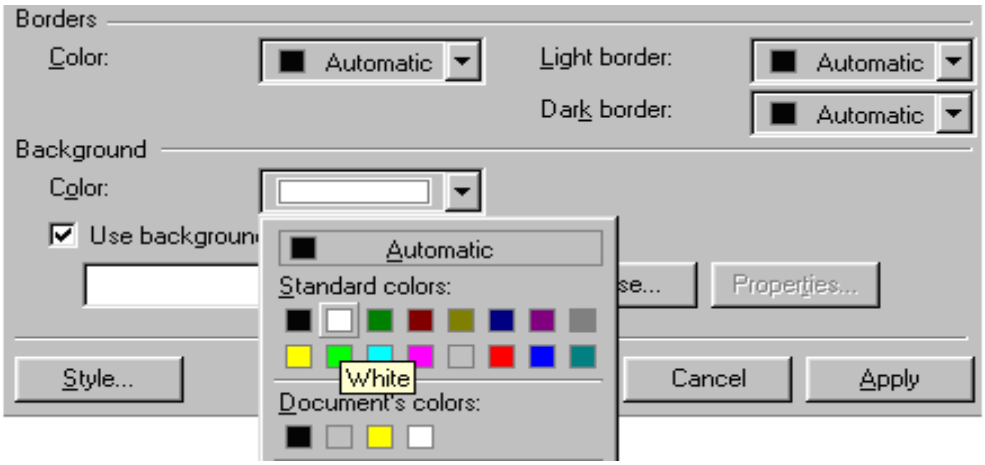


1. اكتب شفرة HTML المسؤولة عن ظهور الجدول التالي.

اسم الطالب	الرقم الجامعي
على الفاضل	43666666
حسن البصري	5656567
ماجد العلي	6756577

الخطوة 2. افترض أنك تريد أن تسند لوناً محدداً لخلفية الجدول، قم باختيار قائمة Color (أسفل الجزء Background)، ثم اختر اللون المناسب، ولإسناد صورة للخلفية بدلا من اللون، قم بالتأشير على الاختيار Use Background Picture، ثم أختر الصورة المناسبة مستخدما الزر Browse .

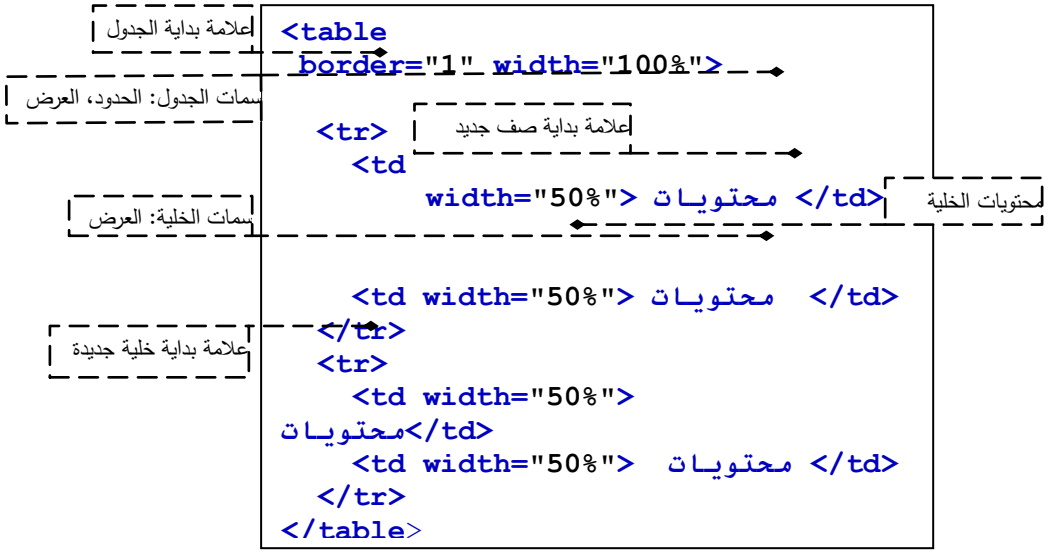
الخطوة 3. ولتغيير خلفية خلية ما، انقر على الخلية التي تريدها بالزر الأيمن للفأرة، عندئذ ستظهر قائمة فرعية، قم باختيار الأمر Cell Properties، سيظهر لك الصندوق الحواري الموضح في شكل 18–2 .والآن يمكنك تغيير خلفية الخلية متبعا نفس أسلوب تغيير خلفية الجدول.



شكل 18-2 : شاشة سمات خلية جدول

شفرة HTML المناظرة للجدول

يوضح شكل رقم 19–2 شفرة HTML المقابلة للجدول، حيث يحتوي الشكل على وصف رسومي مختصر لأهم علامات (Tags) الخاصة بلغة HTML لتصميم جدول.



شكل 19-2 : وصف رسومي لأهم علامات HTML لتصميم جدول

10. الصور (Images)

عزيزي الدارس، إن أهمية استخدام الصور (Images) يعكسها المثل الصيني الذي يقول "إن الصورة تغني عن ألف كلمة ، وبالتالي يفضل استخدام الصور في عرض المعلومات إن وجدت، مع العلم أنه يوجد أكثر من نوع للصور، وهذا يعتمد على الشكل التركيبي للصورة (Image Format). ومن أهم هذه الأنواع: النوع JPG ، والنوع GIF. تتميز صورة JPG بدقة وضوحها العالية عند ظهورها، ولكن يعيب هذا النوع ثقل حجمه (مقاسا بالبايت) والذي يسبب إطالة وقت انتقال الصورة من خادم الويب إلى عميل الويب، أما صورة GIF تتميز بخفة حجمها (مقاسا بالبايت) إذا ما قورنت بنوع JPG ، ولكن يعيب هذا النوع قلة وضوحه إذا ما قورن بنوع JPG ، وبالإضافة إلى ذلك، تتميز صورة GIF بخاصتين:

1- خاصية الشفافية (Transparence) والتي تظهر الصورة دون خلفية، وبالتالي عند وضعها داخل صفحة الويب، يظهر ما وراؤها كخلفية لها.

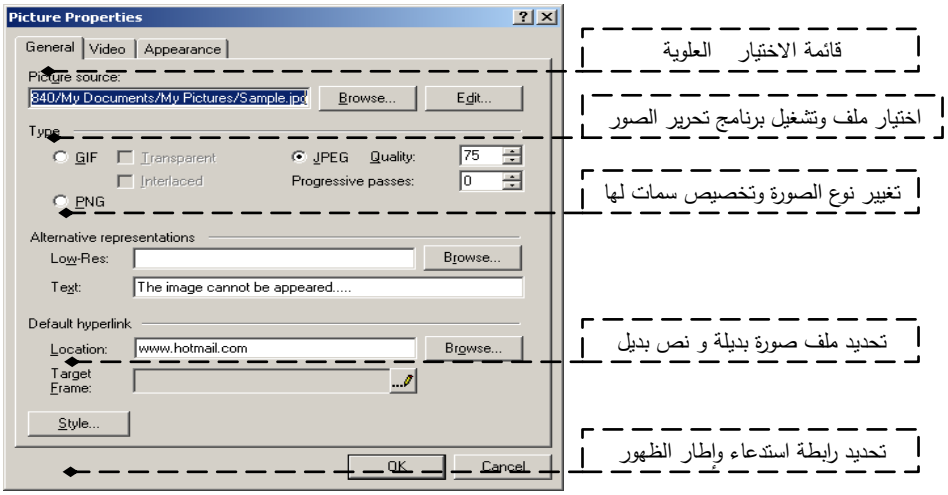
2- خاصية الحركة (Animation)، حيث يمكن تصميم إعلان (Banner) عن موضوع ما وهو عبارة عن مجموعة من الصور تظهر بشكل متتالي، وتخزين هذا الإعلان في صورة GIF، حيث يمكن تصميم هذا النوعية من الإعلانات بواسطة أحد البرامج المتخصصة في إنشاء Animated GIF.

تستخدم الصورة لعرض صورة أي كائن مثل صورة شخص، صورة منتج، شعار لشركة، وهكذا. ويمكن إضافة رابطة استدعاء (Hyperlink) للصورة. وبالتالي عندما يتم الضغط عليها يتم استدعاء صفحة ويب أخرى (أو موقع إنترنت آخر) والتي تظهر في نفس النافذة الحالية، أو في نافذة أخرى.

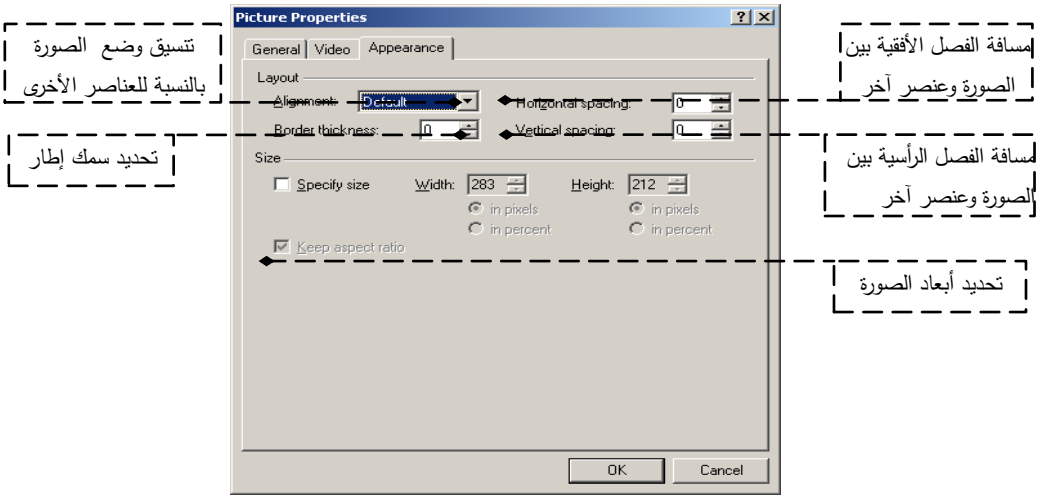
- *كيفية تصميم/تغيير صورة*
- عزيزي الدارس، لإدراج صورة داخل صفحة ويب مستخدماً برنامج MS FRONT PAGE ، اتبع الخطوات التالية:

الخطوة 1. قم باختيار قائمة Insert، ثم القائمة الفرعية Picture، ثم الأمر From File ، ثم قم باختيار الملف الذي يحتوي على الصورة المراد إدراجها من خلال نافذة Select File.

الخطوة 2. ولتغيير صفات الصورة، يتم وضع مؤشر الماوس على الصورة، ثم الضغط بالزر الأيمن، ثم اختيار أمر Picture Properties، عندئذ ستظهر نافذة حوارية والتي تتكون من ثلاث نوافذ فرعية: النافذة Appearance، والنافذة General، والنافذة Video، حيث يتم التنقل بينها من خلال قائمة الاختيار العلوية. وتوضح الأشكال رقم 2-20، و رقم 2-21 كلاً من نافذة General ونافذة Appearance على التوالي، حيث يحتوي هذان الشكلان على مغزى كل مكون من مكونات هاتين النافذتين، أما نافذة Video فهي خاصة بظهور لقطات الفيديو والتي لم نتعرض لها بعد.



شكل20 -2 : وصف مكونات نافذة General



شكل 2-21 : وصف مكونات نافذة Appearance

شفرة HTML المناظرة للصورة

يوضح شكل رقم 22-2 شفرة HTML المقابلة للصورة، حيث يحتوي الشكل على وصف رسومي مختصر لأهم علامات (Tags) الخاصة بلغة HTML لعرض صورة داخل صفحة ويب.



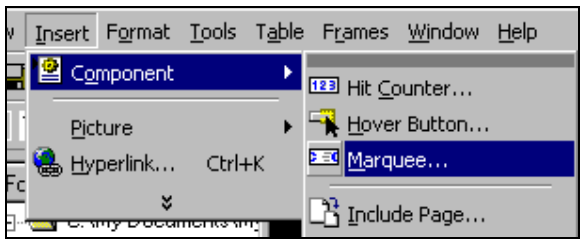

```
<p>
  <a href="www.hotmail.com">
    <img
        border="3" src="C:/images/Sample.jpg"
        alt="The image is not available"
        width="283" height="212"
        align="center" hspace="5" vspace="5"
    >
  </a>
</p>
```

شكل 22-2 : وصف رسومي لأهم علامات HTML لتصميم صورة

11. النصوص المتحركة (Marquees)

عزيزي الدارس النص المتحرك (Marquee) عبارة عن أداة تمكن مصمم صفحات الويب من تصميم نص متحرك يشبه لوحات الإعلانات أو شريط الأخبار في قنوات التلفزيون، ولإدارج مثل تلك النصوص المتحركة، اتبع الخطوات التالية:

الخطوة 1. قم بفتح القائمة Insert، ثم قم باختيار الأمر Component، ثم قم باختيار الأمر Marquee كما هو واضح في الشكل 23-2



شكل 23-2 : اختيار الأمر Marquee

الخطوة 2. وعند تنفيذ الأمر السابق سيظهر لك الصندوق الحواري الموضح في شكل 2-24، ثم اكتب النص الذي تريده أمام الاختيار Text، ولاحظ أنه يوجد عدة اختيارات في الصندوق الحواري السابق والتي يمكن تلخيص استخدامها هكذا:

- يستخدم الاختيار Direction لتحديد اتجاه النص من اليمين أو اليسار.
- يستخدم الاختيار Speed لتحديد سرعة النص.
- يستخدم الاختيار Behavior لتحديد شكل الحركة هل هي Scroll (أي مرور)، أو Slide (أي انزلاق) ، أو Alternate (أي تبادل).

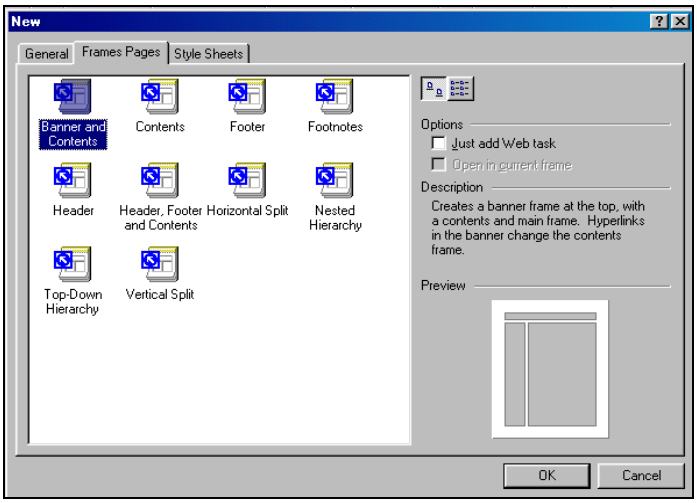


شكل 24-2 : نافذة خصائص النص المتحرك (Marquee)

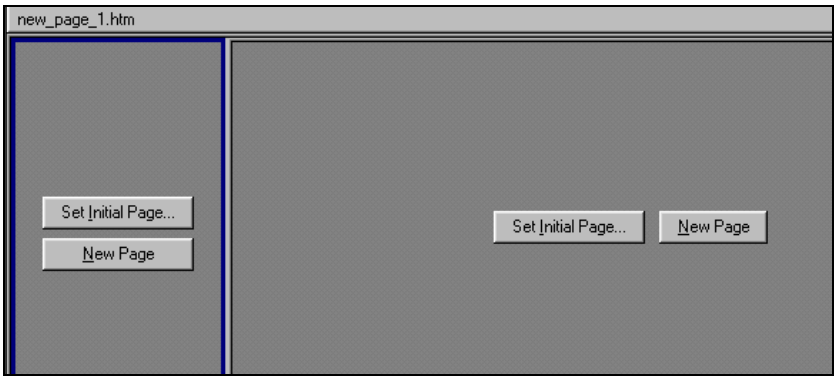
12. التعامل مع الإطارات (Frames)

إن معظم صفحات مواقع الويب يجب أن تحتوي على جزأين (أو أكثر): جزء ثابت وجزء متغير، حيث يمكن تخصيص الجزء الثابت لعرض أزرار التجول ، ويستخدم الجزء المتغير لعرض صفحات الموقع عندما يضغط المستخدم على أحد أزرار التجول، ولعمل ذلك نستخدم الإطارات (Frames) لتقسيم صفحة الويب إلى أجزاء، حيث يتم إنشاء صفحة إطارات تستخدم كوعاء لاحتواء هذه الإطارات، ثم يتم تصميم/إدراج صفحة ويب داخل كل إطار، ولتصميم صفحة تتكون من إطارين، اتبع الخطوات التالية:

الخطوة 1. لإنشاء صفحة الإطارات، قم بفتح قائمة File ، ثم قم باختيار الأمر New، ثم قم باختيار الأمر Page وذلك لإنشاء صفحة جديدة، وعند تنفيذ هذا الأمر سيظهر لك الصندوق الحواري الموضح في شكل 25- 2.

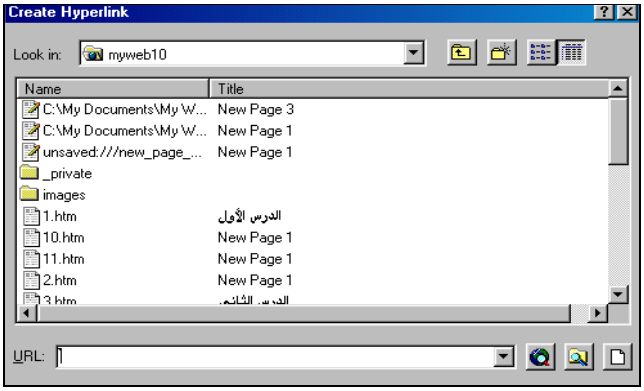


الخطوة 2. لاحظ أنه يوجد ثلاثة اختيارات داخل هذا الصندوق الحواري تؤدي إلى فتح ثلاث نوافذ، اضغط على الاختيار الثاني Frames Pages، ولإنشاء صفحة إطارات تحتوي على إطارين فقط، قم باختيار نوع الملف Contents، ثم اضغط على الزر Ok، عندئذ ستظهر لك صفحة الإطارات الجديدة كما هو واضح في شكل 26-2.



شكل 26 -2 : صفحة إطارات جديدة

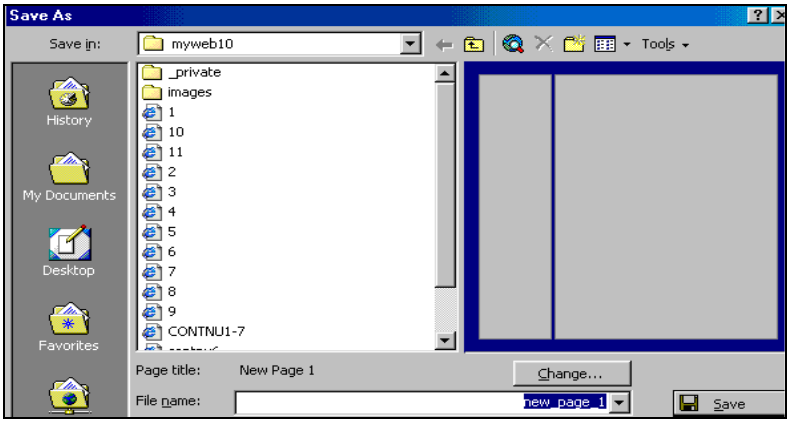
الخطوة 3. إذا كنت تريد أن تدرج صفحة ويب موجودة بالفعل في الإطار الأول (الإطار الأيمن)، انقر على الزر Set Initial Page... عندئذ سيظهر لك الصندوق الحواري الموضح في شكل 27-2 لاختيار صفحة الويب، أو يمكن أن تكتب اسم الصفحة التي تريدها أمام الاختيار URL، لاحظ: يجب أن يكون محتوى هذه الصفحة أزراراً تجول في الموقع والتي تمكن زائر موقعك من التجول داخله بالضغط على أحد أزرار هذه الصفحة.



شكل 27-2: نافذة اختيار ملف ويب

الخطوة 4. أما إذا كنت تريد أن تنشئ صفحة ويب جديدة داخل الإطار الثاني (الإطار الأيسر)، قم بالنقر على الزر New Page ، عندئذ ستظهر لك الصفحة الجديدة داخل الإطار الثاني لكي تبدأ في تصميمها.

الخطوة 5. والآن يجب حفظ ملفات صفحات الويب الجديدة، ولذلك قم بالذهاب إلى القائمة File ، ثم اضغط على الأمر Save as عندئذ سيظهر الصندوق الحواري الموضح في شكل 2-28 طالباً منك اسماً لصفحة الإطارات الرئيسية (اكتب الاسم index أمام File name:)



شكل 28-2 : نافذة تخزين ملف الإطارات الرئيسي

الخطوة 6. ثم سيظهر لك صندوق حواري آخر طالبا منك تخزين الصفحة الجديدة التي قمت بتصميمها داخل الإطار الأول ، قم بكتابة اسم الملف كما فعلت في الخطوة السابقة (الاسم index_left مثلا).

13. النماذج الحوارية (Forms)

عزيزي الدارس، يستخدم النموذج لاستقبال بيانات من المستخدم ثم إرسالها إلى خادم الويب ليتم عمل المعالجة عليها، ثم إرسال النتائج إلى المستخدم مرة ثانية، ولذلك تعتبر النماذج هي أهم الوسائل التي تستخدم لاستقبال بيانات مدخلة بواسطة المستخدم ولأن محتوى شكل البيانات يختلف من عدة أوجه؛ فإنه يوجد هناك أشكالاً مختلفة من حقول الإدخال والتي تستخدم بناء على شكل البيانات المراد استقبالها.

كيفية تصميم/إدراج عناصر النماذج

لتصميم نموذج مستخدماً برنامج MS FRONT PAGE ، يتم اختيار قائمة Insert ، ثم اختيار القائمة الفرعية Form، ثم اختيار أحد أوامر هذه القائمة التي تؤدي إلى إدراج أحد حقول إدخال البيانات المختلفة. وفيما يلي سيتم شرح هذه الأوامر:

الأمر Form : إدراج نموذج جديد

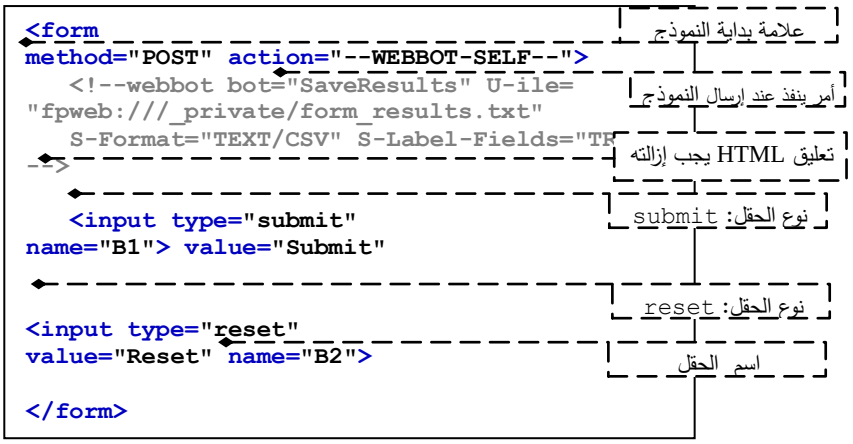
يستخدم هذا الأمر لإدراج نموذج جديد داخل صفحة الويب الحالية، يحتوي هذا النموذج على زرين (كما هو مبين في شكل رقم 29–2):

1. الزر Submit: عندما يتم الضغط على الزر Submit، يتم إرسال محتويات النموذج إلى خادم الويب.
2. الزر Reset: عندما يتم الضغط على الزر Reset، يتم إعادة حقول النموذج



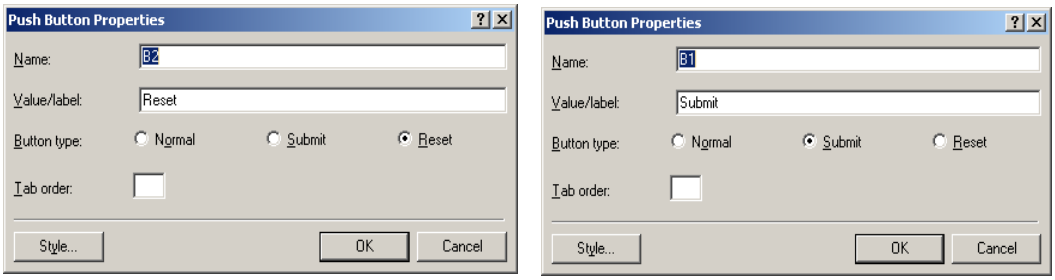
شكل 29-2 : نموذج يحتوي على الزر Submit والزر Reset

يمكن رؤية شفرة HTML المقابلة لهذا النموذج، وذلك بالضغط على رؤية HTML (أسفل نافذة البرنامج). يوضح شكل رقم 30–2 شفرة HTML المناظرة للنموذج السابق، وشرح مختصر لكل من سمات النموذج و سمات حقول النموذج.



شكل 30- 2: وصف رسومي لأهم علامات HTML لإنشاء نموذج

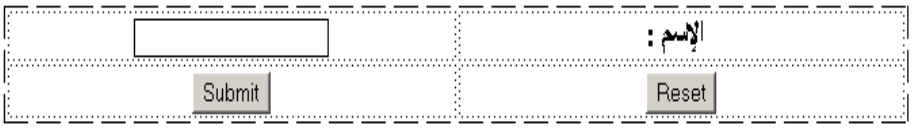
يمكن تغيير سمات كل من الزر Submit و الزر Reset (الاسم، النوع، القيمة/العنوان)، وذلك بالضغط المزدوج على أحدهم، وعندئذ ستظهر نافذة حوارية تمكنك من عمل ذلك (كما هو مبين في شكل 31 – 2).



أمر One-Line Text Box : إدراج حقل أو إدخال سطر

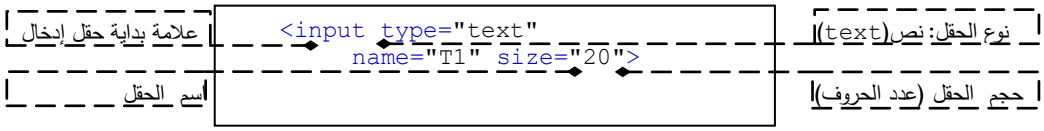
يستخدم هذا الأمر لإدراج حقل أو إدخال سطر، والذي يمكن المستخدم من أن يكتب سطراً واحداً داخل هذا الحقل، ويعتمد حجم هذا السطر على الحجم المحدد له، فعلى سبيل المثال، يمكن استخدام هذا الحقل لاستقبال الاسم الأول للمستخدم، أو الاسم الأخير، أو رقم التليفون. ويوضح شكل رقم 32–2 نموذجاً يحتوي على حقل إدخال سطر، والذي يستقبل اسم المستخدم. ولتصميم شكل رقم 32–2، يجب عليك عزيزي القارئ أن تلاحظ الملاحظات التالية:

- إدراج جدول يحتوي على عموديين وصفَّيْنِ لتنسيق عناصر النموذج .
- تم تخصيص قيمة صفر لحدود هذا الجدول لكي لا تظهر حدوده للمستخدم.
- تم وضع الجدول داخل النموذج وليس العكس.

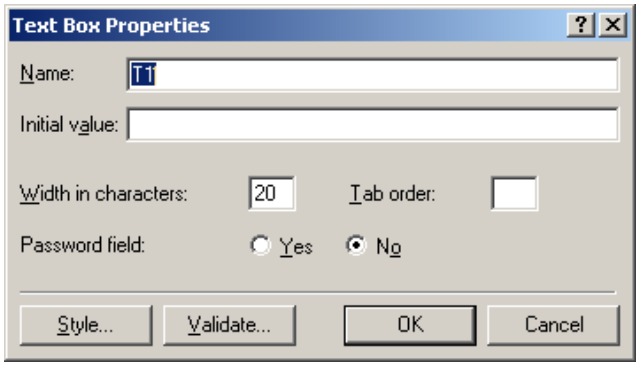


شكل 32-2 : نموذج يحتوي على حقل إدخال سطر

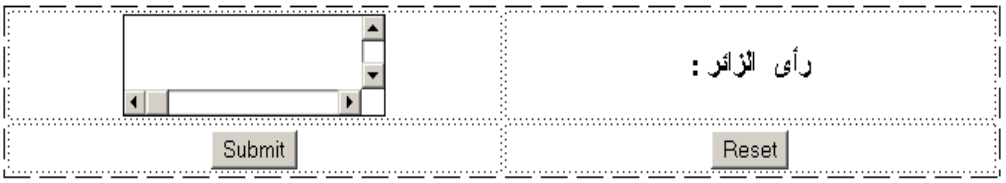
يوضح شكل رقم 33–2 شفرة HTML المناظرة لحقل إدخال سطر واحد وسمات هذا الحقل، كما يمكنك تغيير سمات هذا الحقل عن طريق الضغط المزدوج عليه في رؤية Normal، وعندئذ ستظهر شاشة الحوار الموضحة في شكل رقم 3



شكل 2-33 : وصف رسومي لأهم علامات HTML لحقل إدخال سطر



شكل 2-34 : نافذة تغيير سمات حقل إدخال سطر



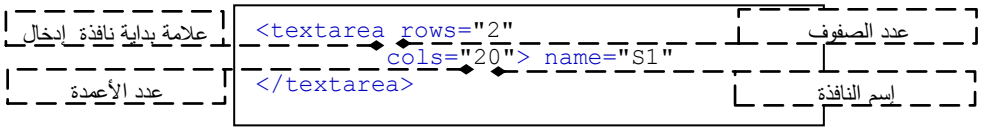
شكل 2-35 : نموذج يحتوي على حقل نافذة إدخال

أمر Scrolling Textbox : إدراج نافذة إدخال

يستخدم هذا الأمر لإدراج نافذة إدخال، والتي تسمح بإدخال عدة سطور، فعلى سبيل المثال يمكن أن تستقبل عنوان المستخدم، أو رأيه في الموقع، ويعتمد حجم البيانات التي يمكن استقبالها على عدد صفوف وعدد أعمدة النافذة حيث تتميز هذه النافذة باحتوائها على أسهم تصفح رأسية و أفقية (Vertical & Horizontal Scrolling Bars) والتي تمكن المستخدم من تصفح المحتويات غير الظاهرة للمستخدم بالضغط على هذه

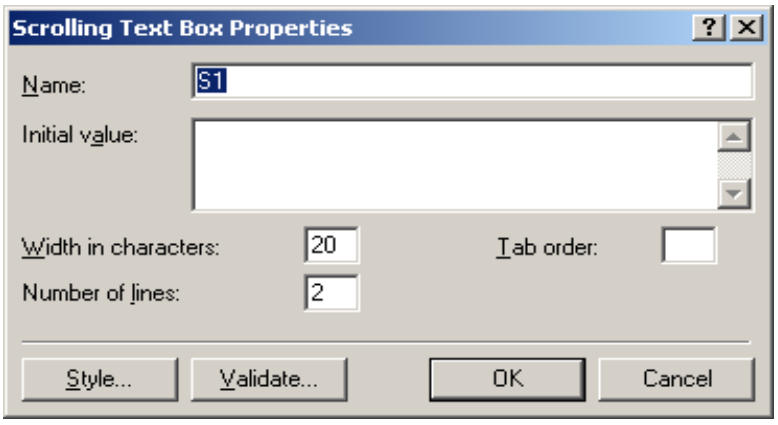
الأسهم. يوضح شكل رقم 2-35 نموذجاً يحتوي على نافذة إدخال والتي تستخدم لاستقبال رأي المستخدم في الموقع.

ويوضح شكل رقم 2-36 شفرة HTML المقابلة لنافذة الإدخال، ووصف لأهم سمات هذا العنصر الهام.



شكل 2-36 : وصف رسومي لأهم علامات HTML لحقل نافذة إدخال

يمكن تغيير سمات نافذة الإدخال وذلك بالضغط على رؤية Normal، ثم الضغط المزدوج على نافذة الإدخال، وعندئذ ستظهر نافذة الحوار المبينة في شكل رقم 2-37، حيث يمكن تغيير سمة الاسم، وعدد الصفوف، وعدد الأعمدة، كما يمكن تخصيص نص ابتدائي يظهر للمستخدم عند بداية ظهور النموذج.



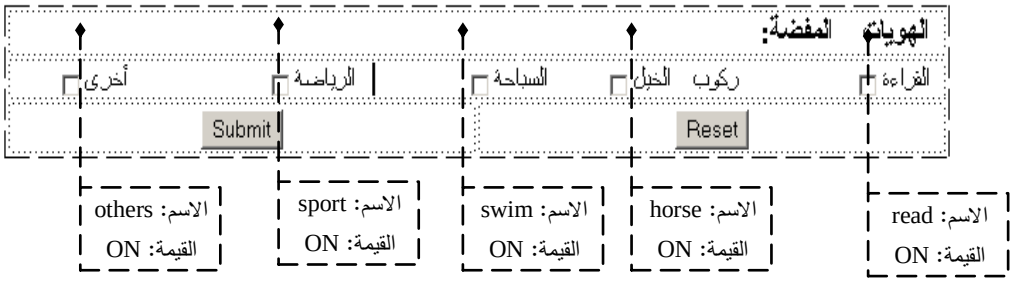
شكل 2-37 : نافذة تغيير سمات حقل نافذة إدخال

أمر Check Box : إدراج حقل تأشير.

عزيزي الدارس، يستخدم هذا الأمر لإدراج حقل تأشير، والذي يمكن المستخدم من اختيار عدة اختيارات من مجموعة اختيارات أكبر، فعلى سبيل المثال عندما نريد أن

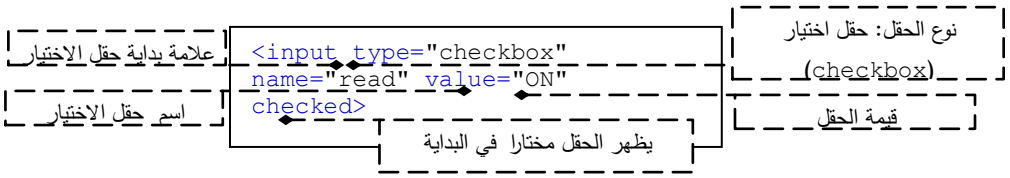
نستقبل هوايات المستخدم، يتم إدراج مجموعة من حقول التأشير للهوايات المختلفة كما هو واضح في شكل رقم 2-38 (لاحظ سمات الحقول: الاسم والقيمة حيث يوجد لكل حقل تأشير اسم مختلف عن الآخر). وعندما يختار المستخدم الهوايات المفضلة لديه، يسند بشكل تلقائي لكل حقل تأشير تم اختياره القيمة "ON"، ولم يسند لحقول التأشير الباقية قيمة (التي لم يتم اختيارها). فمثلا إذا اختار المستخدم كلاً من هواية القراءة وهواية السباحة، ثم ضغط على زر Submit، سيتم استقبال القيم التالية عند خادم الويب:

Swim=ON , Read=ON

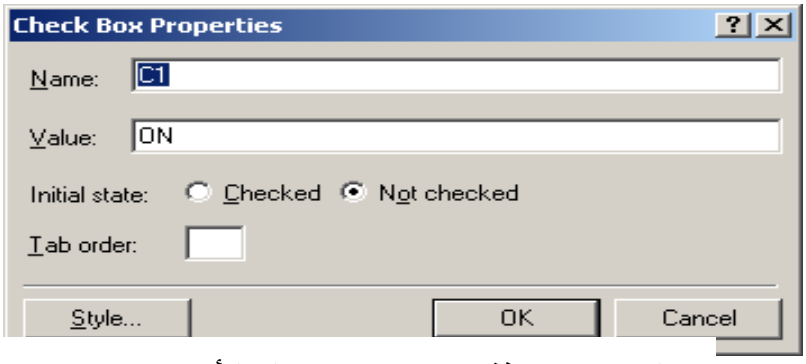


شكل 2-38 : نموذج يحتوي على حقول تأشير بأسماء مختلفة وقيمة واحدة

يوضح شكل رقم 2-39 شفرة HTML المقابلة لحقل التأشير وأهم السمات الخاصة به، كما يمكنك من تغيير سمات حقل التأشير وذلك باختياره، ثم الضغط المزدوج عليه، وعندئذ ستظهر نافذة الحوار المبينة في شكل رقم 2-40.



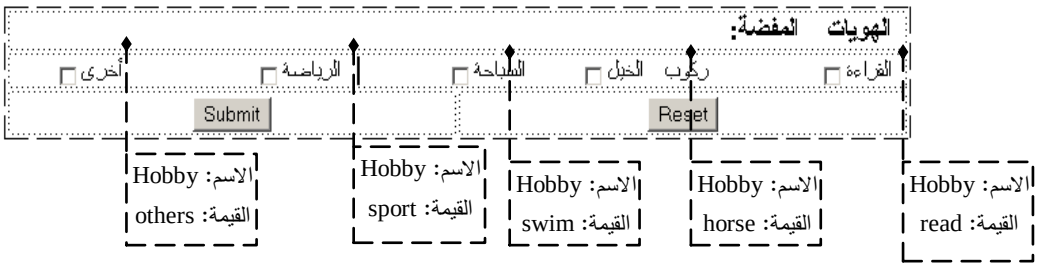
شكل 2-39 : وصف رسومي لأهم علامات HTML لحقل التأشير



شكل 40-2: نافذة تغيير سمات حقل التأشير

لاحظ عزيزي الدارس أنه يمكن إعطاء اسم واحد لكل حقول التأشير ولكن بقيم مختلفة (وليست القيمة ON)، فعلى سبيل المثال، يمكن تخصيص اسم Hobby الذي يعني هواية في المثال السابق لكل حقول التأشير و تخصيص قيمة Read لحقل القراءة، وتخصيص قيمة Swim لحقل السباحة، وهكذا (كما هو واضح في شكل رقم 41-2). وعند اختيار المستخدم لكل من هواية القراءة والسباحة، وإرسال بيانات هذا النموذج، سيتم استقبال القيم التالية عند خادم الويب:

```
Hobby = read , swim
```



شكل 41-2 : نموذج يحتوي على حقول تأشير باسم واحد (Hobby) وقيم مختلفة

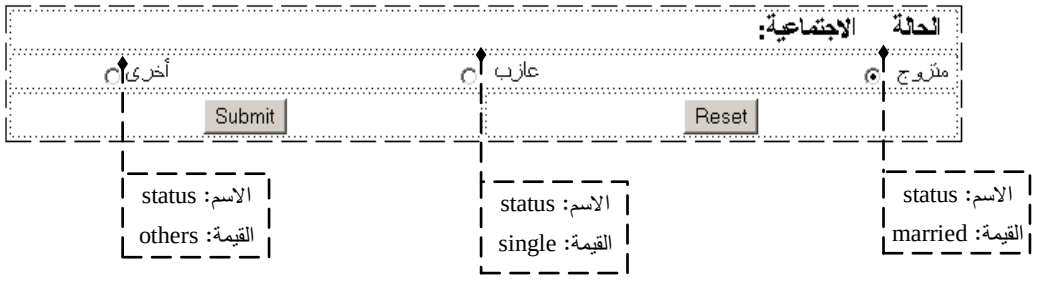
أمر Radio Button : إدراج زر الاختيار.

عزيزي الدارس، يستخدم هذا الأمر لإدراج زر اختيار والذي يمكن المستخدم من اختيار شئ وحيد من عدة اختيارات، فعلى سبيل المثال عندما نريد أن نستقبل الحالة

الاجتماعية للمستخدم، يتم إدراج نموذج يحتوي على ثلاثة أزرار اختيار: متزوج، عازب، وأخرى، فإذا اختار المستخدم أحد هذه الأزرار يتم إلغاء الزر الذي كان مختاراً من قبل ، وبالتالي نضمن اختيار شئ واحد من عدة اختيارات.

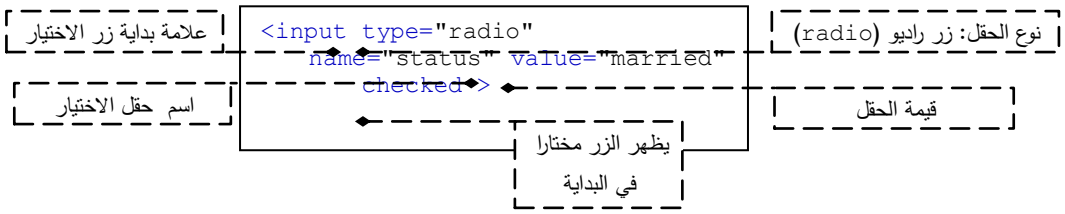
يجب إسناد اسم واحد لجميع أزرار الاختيار ولكن بقيم مختلفة، فعلى سبيل المثال تم تخصيص اسم Status لجميع أزرار الاختيار كما هو واضح في شكل رقم 42-2 ، وإعطاء قيمة مختلفة لكل زر اختيار (متزوج: Married، عازب: Single، أخرى: Others) ، فإذا اختار المستخدم متزوجاً وإرسل النموذج، سيتم استقبال التالي عند خادم الويب.

```
Status= "married"
```

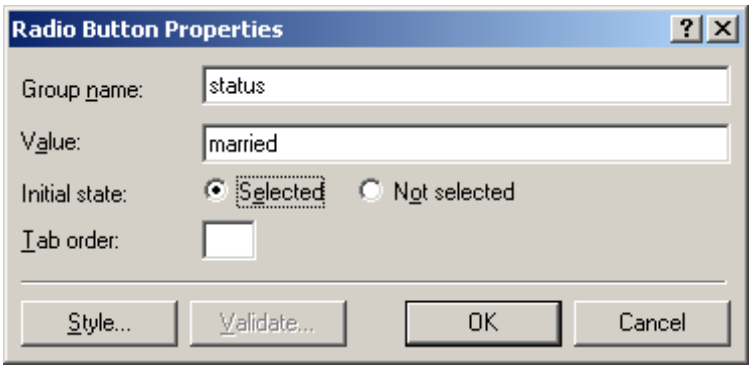


شكل 42-2 : نموذج يحتوى على حقول اختيار

يوضح شكل رقم 43-2 شفرة HTML المقابلة لحقل زر الاختيار و أهم السمات الخاصة به.



شكل 43-2: وصف رسومي لأهم علامات HTML لحقل زر الاختيار



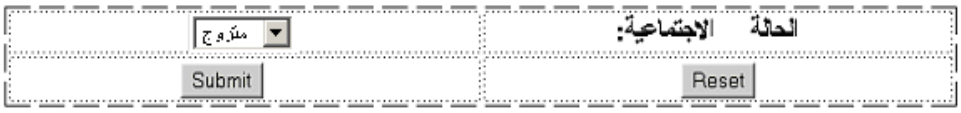
شكل 44-2 : نافذة تغيير سمات حقل زر الاختيار

ولإمكانية تغيير سمات زر الاختيار، يتم اختيار الزر ثم الضغط المزدوج عليه داخل رؤية Normal، وعندئذ تظهر نافذة الحوار المبينة في شكل 44-4 والتي تمكن المبرمج من تغيير هذه السمات.

أمر Drop Down Menu: إدراج قائمة اختيار منسدلة

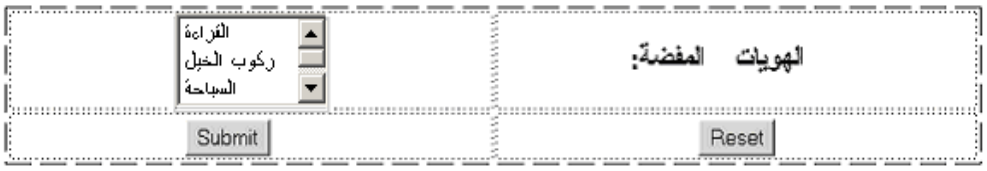
يستخدم هذا الأمر لإدراج قائمة اختيار منسدلة والتي تمكن المستخدم من اختيار شئ واحد (أو عدة أشياء) من مجموعة من الاختيارات، يعتبر هذا الحقل من أهم حقول الإدخال والتي تستخدم بكثرة في تصميم النماذج، ومن أمثلة استخدامها:

1- اختيار واحد من عدة اختيارات، و بالتالي يمكن استخدامها مثل أزرار الاختيار (Radio Buttons)، يوضح شكل رقم 45-2 إعادة لمثال الحالة الاجتماعية باستخدام قائمة الاختيار المنسدلة.



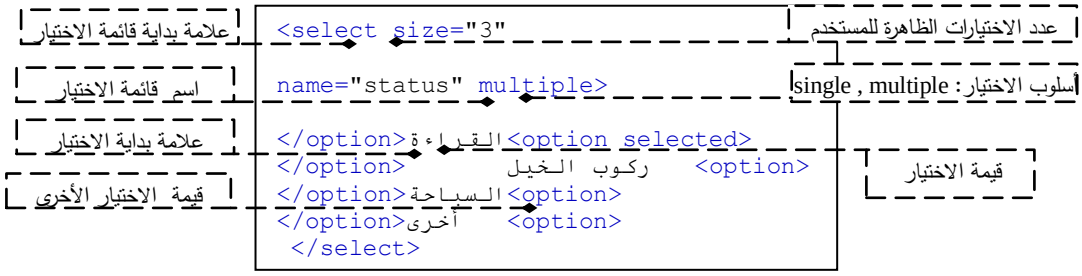
شكل 45-2 : نموذج يحتوي على حقل قائمة الاختيار المنسدلة لاختيار عنصر من عدة عناصر

2- اختيار عدة اختيارات من مجموعة أكبر من الاختيارات، وبالتالي يمكن استخدامها مثل حقول التأشير Check Boxes. يوضح شكل رقم 2-46 إعادة لمثال الهوايات باستخدام قائمة الاختيار المنسدلة.



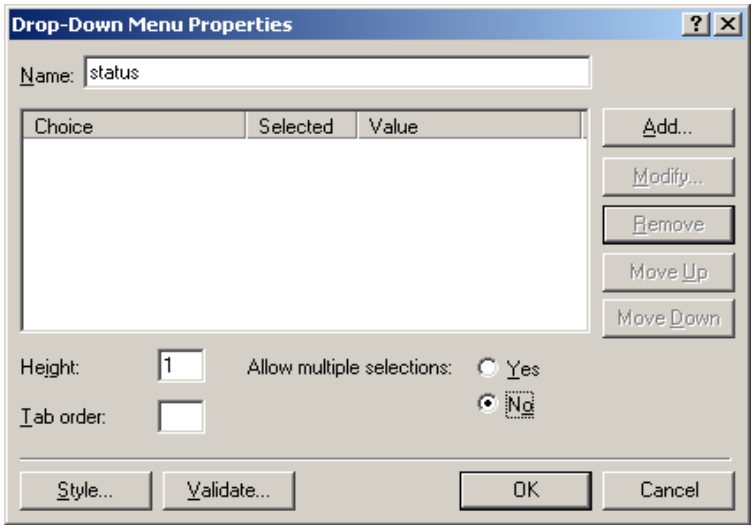
شكل 2-46 : نموذج يحتوي على حقل قائمة الاختيار المنسدلة لاختيار عدة عناصر من مجموعة أكبر

يوضح شكل رقم 2-47 شفرة HTML المقابلة لحقل قائمة الاختيار المنسدلة وأهم السمات الخاصة بها.



شكل 2-47 : وصف رسومي لأهم علامات HTML لحقل قائمة الاختيار المنسدلة

يمكن إضافة وتعديل الاختيارات المتاحة داخل قائمة الاختيار المنسدلة وتعديل السمات الخاصة بها، عن طريق اختيارها داخل رؤية Normal ثم الضغط المزدوج عليها، وعندئذ تظهر نافذة الحوار المبينة في شكل رقم 2-48 والتي تمكن المبرمج من عمل ذلك.



شكل 48-2: نافذة تغيير سمات حقل قائمة الاختيار المنسدلة

عزيزي الدارس لاحظ التالي:

- أنه يجب تخصيص اسم لقائمة الاختيار المنسدلة.
- يمكن إعطاء قيمة لكل اختيار (عنصر) داخل القائمة (قيمة رقمية مثلا)، وعندما يختار المستخدم عنصراً محدداً، ترسل هذه القيمة إلى خادم الويب بدلا منه، وإذا لم يعطي قيمة لكل عنصر، يرسل العنصر نفسه إلى خادم الويب.

أسئلة تقويم ذاتي



1. ما فائدة استخدام النماذج؟
2. اذكر خمسة أنواع من حقول النماذج مع ذكر وظيفة كل منها.
3. صمم نموذجاً مستخدماً برنامج MS FRONT PAGE يحتوي على البيانات التالية: الاسم، العنوان، النوع (ذكر، أنثى)، الهواية (كرة القدم، سباحة، ركوب الخيل)، بشرط أن تستخدم أربعة حقول مختلفة، ويجب أن تسمي الحقول أسماء ذات معنى.
4. اكتب شفرة HTML للنموذج الذي تم تطويره في المثال السابق.
5. ما الفرق بين حقل إدخال سطر و حقل نافذة الإدخال؟
6. صمم نموذجاً يستقبل رأي زائر في موقعك.
7. ما الفرق بين زر الاختيار، وزر التأشير؟

الخلاصة

في هذه الوحدة عزيزي الدارس، قمنا بدراسة عميل الويب (Web Client)، وخادم الويب (Web Server)، والأوامر النصية التي تعالج بهما، وعندما تعالج هذه الأوامر على عميل الويب، أطلقنا عليها اسم "البرمجـة لدى العميل" (Client-Side Programming) الذي هو محور اهتمامنا خلال هذه الوحدة، وعند معالجتها على خادم الويب نطلق عليـها اسم "البرمجة لدى الخادم" (Server-Side Programming) الذي سيكون موضوع الوحدة الخامسة

لمحة مسبقة عن الوحدة التالية

عزيزي الدارس، في وحدتنا التالية سوف نتحدث عن مواقع الإنترنت و أوامرها النصية (Scripting Code)، والتي يمكن معالجتها على كل من عميل الويب (Web Client)، وعلى خادم الويب (Web Server)،

إجابات التدريبات

```
<table border="1" width="100%">
                                    <tr>
<td width="50%" align="center">     الرقم
الجامعى</td>
<td width="50%" align="center">     اسم الطالب</td>
                                    </tr>
                                     <tr>
                                  <td width="50%"
                              align="center">43666666</td>
<td width="50%" align="center">     على ال</td>
                                    </tr>
                                     <tr>
                                  <td width="50%"
                               align="center">5656567</td>
<td width="50%" align="center">     حسن
البصر</td>
                                    </tr>
                                     <tr>
                                  <td width="50%"
                               align="center">6756577</td>
<td width="50%" align="center">     ماجد العلى</td>
                                    </tr>
```

المراجع

[1] عبدالكريم العقيل، "المخطط الأزرق لبناء صفحات ويب ديناميكية"، مكتبة جرير، الرياض، 2001م.

[2] د. أحمد أحمد شعبان دسوقى، أ. د. السيد محمود الربيعي، "تقنيات وبرمجة الإنترنت" مكتبة الرشد، الرياض، 2004م.

[3] د. عبد القادر عبد الله الفنتوخ، "الإنترنت للمستخدم العربي"، مكتبة العبيكان، الرياض، 2001م.

[4] د. إبراهيم فتحي معوض، "أساسيات برمجة الإنترنت"، معهد العالمية للحاسب والتقنية، 2005م.



الوحدة الثالثة

البرمجة لدى العميل

(Client-Side Programming)

الجزء الأول

محتويات الوحدة

الموضوع	الصفحة
المقدمة	73
تمهيد	73
أهداف الوحدة	74
1. إ ضافة شفرة JavaScript إلى مستند HTML	75
2. لغة JavaScript: نظرة عامة	79
1.2 نوافذ الإدخال والإخراج (I/O Dialogues)	79
2.2 المتغيرات وأنواع لغة JavaScript	83
3.2 التعابير الحسابية (Arithmetic Expressions)	84
4.2 تعابير المقارنة (Comparison Expressions)	85
5.2 التعابير المنطقية (Logical Expressions)	86
6.2 العبارة الشرطية IF …Else	87
7.2 عبارة التكرار For	87
8.2 عبارة التكرار While	89
9.2 تعريف الدوال (Functions)	90
الخلاصة	92
لمحة مسبقة عن الوحدة التالية	92
إجابات التدريبات	93
المراجع	94

المقدمة

تمهيد

عزيزي الدارس، عادة ما تشتمل مواقع الإنترنت على أوامر نصية (Scripting Code)، والتي يمكن معالجتها على كل من عميل الويب (Web Client)، وعلى خادم الويب (Web Server)، وعندما تعالج هذه الأوامر على عميل الويب، نطلق عليها اسم "البرمجـة لدى العميل" (Client-Side Programming) الذي سيكون محور اهتمامنا خلال هذه الوحدة، وعند معالجتها على خادم الويب نطلق عليـها اسم "البرمجة لدى الخادم" (Server-Side Programming) الذي سيكون موضوع الوحدة الخامسة. وهنا يتبادر إلى الذهن سؤال: ما هي أهمية تنفيذ هذه الأوامر لدى عميل الويب حيث كان يكفي أن يقوم خادم الويب بتنفيذها فقط؟ وللإجابة على هذا السؤال دعني أطرح لك مثالاً واقعياً لفهم أهمية البرمجة لدى العميل.

وفى نموذج مختلف آخر، إذا أحتاج أحد الموظفين إنهاء مهمة إدارية ما، يجب عليه أولاً أن يقوم بتنفيذ بعض الإجراءات البسيطة التي تأخذ بعض الوقت والجهد، والتي من شأنها أن تزيل من على عاتق الموظف الإداري بعض المجهود. إن هذا الأسلوب يشبه البرمجة لدى الخادم والعميل معاً لأن معالجة مهام تطبيق الويب ستنفذ لدى كل من الخادم والعميل، وذلك لتحسين أداء الموقع. فإذا أردنا مثلاً إرسال بيانات المستخدم إلى الخادم لتسجيلها في قاعدة البيانات، يجب على عميل الويب أولاً أن يتأكد من صحة هذه البيانات قبل إرسالها إلى خادم الويب، حيث يتأكد من صحة كل بيان من بيانات المستخدم وإظهار رسائل خطأ للمستخدم ، وعندئذ لا يتم إرسال بيانات فائضة إلى خادم الويب، وفى حال التأكد من صحة جميع البيانات، يتم إرسال البيانات إلى خادم الويب الذي سيقوم بتخزين البيانات في قاعدة البيانات دون التأكد من صحتها. ويمكننا القول بأن البرمجة لدى العميل تؤدي إلى تحسين أداء عمل الخادم ومن ثم تحسين أداء الموقع.

أهداف الوحدة

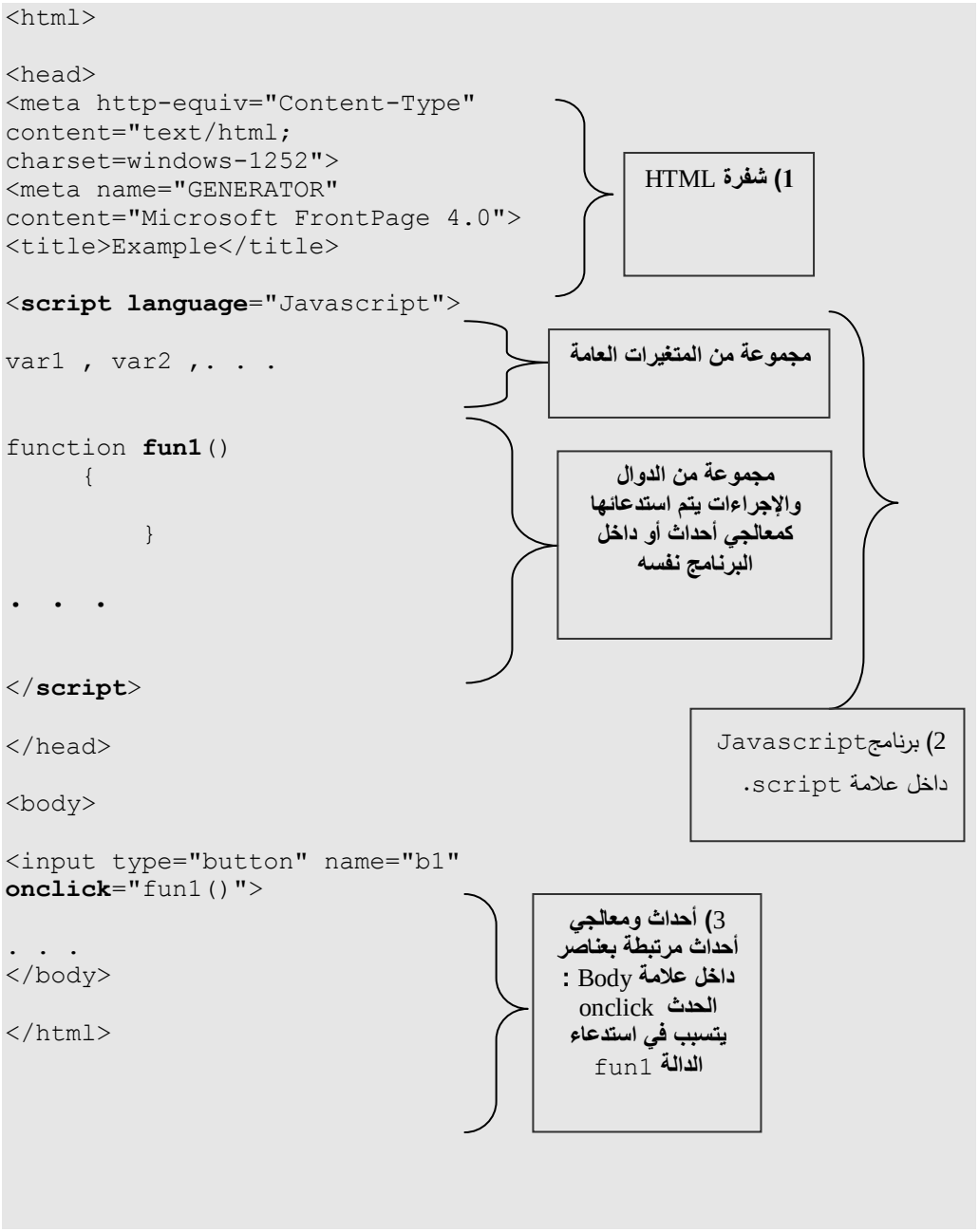


عزيزي الدارس، بنهاية دراسة هذه الوحدة ينبغي أن تكون قادرا على أن :

1. **تُعِرّف** إضافة شفرة JavaScript إلى مستند HTML.
2. **تبين** نوافذ الإدخال والإخراج.
3. **تحدد** المتغيرات وأنواعها في لغة JavaScript .
4. **توضّح** عبارات التحكم في لغة JavaScript.
5. **تتعامل** مع الدوال في لغة JavaScript.

1. إضافة شفرة JavaScript إلى مستند HTML

عزيزي الدارس، لتنفيذ البرامج الفرعية الخاصة بلغة JavaScript لدى عميل الويب، لابد من إضافة هذه البرامج إلى مستندات HTML (انظر شكل رقم 1-3)، ومن ثم يمكن أن ترتبط هذه البرامج بمجموعة من الأحداث (تصدر بواسطة المستخدم أو غيره) مثل الضغط على الزر الأيسر للماوس، وعندئذ يتم استدعاء هذه البرامج وتنفيذها بواسطة محرك لغة JavaScript المتواجد داخل عميل الويب




```
<html>

<head>
<meta http-equiv="Content-Type"
content="text/html;
charset=windows-1252">
<meta name="GENERATOR"
content="Microsoft FrontPage 4.0">
<title>Example</title>

<script language="Javascript">

var1 , var2 ,. . .


function fun1()
     {

         }


. . .


</script>

</head>

<body>

<input type="button" name="b1"
onclick="fun1()">

. . .
</body>

</html>
```

شكل 1-3 : وصف لمكونات مستند HTML بعد إضافة شفرة JavaScript

من شكل 1-3 يتضح لنا أن مستند HTML الذي يحتوي على برامج تعالج لدى عميل الويب يتكون من المكونات التالية:

1) شفرة HTML المتعارف عليها وهى المسئولة عن ظهور محتويات الويب (مثل الجدول، الصور...إلخ) بشكل منسق.

2) شفرة JavaScript التي تتكون من مجموعة من الدوال والإجراءات المكتوبة بلغة JavaScript، ويمكن أن تنقسم هذه الدوال والإجراءات بناءً على طريقة استدعائها إلى قسمين:

- قسم من الدوال والإجراءات يتم استدعاؤه عند حدوث أحداث مرتبطة بعناصر صفحة الويب، ويطلق على هذه الدوال والإجراءات Event Handlers (دوال معالجي الأحداث).
- قسم من الدوال والإجراءات يتم استدعاؤها بواسطة دوال أخرى داخل شفرة JavaScript.

3) أحداث مرتبطة مع عناصر صفحة الويب، حيث يكتب الحدث كسمة مع عنصر HTML . وقيمة هذه السمة هي اسم دالة معالجة الحدث (Event Handler)، وهي الدالة المعرفة بلغة Javascript داخل علامة Script.

لاحظ أن علامة Script تحتوي على مجموعة من السمات مثل سمة اللغة المستخدمة لكتابة البرامج الفرعية (language="Javascript")، ومن ثم يتم استدعاء محرك اللغة المناسب لتنفيذ هذه البرامج، سنكتفي في باقي أمثلة هذه الوحدة بكتابة كل من شفرة Javascript وعناصر HTML المرتبطة بالأحداث فقط وإلغاء شفرة HTML الباقية المتعارف عليها.

مثال 1-3: إلقاء التحية عند الضغط على زر.

يوضح الشكل رقم 2-3 مستند ويب و شفرة HTML المناظرة لها حيث يحتوي هذا المستند على شفرة كل من HTML المسئولة عن إظهار زر بعنوان "ألقي التحية" وعلى

شفرة Javascript التي تظهر رسالة التحية "السلام عليكم ورحمة الله" عند معالجتها لدى العميل نتيجة الضغط على هذا الزر.

```
<script language="Javascript">

function welcome()
          {
             alert("السلام عليكم ورحمة الله");
          }

</script>

<body>

<input type="button" value="ألقي التحية" name="B3"
nclick="welcome();">
```



شكل 2-3 : إلقاء التحية

لاحظ أن الدالة ( )Welcome (المعرفة داخل علامة Script) يتم استدعاؤها نتيجة حدوث الحدث Onclick المرتبط بالزر "ألقي التحية" (أي عندما يضغط المستخدم على هذا الزر)، ولاحظ أيضا أن الدالة ( )Welcome تستدعي الدالة ( )alert المسؤولة عن إظهار رسالة التحية (أحد دوال مكتبة لغة Javascript التي سنتعرف عليها خلال الموضوع التالي).

2. لغة JavaScript: نظرة عامة

عزيزي الدارس، تمثل لغة JavaScript أحد أهم اللغات النصية (Scripting Languages) التي تستخدم لكتابة برامج نصية تنفذ لدى عميل الويب، وذلك لكثرة برامج متصفحات الإنترنت (عملاء الويب) التي لديها القدرة على تنفيذ أوامر هذه اللغة مثل برنامج MS Internet Explorer ، وبرنامج Netscape Navigator، وذلك على عكس لغات نصية أخرى مثل لغة VBScript التي اشتهر استخدامها لكتابة برامج فرعية لدى الخادم.

إن لغة JavaScript تمثل لغة برمجة كاملة تحتاج إلى كتاب منفصل لشرحها بالتفصيل، ولكن سنكتفي في هذا الجزء بعرض أهم أوامر ودوال هذه اللغة التي نحتاجها لتطوير برامج فرعية جيدة لتعالج لدى عميل الويب، ومن يريد الاستزادة يمكنه أن يعود إلى أحد الكتب المختصة بشرح هذه اللغة أو بأحد منتديات الإنترنت المختصة في ذلك.

1.2 نوافذ الإدخال والإخراج (I/O Dialogues)

عزيزي الدارس، تحتوي مكتبة لغة JavaScript على العديد من الدوال التي تمكن المبرمج من تنفيذ العديد من المهام، وتعتبر الدالة ( )Alert والدالة ( )Prompt من أهم هذه الدوال، حيث تمكن الدالة ( )Alert المبرمج من عرض رسائل حوارية، و تمكن الدالة ( )Prompt المبرمج من استقبال بيانات من المستخدم. وفيما يلي سيتم شرح هاتين الدالتين لتطوير برامج تفاعلية تعمل على عميل الويب.

- **الدالة ( )Alert**

عزيزي الدارس، تستخدم هذه الدالة لإظهار رسالة داخل نافذة حوارية تحتوي على الرسالة الممررة لها، وإظهار زر بعنوان OK، يوضح الأمر التالي الصيغة العامة للدالة ()Alert حيث يمثل المعامل <Message> الرسالة التي ستظهر للمستخدم.

```
Alert( < Message> )
```

مثال 2-3: استخدام الدالة ( )Alert

يوضح المثال التالي (الموضح في شكل 3-3) استخدام الدالة ()Alert لإظهار الرسالة "السلام عليكم ورحمة الله".

```
Alert("السلام عليكم ورحمة الله")
```



شكل 3-3 : استخدام الدالة ( )Alert

تدريب (1)



تحتوي صفحة ويب على زر بعنوان "جمع رقمين"، وعند الضغط على هذا الزر يستدعى الدالة ( )Add المسؤول عن حساب مجموع رقمين، حيث يتم إظهار رسالة تطلب من المستخدم إدخال الرقم الأول، ثم رسالة تطلب إدخال الرقم الثاني، ثم تظهر الناتج داخل رسالة. عرف الدالة Add() .

- **الدالة ( )Prompt**

عزيزي الدارس، تستخدم الدالة ( )Prompt لإظهار نافذة حوارية تحتوي على رسالة وحقل إدخال (Text Box) وزرين (Buttons): الزر Ok والزر Cancel، ثم يضغط المستخدم على أحد هذه الأزرار، فإذا ضغط المستخدم على زر OK تعيد الدالة القيمة المدخلة في حقل الإدخال، أما إذا ضغط المستخدم على زر Cancel ، تعيد الدالة القيمة null.

يوضح الأمر التالي الصيغة العامة للدالة Prompt( ) حيث نمرر إليها الرسالة التي ستحث المستخدم على إدخال البيانات (Message)، والقيمة الافتراضية التي ستظهر في حقل الإدخال عند بداية ظهور النافذة (DefaultValue) مع العلم أن هذه القيمة اختيارية، فإذا لم تكتب هذه القيمة تظهر الكلمة "undefined" في حقل الإدخال، ثم نستقبل القيمة النصية العائدة من الدالة Prompt( ) داخل المتغير <Var>.

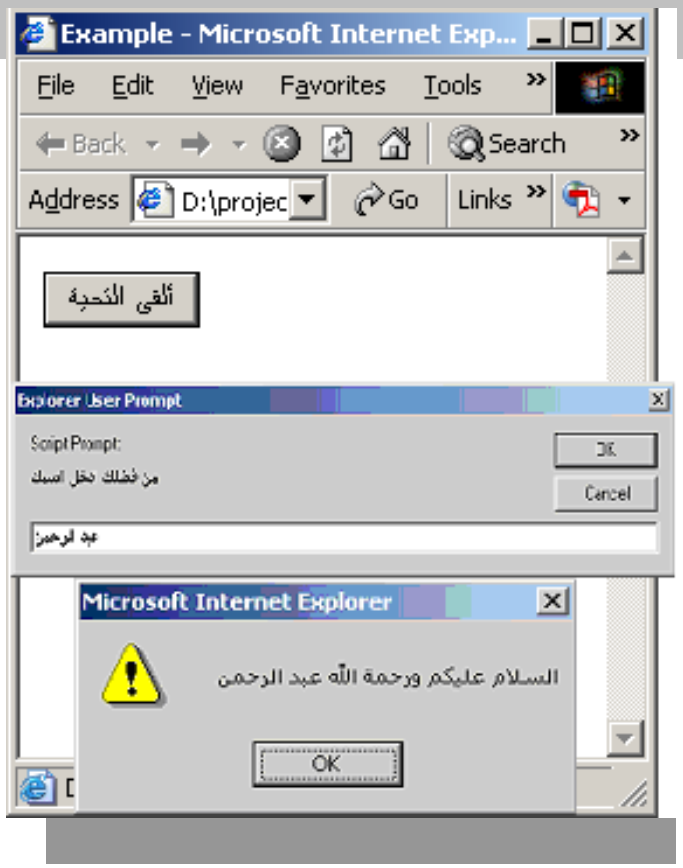
```
<Var > =  Prompt ( <Message> [ , Default Value ] )
```

مثال 3-3: استخدام الدالة Prompt( )

يوضح المثال التالي (الموضح في شكل 3-4) استخدام كل من الدالة Alert() والدالة Prompt( ) لتعديل برنامج المثال السابق (مثال إلقاء التحية عند الضغط على زر "ألقي التحية") حيث نحث المستخدم أولا أن يدخل اسمه ثم نظهر تحية موجهة له بواسطة لصق الاسم المدخل مع التحية "السلام عليكم ورحمة الله"، وذلك بواسطة استخدام معامل لصق النصوص، يوضح المثال أوامر الدالة Welcome() المعدلة ومخرجات البرنامج فقط.

```
<script language="Javascript">

function welcome()
   {
var uName ;
      uName = prompt("من فضلك ادخل اسمك");
      alert("  السلام عليكم ورحمة الله " + uName  );
   }
</script>
```



شكل 4-3 : استخدام الدالة Prompt( )

أسئلة تقويم ذاتي

صمم صفحة ويب تحتوى على زر بعنوان "أدخل بياناتك"، وعند الضغط على هذا الزر يستدعى إجراء يظهر الرسائل التالية:

- رسالة تلقى التحية على المستخدم: السلام عليكم.
- رسالة تطلب الاسم الأول للمستخدم.
- رسالة تطلب الاسم الأخير للمستخدم.
- رسالة تظهر له الاسم كاملا كما أدخل.

2.2 المتغيرات وأنواع بيانات لغة Javascript

عزيزي الدارس، المتغيرات (Variables) هي أسماء لأماكن في الذاكرة تحتوي على قيم، فعلى سبيل المثال، يمكن أن تستخدم المتغير myAge ليخزن القيمة 36، ومن الممكن أن تستخدم المتغيرات لإجراء أنواع مختلفة من العمليات الحسابية أو العمليات الأخرى، وللإعلان عن متغير لاستخدامه فيما بعد داخل أوامر البرنامج، نستخدم الكلمة المحجوزة Var، ثم نتبعها باسم المتغير ثم الرمز (;) لإنهاء الأمر (لا حظ أن أوامر لغة JavaScript تنتهي بالرمز ;) كما هو واضح من الصيغة العامة التالية:

```
Var  <variableName1> [ , <variableName2> , .. ] ;
```

ويمكن إسناد قيمة أحد أنواع بيانات لغة Javascript إلى المتغير مستخدما معامل الإسناد (=)، حيث تحتوي لغة Javascript على أربعة أنواع من البيانات: الأرقام (Numbers)، النصوص (Strings)، القيم المنطقية (Boolean)، لا شئ (Null). ويوضح جدول 3-1 وصفاً مختصراً لهذه الأنواع.

جدول 1-3: أنواع بيانات لغة JavaScript

أنواع البيانات	الوصف
الأرقام (Numbers)	يمثل أي رقم مثل 17، 21.5.... إلخ.
النصوص (Strings)	يمثل أي قيمة نصية مثل العبارة "كيف حالك" مع العلم أنه يجب إحاطة النص بالرمز، " أو بالرمز و.
قيمة منطقية (Boolean)	يمثل القيمة true أو القيمة false.
لا شئ (Null)	الكلمة Null تعني لا شئ.

مثال 4-3: الإعلان عن المتغيرات

يوضح المثال التالي (شكل 5-3) الإعلان عن متغيران: المتغير myName والمتغير myAge، ثم تخزين نص ("Mohammed") داخل المتغير myName وتخزين الرقم 8 داخل المتغير myAge .

```
Var  myName , myAge ;
myName = "Mohammed" ;
myAge =8 ;
```

شكل 5-3 : الإعلان عن المتغيرات

3.2 التعابير الحسابية (Arithmetic Expression)

عزيزي الدارس، يستخدم التعبير الحسابي لحساب علاقة حسابية التي تربط مجموعة من المتغيرات الرقمية ومجموعة من الأرقام بمعاملات حسابية، حيث يمكن إسناد نتيجة معالجة هذه التعابير إلى متغير أو طباعته على الشاشة. يوضح الجدول رقم 2-3 وصفاً مختصراً عن هذه المعاملات وأسبقية تنفيذ كل منها، وإذا وجد معاملان لهم نفس

أولوية التنفيذ، تتم المعالجة من اليسار إلي اليمين، أما إذا أردت أن تغير أسبقية التنفيذ (تقدم معالجة بعض المعاملات عن الآخرين) ، يجب استخدام الأقواس ( ).

جدول 2-3 : وصف مختصر عن المعاملات الحسابية وأسبقية تنفيذها

المعامل	*الوصف*	*مثال*	*الأسبقية*
-	لإعطاء إشارة سالبة للرقم	- 5	1
* ، /	لحساب الضرب والقسمة	5*2/3	2
%	لحساب باقي القسمة	2%5	3
+ ، -	لحساب الجمع والطرح	5+2-1	4

كما يمكن استخدام المعامل ++ لزيادة المتغير بمقدار واحد، فعلى سبيل المثال يمكن زيادة المتغير X بمقدار واحد بواسطة الأمر X++، ويمكن استخدام المعامل -- لنقصان المتغير بمقدار واحد، فعلى سبيل المثال يمكن نقصان المتغير X بمقدار واحد بواسطة الأمر X-- .

4.2 تعابير المقارنة (Comparison Expressions)

عزيزي الدارس، تستخدم **تعابير** المقارنة لإجراء عملية مقارنة بين قيمتين رقميتين وإصدار نتيجة منطقية (صواب True أو خطأ False)، حيث يحتوي **تعبير** المقارنة على معاملات مقارنة مثل معامل أكبر من (>)، و معامل أصغر من (<) ، وقيم رقمية. ويوضح الجدول رقم 3-3 وصفاً مختصراً لهذه المعاملات .

جدول 3-3 : وصف مختصر عن معاملات المقارنة

المعامل	الوصف	مثال
>	يختبر إن كانت قيمة ما أكبر من قيمة أخرى	Y>X
<	يختبر إن كانت قيمة ما أصغر من قيمة أخرى	Y<X
>=	يختبر إن كانت قيمة ما أكبر أو تساوي قيمة أخرى	Y>=X
=<	يختبر إن كانت قيمة ما أصغر أو تساوي قيمة أخرى	Y=<X
==	يختبر إن كانت قيمة ما تساوي قيمة أخرى	Y==X
=!	يختبر إن كانت قيمة ما لا تساوي قيمة أخرى	Y!=X

5.2 التعابير المنطقية (Logical Expressions)

عزيزي الدارس، تستخدم التعابير المنطقية لإجراء عمليات منطقية والتي تحتوي على معاملات منطقية وعلى قيم منطقية، ثم تنتج أيضا قيماً منطقية (صواب true أو خطأ false). ويوضح الجدول رقم 4-3 وصف مختصر للمعاملات المنطقية وأسبقية تنفيذهم.

جدول 4-3: وصف مختصر عن المعاملات المنطقية وأسبقية تنفيذهم

المعامل	الوصف	مثال	الأسبقية
!	تعكس القيمة التي تدخل عليها، فإذا كانت True تجعلها False والعكس صحيح.	! X	1
&&	تنتج نتيجة صواب True فقط إذا كانت كلا القيمتين صواب True.	X && Y	2
||	تنتج نتيجة صواب True فقط إذا كانت إحدى القيمتين على الأقل صواب True.	X || Y	3

6.2 العبارة الشرطية IF. .Else

عزيزي الدارس، تستخدم هذه العبارة لتنفيذ مجموعة من الأوامر (الموجودة بعد الكلمة IF) إذا تحقق الشرط المصاحب لها (إذا كانت نتيجة اختباره صواب True) وتنفيذ مجموعة أخرى من الأوامر (الموجودة بعد الكلمة Else) إذا لم تتحقق نتيجة هذا الشرط (إذا كانت نتيجة اختباره خطأ False) ، مع العلم أن الجزء الخاص بالكلمة Else اختياري، أي يمكن كتابة جزء عبارة IF فقط. توضح الأوامر التالية الصيغة العامة لأمر IF..Else، حيث إن العبارة <condition> تمثل الشرط المراد التحقق من صحته، فإذا كانت نتيجته صواب (True) ستنفذ مجموعة الأوامر الأولى، أما إذا كانت نتيجة الشرط خطأ (False) ستنفذ مجموعة الأوامر الثانية.

```
If( <condition>)
    {
     <Set 1 of statements> ;
     }
Else
   {
     <Set 2 of Statements> ;
     }
```

7.2 عبارة التكرار For

عزيزي الدارس، تتيح لك عبارة التكرار For أن تنفذ مجموعة معينة من الأوامر عدداً محدداً من المرات، فعلى سبيل المثال ربما ترغب في إنشاء خمس مربعات نص على صفحة الويب، فبدلاً من كتابة الأوامر اللازمة لإنشاء مربع النص خمس مرات، يمكنك وضع هذه الأوامر داخل تكرار For حيث يتم تنفيذها عدد المرات الذي تريده.

وعند إنشاء عبارة For ستستخدم متغيراً يعمل كعداد (Counter) للتكرار، وسوف تحدد القيمة التي سيبدأ عندها العداد والشرط الذي يتحكم في تنفيذ أوامر التكرار، حيث يستمر تنفيذ التكرار إلى أن تصبح قيمة الشرط خطأ (False)، وتحدد أيضا قيمة الزيادة (أو قيمة النقصان) في كل دورة للتكرار. ستقوم لغة JavaScript بتنفيذ الأوامر مرة ثم تزيد (أو تقل) من قيمة العداد، وعندما تصل قيمة العداد إلى القيمة التي تجعل الشرط خطأ سينتهي تنفيذ التكرار وتبدأ لغة JavaScript في تنفيذ الأوامر التي تلي عبارة For. توضح الأوامر التالية الصيغة العامة لعبارة التكرار For.

```
For( <Counter> =  <Start> ;  <Condition> ; <Inc/Dec>)
{
        <Set  of statements> ;
}
```

مثال 5-3: حساب متتالية حسابية مستخدما التكرار For

يوضح المثال التالي (الموضح في شكل 6-3) كيفية استخدام التكرار For لحساب المتتالية الحسابية التالية وتخزين الناتج داخل المتغير S:

1 + 2 + 3 + 4 + …….. +100

```
Var  i , S;
S = 0 ;
For( i=1; i =< 100 ; i++)
    {
      S = S + i  ;
     }
```

شكل 6-3 : حساب متتالية حسابية مستخدما التكرار *For*

8.2 عبارة التكرار While

عزيزي الدارس، تستخدم التكرار While لمعالجة مجموعة من الأوامر طالما نتيجة اختبار شرط التكرار المصاحب صحيحة (True)، وتتوقف عن تكرار معالجة هذه الأوامر عندما تصبح نتيجة اختبار شرط التكرار خاطئة (False)، ويجب أن تلاحظ أن هذه العبارة تختبر أولا شرط التكرار ثم تنفذ الأوامر ، ومعنى ذلك ربما تكون نتيجة اختبار شرط التكرار أول مرة خاطئة وعندئذ لم يتم تنفيذ هذه الأوامر مطلقا. توضح الأوامر التالية الصيغة العامة لعبارة التكرار While.

```
While (<Condition>)
    {
        <Set of statements> ;
    }
```

مثال 6-3: حساب متتالية حسابية مستخدما التكرار While

يوضح المثال التالي (الموضح في شكل 7-3) كيفية استخدام التكرار While لحساب المتتالية الحسابية التالية وتخزين الناتج داخل المتغير S :

1 + 3 + 5+ 7 + …….. + 99

```
var  i , S ;
S = 0 ;
i = 1 ;
while(i < = 99)
{
S = S + i ;
i = i + 2 ;
}
```

9.2 تعريف الدوال (Functions)

عزيزي الدارس، تتكون الدالة (Function) من مجموعة من الأوامر والتي تقوم بأداء مهمة محددة عند استدعائها، حيث يمكن للدالة أن تستقبل معاملات (Parameters) عند استدعائها لإجراء المعالجة عليها. يتيح لك استخدام الدوال تنظيم البرنامج إلى مجموعات أصغر من البرامج الفرعية (الدوال) والتي تتميز بسهولة إدارتها بالإضافة إلى سهولة عملية اكتشاف أخطائها، هذا بالإضافة إلى إعادة استخدام كل من هذه الدوال أكثر من مرة في البرنامج الحالي أو في برامج أخرى. فعلى سبيل المثال، يمكن أن تكتب مجموعة من الأوامر لعرض رسالة تحذير عندما يدخل المستخدم بيانات غير صالحة في النموذج، و بدلاً من كتابة هذه الأوامر لعرض الرسالة في كل مرة تحتاج إليها، يمكنك أن تضعها داخل دالة ثم تستدعي هذه الدالة في كل مرة تحتاج إليها.

ويطلق على هذه النوعية من الدوال (Programmer Defined Functions)، أي دوال معرفة من قبل المبرمج وليست معرفة من قبل مطوري لغة البرمجة، حيث يتم تعريفها بواسطة المبرمج عند عدم وجود دالة جاهزة تؤدي المهمة المطلوبة، ولقد تعاملنا بالفعل مع كلا النوعين من الدوال، حيث عرفنا الدالة Welcome() سابقا (دالة معرفة من قبل المبرمج)، كما استخدمنا دوال مكتبة لغة Javascript الجاهزة مثل الدالة Alert() والدالة Prompt() (دوال معرفة من قبل مطوري اللغة). توضح الأوامر المبينة في شكل 8-3 الصيغة العامة لتعريف الدالة.

```
function <function_name>( [ <parameters> ] )
   {
      <Set of Statements> ;
   }
```

شكل 8- 3 الصيغة العامة لتعريف الدالة

تدريب (2)



تحتوي صفحة ويب على زر بعنوان "حجم أسطوانة"، وعند الضغط على هذا الزر يستدعي الدالة ( ) Volume المسؤولة عن حساب حجم أسطوانة، حيث يتم إظهار رسالة تطلب من المستخدم إدخال قيمة نصف قطر دائرة الاسطوانة، ثم ارتفاع الأسطوانة ، ومن ثم طباعة حجم أسطوانة ، مع العلم أن حجم أسطوانة يساوي: ط * نق2 * الارتفاع. عرف الدالة ( )Volume؟

أسئلة تقويم ذاتي



1) صمم صفحة ويب تحتوي على زر بعنوان "مساحة الدائرة"، وعند الضغط على هذا الزر يستدعي الدالة ( )Area المسؤولة عن حساب مساحة الدائرة، حيث يتم إظهار رسالة تطلب من المستخدم قيمة نصف قطر الدائرة، ومن ثم طباعة الناتج، مع العلم أن مساحة الدائرة تساوى: ط * نق2 .

2) صمم صفحة ويب تحتوي على زر بعنوان "عمليات المستطيل"، وعند الضغط على هذا الزر يستدعى الدالة ( )Rect المسؤولة عن حساب كل من مساحة المستطيل ومحيط المستطيل، حيث يتم إظهار رسالة تطلب من المستخدم إدخال قيمة كل من طول وعرض المستطيل، ومن ثم طباعة ناتج هذه العمليات.

الخلاصة

في هذه الوحدة عزيزي الدارس، قمنا بدراسة عميل الويب (Web Client)، خادم الويب (Web Server)، والأوامر النصية التي تعالج بهما، وعندما تعالج هذه الأوامر على عميل الويب، أطلقنا عليها اسم "البرمجـة لدى العميل" (Client-Side Programming) الذي هو محور اهتمامنا خلال هذه الوحدة، وعند معالجتها على خادم الويب نطلق عليـها اسم "البرمجة لدى الخادم" (Server-Side Programming) الذي سيكون موضوع الوحدة الخامسة.

لمحة مسبقة عن الوحدة التالية

عزيزي الدارس، في وحدتنا التالية سوف نستكمل موضوع البرمجة لدى عميل الويب خلال هذه الوحدة، حيث نعرض كيف نتحقق من صحة البيانات المدخلة بواسطة المستخدم على عميل الويب قبل إرسالها إلى خادم الويب، كما سنتعرض إلى أشهر الأحداث (Events) التي يمكن أن تحدث لدى عميل الويب وتحتاج من المبرمج تطوير برامج فرعية تنفذ استجابة لتلك الأحداث، وأخيرا نتعرف على كائنات مكتبة DHTML التي تساعد مطوري تطبيقات مواقع الويب إلى تطوير واجهة ذات طبيعة تفاعلية. نتمنى أن تستمتع بدراسة هذا الجزء من رحلة بناء النظم.

إجابات التدريبات

الإجابة:

```
<script language = "javascript">
Function Add()
{
         var firstNumber,   // first string entered by user
         secondNumber,  // second string entered by user
         res;           // sum of number1 and number2
         // read in first number from user as a string
         firstNumber = prompt( "أدخل الرقم الاول", "0" );
         secondNumber = prompt( "أدخل الرقم الثانى  ", "0" );
         res = firstNumber * secondNumber;
         // display the results
         alert( "ناتج الضرب يساوى " + sum );
}
</script>
```

الإجابة:

```
<script language = "javascript">
Function Volume()
{
         var high,
         rad,
         vol;
         High = prompt( "أدخل ارتفاع الاسطوانة", "0" );
         rad = prompt("أدخل نصف قطر قاعدة الاسطوانة  ", "0" );
         vol = high * rad * rad * 3.14;
         // display the results
         alert( "حجم الاسطوانة يساوى " + vol );
}
</script>
```

المراجع

[1] عبدالكريم العقيل، "المخطط الأزرق لبناء صفحات ويب ديناميكية"، مكتبة جرير، الرياض، 2001م.

[2] د. أحمد أحمد شعبان دسوقى، أ. د. السيد محمود الربيعي، "تقنيات وبرمجة الإنترنت" مكتبة الرشد، الرياض، 2004م.

[3] د. عبد القادر عبد الله الفنتوخ، "الإنترنت للمستخدم العربي"، مكتبة العبيكان، الرياض، 2001م.

[4] د. إبراهيم فتحي معوض، "أساسيات برمجة الإنترنت"، معهد العالمية للحاسب والتقنية، 2005م.



الوحدة الرابعة

البرمجة لدى العميل
(Client-Side Programming):
الجزء الثاني

محتويات الوحدة

الموضوع	رقم الصفحة
المقدمة	98
تمهيد	98
الأهداف	99
1. التحقق من صحة بيانات النماذج	100
1.1 أولا: شفرة النموذج	103
2.1 ثانيا: شفرة دالة التحقق من صحة البيانات	104
1.2.1 التحقق من صحة بيانات نموذج تسجيل مستخدم جديد	105
2.2.1 التحقق من أزرار الاختيار (Radio Buttons)	107
3.2.1 التحقق من القائمة المنسدلة (Drop-down Menu)	107
4.2.1 التحقق من صناديق التأشير (Check Boxes)	108
2. أحداث مكتبة DHTML	109
1.2 الحدث On Click	109
2.2 الحدث OnLoad	111
3.2 الأحداث OnMouseOut, OnMouseOver, OnMouseMove	113
3. كائنات مكتبة DHTML	115

4. كيفية التعامل مع كائنات DHTML	115
1.4 الكائن Window	116
2.4 الكائن Document	118
3.4 الكائن Navigator	120
4.4 الكائن IMG	121
5.4 الكائن Form	121
6.4 الكائن Input	123
7.4 الكائن Select	124
الخلاصة	128
لمحة مسبقة	128
اجابات التدريبات	129
المراجع	130

المقدمة

تمهيد

عزيزي الدارس، لقد علمنا من الوحدة السابقة أهمية البرمجة لدى كل من عميل الويب وخادم الويب الذي يؤدي إلى زيادة كفاءة عمل خادم الويب بسبب إجراء بعض معالجة تطبيق الويب على عميل الويب، ولقد تعرفنا أيضا في الوحدة السابقة على كيفية إدراج برامج لغة Javascript داخل مستند HTML، كما قدمنا لغة Javascript بشيء من التفصيل الذي يعين مبرمجي تطبيقات الويب على تطوير برامج تعالج لدى عميل الويب.

وسوف نستكمل موضوع البرمجة لدى عميل الويب خلال هذه الوحدة، حيث نعرض كيف نتحقق من صحة البيانات المدخلة بواسطة المستخدم على عميل الويب قبل إرسالها إلى خادم الويب، كما سنتعرض إلى أشهر الأحداث (Events) التي يمكن أن تحدث لدى عميل الويب وتحتاج من المبرمج تطوير برامج فرعية تنفذ استجابة لتلك الأحداث، وأخيرا نتعرف على كائنات مكتبة DHTML التي تساعد مطوري تطبيقات مواقع الويب على تطوير واجهة ذات طبيعة تفاعلية.

أهداف الوحدة

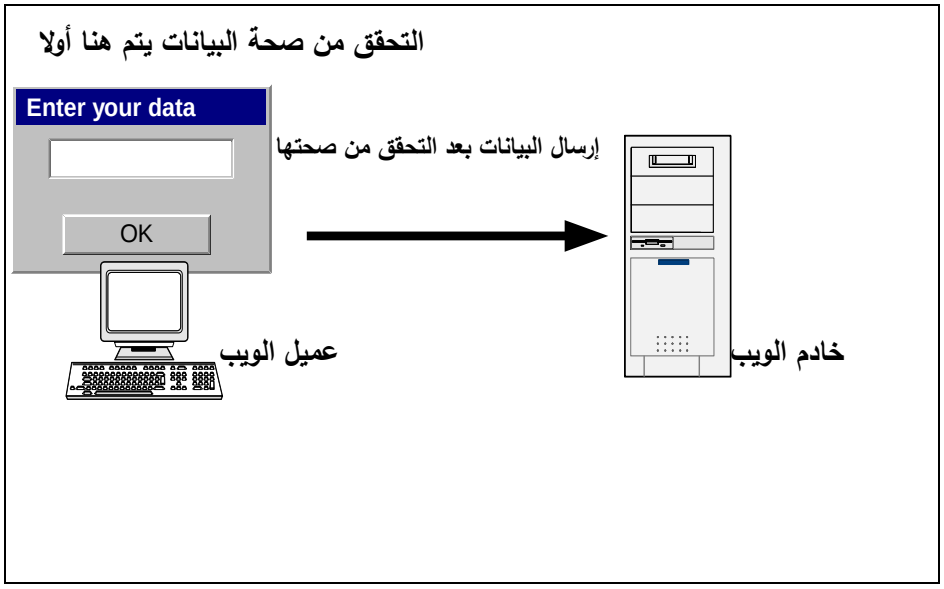


عزيزي الدارس، بنهاية دراسة هذه الوحدة ينبغي أن تكون قادرا على أن:

1. تتحقق من صحة بيانات النماذج.
2. **تُعرّف** شفرة النموذج.
3. **تبين** أحداث مكتبة DHTML .
4. **تحدد** كائنات مكتبة DHTML.

1. التحقق من صحة بيانات النماذج

عزيزي الدارس، إن معظم تطبيقات الويب تستقبل بيانات من المستخدم كي يتم تخزينها أو معالجاتها لدى خادم الويب، مثل استقبال بيانات المستخدم عند تسجيله في موقع ما، ولكن يجب أولاً أن نتحقق من صحة البيانات لدى عميل الويب قبل إرسالها إلى خادم الويب، وذلك لتقليل وقت ومجهود المستخدم، وتوفير موارد خادم الويب للمهام الأخرى (انظر شكل 1-4). فإذا طلبنا من المستخدم مثلا أن يملأ جميع بياناته داخل نموذج التسجيل الذي يحتوي على حقل تاريخ الميلاد (يجب أن تكون القيمة المدخلة تاريخ)، ثم يدخل المستخدم جميع بياناته بشكل صحيح. ولكن إذا أدخل مجموعة من الحروف داخل حقل تاريخ الميلاد بدلا من إدخال تاريخ ميلاده، ثم قام المستخدم بإرسال النموذج إلى الخادم، فإذا لم نتحقق من البيانات قبل إرسالها إلى خادم الويب، سيظهر خادم الويب رسالة خطأ ويضطر المستخدم إلى إعادة إدخال جميع البيانات مرة ثانية مع تصحيح أخطائه.



شكل 1- 4: التحقق من صحة بيانات النموذج لدى عميل الويب

دعنا نصمم نموذجاً بسيطاً يحتوي على برنامج فرعي بلغة Javascript للتحقق من صحة البيانات، سيحتوي النموذج على حقل نص واحد (Text Field) لإدخال اسم المستخدم وعلى زر إرسال (Submit Button)، سنتحقق في هذا المثال من عدم وجود قيمة فارغة داخل حقل النص عند إرسال النموذج إلى الخادم (أي أن المستخدم لم يترك الاسم فارغاً)، فإذا لم يدخل المستخدم اسمه ثم ضغط على زر إرسال ستظهر له رسالة الخطأ "يجب أن تكتب اسمك أولا" (كما هو واضح في شكل 2-4)



شكل 2-4: التحقق من عدم إدخال قيمة نصية فارغة داخل اسم المستخدم

ويوضح شكل 3-4 كل من شفرة HTML الخاصة بالنموذج وشفرة لغة Javascript المسؤولة عن إظهار رسالة الخطأ عند اكتشاف قيمة فارغة داخل الاسم، ومن ثم عدم إرسال بيانات النموذج إلى خادم الويب.

```
<html>
<head>
</title>التحقق من الاسم<title>

<script language="JavaScript">
<!--
function validate_form()
{
valid = true;
if(document.formName.txtName.value=="")
{
alert("   يجب أن تكتب اسمك أولا ");
       valid = false;
        }
return valid
}
//-->
</script>

</head>
<body>

<form name="formName" method="POST" action="welcome.asp"
onSubmit="return validate_form();">
</p>  الاسم <p><input type="text" name="txtName" size="20">
</p> " name="send">إرسال<p><input type="submit" value="
</form>

</body>
</html>
```

شفرة Javascript للتحقق من صحة بيانات النموذج

شفرة النموذج

شكل 3- 4: كل من شفرة النموذج وشفرة *JavaScript* المسؤولة عن التحقق من صحة البيانات

والآن عزيزي الدارس هيا لنتصفح شفرة HTML الخاصة بالنموذج وشفرة لغة Javascript لنتعلم كيف نكتب برامج تتحقق من صحة بيانات النموذج لدى عميل الويب قبل إرسالها إلى خادم الويب.

1.1 أولا: شفرة النموذج

يوضح الجزء الأول من شفرة النموذج (انظر الأوامر التالية) أن اسم النموذج هو "formName" الذي سوف يستخدم للإشارة إلى النموذج داخل شفرة Javascript، كما يوضح أن النموذج يستخدم أسلوب "POST" لإرسال محتوى النموذج إلى المستند "welcome.asp" المخزن لدى خادم الويب والمسؤول عن تحية المستخدم في حال إدخال الاسم بشكل صحيح، و أخيرا يوضح أن الدالة validate_form() المسؤولة عن التحقق من صحة البيانات سوف يتم استدعاؤها عند حدوث الحدث onSubmit، لاحظ أن الأمر return يسمح لنا إعادة القيمة العائدة من الدالة validate_form() إلى عميل الويب، حيث إن الدالة validate_form() سوف تعيد القيمة true والتي تعني إرسال محتويات النموذج إلى خادم الويب أو القيمة false التي تعني عدم إرسال محتويات النموذج إلى خادم الويب.

```
<form name="formName" method="POST" action="welcome.asp"
onSubmit="return validate_form();">
```

ثم يطلب بقية النموذج من المستخدم أن يدخل اسمه داخل حقل النص txtName (Text Field)، كما يحتوى على الزر "إرسال" (Submit Button) المسؤول عن إرسال محتويات النموذج إلى خادم الويب كما هو واضح من الأوامر التالية:

```
<p><input type="text" name="txtName" size="20">  الاسم </p>
<p><input type="submit" value="إرسال" name="send"> </p>
</form>
```

2.1 ثانيا: شفرة دالة التحقق من صحة البيانات

عزيزي الدارس ،الآن دعنا نتصفح أوامر الدالة validate_form() المسؤولة عن التحقق من صحة محتويات النموذج، ومن ثم إعادة القيمة True إذا كانت البيانات صحيحة أو إظهار رسالة خطأ وإعادة القيمة False إذا كانت البيانات خاطئة (أي فارغة). يتواجد تعريف الدالة validate_form() داخل العلامة <head> و العلامة </head> القريبتين من رأس مستند HTML (انظر الأوامر التالية)، حيث تبدأ أوامر لغة JavaScript بالأمر (<script language="JavaScript">) الذي يخبر عميل الويب أن الأوامر التالية مكتوبة بلغة JavaScript، ثم الأمر (<!--) الذي يمثل تعليق شفرة HTML لكي نخفي أوامر لغة JavaScript عن المتصفحات القديمة التي لا تستطيع تنفيذ أوامر لغة JavaScript.

```
<script language="JavaScript">
<!--
function validate_form()
{
   valid = true;
   if(document.formName.txtName.value=="")
{
alert("يجب أن تكتب اسمك أولا")
valid = false;
}
return valid;
}
//-->
</script>
```

ثم تبدأ أوامر الدالة validate_form() بتعريف المتغير valid وإسناد القيمة true له. سوف نستخدم هذا المتغير لمعرفة هل تم تعبئة البيانات بشكل صحيح أم لا، بمعنى أننا نفترض في البداية أن جميع بيانات النموذج صحيحة ولذلك أسندنا له القيمة true، ولكن سوف تتغير هذه القيمة إلى false إذا فشل أحد اختبارات التحقق من صحة البيانات وبالتالي لم يرسل محتوى النموذج إلى خادم الويب.

ثم نتحقق من عدم إدخال قيمة نصية فارغة داخل حقل اسم المستخدم بواسطة الشرط if(document.formName.txtName.value=="") الذي يقارن القيمة النصية الفارغة ("") بالصفة value لحقل النص txtName المتواجد داخل النموذج formName المنتمي للمستند الحالي (document)، فإذا تحقق هذا الشرط سوف تظهر رسالة الخطأ "يجب أن تكتب اسمك أولا " للمستخدم ثم نسند القيمة false إلى المتغير valid.

وأخيرا نعيد قيمة المتغير valid إلى الصفة onSubmit، فإذا كانت القيمة العائدة تساوي true سوف يتم إرسال محتوى النموذج إلى خادم الويب، أما إذا كانت القيمة العائدة false عندئذ لن يتم إرسال محتوى النموذج إلى خادم الويب، ثم نختم أوامر لغة JavaScript بنهاية تعليق HTML (//-->) وإنهاء علامة الأوامر النصية (</script>).

1.2.1 التحقق من صحة بيانات نموذج تسجيل مستخدم جديد

عزيزي الدارس، لتعميم ما تعلمناه من مفاهيم، دعنا نطبق هذه المفاهيم للتحقق من محتويات نموذج يستخدم لتسجيل مستخدم جديد داخل موقع ما، ومن ثم سوف نتعلم كيف نتحقق من صحة البيانات المدخلة داخل حقول أخرى مثل أزرار الاختيار (Radio Buttons)، وصناديق التأشير (Checkboxes)، والقوائم المنسدلة (Drop-down lists). يوضح شكل 4-4 كلاً من نافذة نموذج التسجيل المراد التحقق من صحة بياناته وشفرة HTML المناظرة له، حيث يحتوي هذا النموذج على حقل نصي (txtName) لإدخال اسم المستخدم، وأزرار اختيار (rdbGender) لاختيار نوع المستخدم (ذكر أو أنثى)، وقائمة منسدلة (cmbNationality) لاختيار جنسية المستخدم، وصندوق تأشير (chkAgree) لمعرفة هل المستخدم موافق على شروط التسجيل بالموقع أم لا.

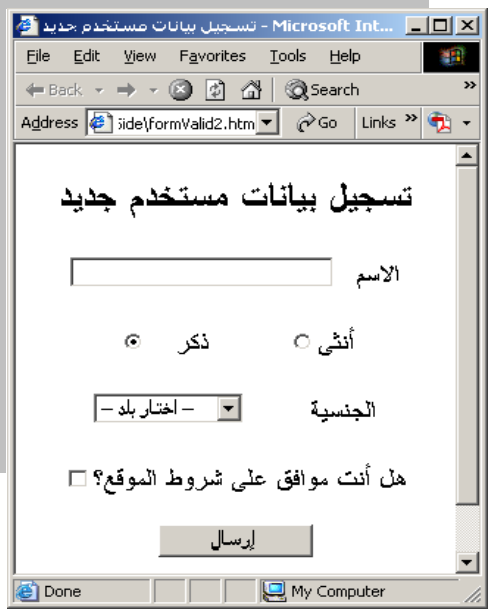
```
<html>
<head>
<title>تسجيل بيانات مستخدم جديد</title>
</head>

<body>
<h2 align="center"><b>تسجيل بيانات مستخدم جديد</b></h2>

<form method="POST" name="rigstForm"
   action="regist.asp" onSubmit="return validate_form();">
  <input type="text" name="txtName" size="26">الاسم
  <input type="radio" name="rdbGender" value="V1" >أنثى
  <input type="radio" name="rdbGender" value="V2">ذكر
  <select size="1" name="cmbNationality">
    <option selected>-- اختار بلد --</option>
    <option>مصر</option>
    <option>السودان</option>
    <option>السعودية</option>
  </select>الجنسية
 <input type="checkbox"
    name="chkAgree" value="ON">
    هل انت موافق على شروط الموقع؟
  <input type="submit" value="إرسال"
   name="btnSend">
</form>

</body>

</html>
```



شكل 4- 4: نموذج التسجيل وشفرة HTML المناظرة له

وفيما يلي سنتعرف كيف نتعامل مع كل من هذه الحقول للتحقق من صحة بياناتها، ولكن يجب علينا أن نعرف الإجراء validate_form() أولاً كما فعلنا في المثال السابق، ولذلك سنكتفي في هذا المثال بإضافة أوامر التحقق من الحقول الجديدة فقط.

2.2.1 التحقق من أزرار الاختيار (Radio Buttons)

عزيزي الدارس لمعرفة ما إذا كان المستخدم اختار أحد أزار الاختيار أم لا لتحديد نوع المستخدم، نكتب أمر If التالي الذي يختبر حالة كل زر على حدة، حيث تكون الصفة Checked للزر المختار تساوى true وتكون false للزر الغير مختار، فإذا كانت قيمة الصفة Checked لكلا الزرين تساوي false فهذا يعني أن المستخدم لم يقم باختيار نوع المستخدم، وبالتالي يجب إظهار رسالة خطأ وتغيير قيمة المتغير valid إلى القيمة false.

```
if(document.rigstForm.rdbGender[0].Checked ==false &&
   document.rigstForm.rdbGender[1].Checked ==false  )
         {
         Alert("first writing of the name")
            valid = false;
         }
```

لاحظ أن كلاً من زري الاختيار قد تم تسميتهما بنفس الاسم (rdbGender) ومن ثم تكونت مصفوفة تحتوي على هذين الزرين: زر الاختيار الأول الذي نشير إليه بالفهرس رقم صفر، وزر الاختيار الثاني الذي نشير إليه بالفهرس رقم واحد.

3.2.1 التحقق من القائمة المنسدلة (Drop-down Menu)

ولمعرفة ما إذا كان المستخدم اختار أحدى البلاد أم لا لتحديد جنسية المستخدم، نكتب أمر If التالي الذي يختبر قيمة فهرس القائمة المنسدلة، فإذا كانت قيمة فهرس القائمة المنسدلة تساوي صفراً فهذا يعني أن المستخدم لم يقم باختيار إحدى البلاد، ومن ثم يجب إظهار رسالة خطأ وتغيير قيمة المتغير valid إلى القيمة false.

```
if(document.rigstForm.cmbNationality.selectedIndex == 0 )
{
   alert("   يجب أن تقوم باختيار جنسية المستخدم أولا")
    valid = false;
}
```

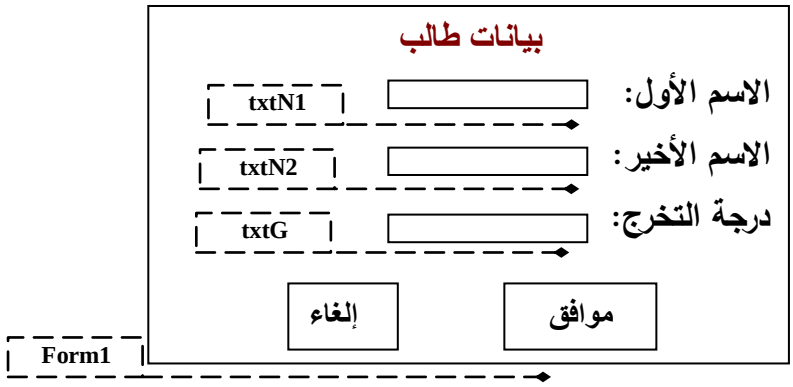
4.2.1 التحقق من صناديق التأشير (Check Boxes)

ولمعرفة ما إذا كان المستخدم قد وافق على شروط الموقع أم لا، نكتب أمر If التالي الذي يختبر حالة صندوق التأشير، حيث تساوى الصفة Checked الخاصة به القيمة true إذا كان مختارا، وتكون false إذا كان صندوق التأشير غير مختار، فإذا كانت قيمة الصفة Checked لصندوق تأشير الموافقة على شروط الموقع تساوي false فهذا يعني أن المستخدم لم يقم بالموافقة، وبالتالي يجب إظهار رسالة خطأ وتغيير قيمة المتغير valid إلى القيمة false.

تدريب(1)



مستخدما برنامج MS Front Page صمم النموذج التالي المسؤول عن استقبال بيانات طالب (الاسم الأول والاسم الأخير ودرجة التخرج).



2. أحداث مكتبة DHTML

عزيزي الدارس، كما هو الحال في جميع البرامج التي تعمل تحت بيئة الويندوز، فإن محتويات صفحة الويب تستطيع أن تستجيب إلى أحداث تصدر بواسطة المستخدم أو غيره بواسطة تنفيذ مجموعة دوال معرفة بواسطة المبرمج والتي تسمى بمعالجات الأحداث (Event Handlers) لأن نظام التشغيل يقوم باستدعائها عند إصدار الأحداث المرتبطة بها، ومن أمثلة هذه الأحداث: حدث Onclick (يصدر عند الضغط على الزر الأيسر للماوس)، حدث OnLoad (عند تحميل محتويات كائن محدد في الذاكرة)، سنتعرف في هذا الموضوع على أهم هذه الأحداث، يوضح الأمر التالي الصيغة العامة للحدث ومعالج الحدث المرتبط به.

```
< EventName > = "< EventHandler> "
```

تكتب هذه الصيغة داخل علامة HTML لأحد عناصر صفحة الويب مثل الصور (Images)، والأزرار (Buttons)... إلخ حيث يمثل EventName اسم الحدث مثل الحدث OnClick، ويمثل EventHandler دالة معالجة الحدث التي ستستدعى عند حدوث ذلك الحدث، مع العلم أنه يمكن كتابة أوامر Javascript مباشرة بدلا من اسم دالة معالجة الحدث.

1.2 الحدث On Click

يعتبر هذا الحدث من أهم الأحداث وأكثرها استخداماً، ويصدر هذا الحدث عندما يضغط المستخدم على الزر الأيسر للماوس ويتركه، أو عندما يضغط على مفتاح Enter عند التعامل مع النموذج.

مثال: الاستجابة إلى الحدث Onclick

يوضح المثال التالي (شكل 5-4) صفحة ويب تحتوى على زر "Click me" ، وعند الضغط عليه تظهر رسالة تحتوي على التحية " أهلا بك في عالم البرمجة لدى العميل" ، ومن الملاحظ أن شفرة معالجة الحدث (الأمر Alert()) قد تم كتابته مباشرة بدلا من اسم دالة معالجة الحدث.

```
<head>

<script language="javascript">
</script>

</head>

<body>

<input type="button" value="Click me" onclick
 = "alert('أهلا بك في عالم البرمجة لدى العميل');">
```



شكل 5- 4: الاستجابة إلى الحدث OnClick

تدريب (2)



أضف الدالة validate_form() إلى النموذج السابق في تدريب (1) لتتحقق من صحة البيانات المدخلة بواسطة المستخدم وإظهار رسالة خطأ عند إدخال قيمة فارغة لإحدى هذه القيم.

2.2 الحدث On Load

يصدر هذا الحدث بعد تحميل أحد عناصر صفحة الويب بواسطة عميل الويب مباشرة داخل الذاكرة، حيث يتم تحميل عناصر صفحة الويب عند قراءة علامات HTML المناظرة لهما أثناء قراءة شفرة مستند HTML، وعادة يستخدم هذا الحدث لكتابة أوامر تنفذ عند نجاح تحميل مستند HTML ، وذلك لإظهار رسائل بخصوص موضوع ما أو لإسناد قيم ابتدائية لبعض المتغيرات.

مثال :الاستجابة إلى الحدث OnLoad

يوضح المثال التالي (الموضح في شكل 6-4) استخدام حدث onLoad لتطوير برنامج يستخدم لحساب حجم المكعب، حيث تم تطوير الدالة cube() وربطها بالحدث OnLoad لمستند HTML، ومن ثم ستظهر رسالة تحث المستخدم على إدخال طول ضلع المكعب، ثم تظهر له رسالة أخرى تحتوي على حجم المكعب.

شكل 6- 4: الاستجابة إلى الحدث
OnLoad

```
<script language="javascript">

function cube()
  {
 var number , cube  ;
number=prompt("أدخل طول ضلع المكعب");
 cube = number * number * number;
alert("   :حجم المكعب" + cube );
  }

</script>

<body onLoad = "cube();">

<H1> برنامج حساب حجم مكعب</H1>

</body>

</html>
```



3.2 الأحداث OnMouseOut, OnMouseOver, OnMouseMove

عزيزي الدارس، يصدر حدث OnMouseMove عندما يتحرك الماوس داخل حدود عنصر ما مثل الصورة، الجدول،... إلخ، وفي الحقيقة يوجد هناك حدثان مرتبطان بهذا الحدث ارتباطاً وثيقاً ويجب الإشارة إلى تلك الأحداث وهي: الحدث OnMouseOver، والحدث OnMouseOut، حيث يصدر الحدث OnMouseOver أولاً وذلك عندما يقع مؤشر الماوس في بداية حدود العنصر، ثم يصدر الحدث OnMouseMove عندما يتحرك مؤشر الماوس داخل حدود العنصر، ثم يصدر الحدث OnMouseOut عندما يتحرك مؤشر الماوس بعيداً عن حدود العنصر.

مثال: الاستجابة إلى الأحداث

OnMouseOut, OnMouseOver OnMouseMove

يوضح المثال التالي (الموضح في شكل 7-4) استخدام كل من الحدث OnMouseOver ، والحدث OnMouseMove ، والحدث OnMouseOut لتغيير عنوان الزر btnMove بشكل مستمر، وذلك بواسطة استدعاء الدالة overFun() عند إصدار الحدث OnMouseOver لتغيير العنوان إلى (المؤشر سيتحرك) ، واستدعاء الدالة moveFun() عند إصدار الحدث OnMouseMove لإضافة نقطة على عنوان الزر btnMove مع كل حركة للماوس، واستدعاء الدالة outFun() عند إصدار الحدث OnMouseOut لتغيير عنوان الزر btnMove إلى (المؤشر ذهب)، لاحظ أن الدوال تغير عنوان الزر btnMove بواسطة تغيير قيمة الصفة Value للزر btnMove.

شكل 7-4: الاستجابة إلى الأحداث OnMouseOut, OnMouseOver, OnMouseMove

```
<script language="javascript">

function overFun()
{
 btnMove.value="المؤشر سيتحرك" ;
}

function moveFun()
 {
        btnMove.value= "." & btnMove.value ;
 }

function outFun()
{
btnMove.value=  "المؤشر ذهب" ;
}
</script>
<body>
 <input type="button" value="لا توجد أحداث"
name="btnMove"
OnMouseMove="moveFun();"
onmouseover="overFun();"
OnMouseOut="outFun();" >

</body>
```



3. كائنات مكتبة DHTML

عزيزي الدارس، تحتوي مكتبة DHTML على مجموعة من الكائنات والتي تمكن المبرمج من التحكم في عناصر صفحة الويب وتغيير ظهورها للمستخدم، وذلك عن طريق استدعاء دوال هذه الكائنات أو تغيير قيم السمات الخاصة بها (Attributes)، وذلك عند إصدار أحداث من قبل المستخدم مثل حدث OnClick (الضغط على الزر الأيسر للماوس) أو غيره مثل حدث OnLoad (تحميل عنصر HTML ما).

كما ذكرنا سابقاً، تتكون صفحة الويب من عدة عناصر مثل الصور، الجداول، النصوص، النماذج. حيث إن مكتبة DHTML تحتوي على كائنات تناظر لهذه العناصر، مع العلم أن هذه الكائنات تنتظم وترتبط في شكل شجري حيث يمثل الكائن Document جذر هذه الشجرة.

4. كيفية التعامل مع كائنات DHTML

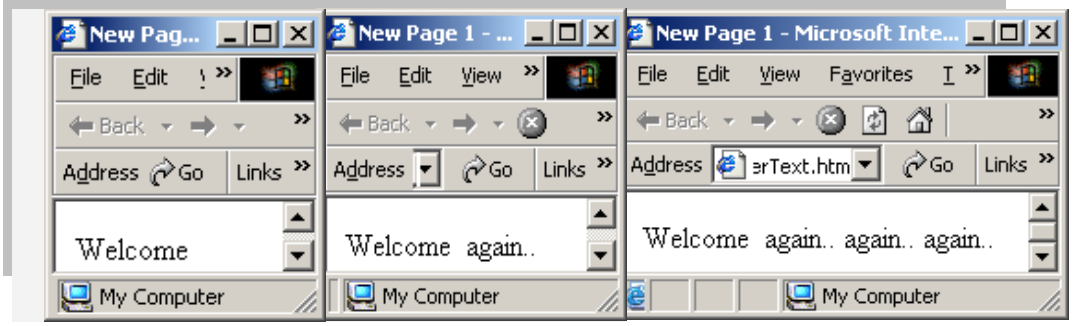
عزيزي الدارس، يمكن استدعاء دوال الكائنات وتغيير سماتها عن طريق إعطاء اسم فريد لعناصر صفحة الويب مستخدماً السمة ID التي تستخدم لإعطاء اسم وحيد لأحد عناصر صفحة الويب، ثم يتم استخدم هذا الاسم للإشارة إلى الكائن المناظر له في أوامر لغة Javascript

مثال: تغيير نص عند الضغط عليه

يوضح المثال التالي استخدام السمة id للتعامل مع عناصر صفحة الويب حيث تم تعريف السمة id للكلمة "Welcome" وإسناد الاسم "welPara" لها ثم الإشارة إلى الكائن المناظر لهذا النص مستخدما الاسم "welPara"، وتغيير نص الكائن بإضافة الكلمة "..again" للصفة innerText الخاصة به عند كل مرة ضغط على النص (صدور الحدث onclick واستدعاء الدالة moreWelcome()).

```
<script language="javascript"  >
function moreWelcome()
{
welPara. innerText = welPara. innerText + " again.." ;
}
</script>

<body>
<p id="welPara" onclick= "moreWelcome();">Welcome</b>
</p>
</body>
```



شكل 8-4: تغيير نص عند الضغط عليه

1.4 الكائن Window

عزيزي الدارس، يمثل هذا الكائن نافذة عميل الويب (متصفح الإنترنت) ، ويحتوي هذا الكائن على العديد من الدوال والصفات التي تمكن المبرمج من استرجاع معلومات حول حالة نافذة متصفح الإنترنت، ومن التعامل مع صفحة الويب (Document) التي توجد بداخل هذه النافذة، و أخيرا من استرجاع معلومات حول متصفح الإنترنت نفسه.

يجب التنويه أن متصفح الإنترنت ينشئ كائن Window واحد فقط عند عرض صفحة ويب داخله، ولكن إذا احتوت هذه الصفحة على أكثر من إطار (Frame)، فإن متصفح الإنترنت ينشئ كائن Window إضافي لكل إطار يوجد داخل صفحة الويب، وتعتبر هذه الكائنات (Window objects) الخاصة بالإطارات فروعاً من كائن

Window الأصلي وتتأثر به، فمثلاً إذا أغلقت النافذة الأصلية، تغلق جميع كائنات Window الأخرى. يوضح الجدول رقم 1-4 أهم دوال (Function) وسمات (Attributes) الكائن Window.

جدول 1- 4 وصف أهم دوال و صفات الكائن Window

الدوال (Functions)	
Alert (<Mess>)	تستخدم هذه الدالة لإظهار رسالة تنبيه للمستخدم.
Close()	تستخدم هذه الدالة لغلق النافذة الحالية.
Navigate(<Url>)	تستخدم هذه الدالة للانتقال إلى موقع جديد عنوانه URL
الصفات (Attributes)	
Document	تستخدم لاسترجاع كائن Document الذي يمثل صفحة الويب التي توجد داخل كائن Window.
History	تستخدم لاسترجاع مجموعة تحتوي على عناوين المواقع التي زارها المستخدم.
Navigator	تستخدم لاسترجاع كائن Navigator (متصفح الويب)
Status	تستخدم لإظهار رسالة أو لمعرفة النص الحالي لشريط الحالة المتواجد أسفل النافذة.
Parent	تستخدم لاسترجاع كائن Window المباشر للإطار الحالي (Current Frame).
Top	يستخدم لاسترجاع كائن Window الذي يمثل النافذة الرئيسية.

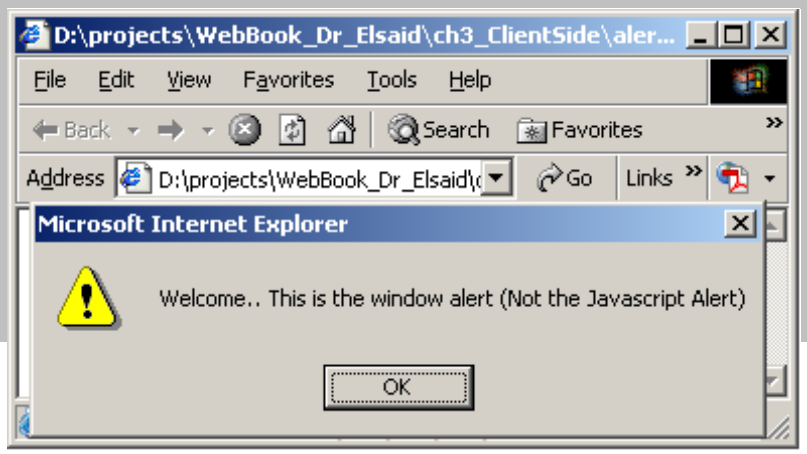
مثال: استخدام إحدى دوال الكائن Window

يوضح المثال التالي (الموضح في شكل 9-4) استخدام دالة window. alert( ) لإظهار رسالة عند تحميل صفحة الويب.

```
<script language="Javascript"  >
function Welcome()
  {
    window. alert("Welcome.. This is the window alert (Not the
                  Javascript Alert)") ;
  }

<body OnLoad="Welcome();">

</body>
```



شكل 9-4: استخدام إحدى دوال الكائن Window

2.4 الكائن Document

عزيزي الدارس، يستخدم هذا الكائن للتعامل مع صفحة الويب التي تظهر بداخل نافذة متصفح الإنترنت. ويحتوى هذا الكائن على العديد من الدوال والصفات التي تمكن المبرمج من استرجاع معلومات حول صفحة الويب، ومن تعديل مظهر محتويات صفحة الويب (عناصر HTML مثل الصور والنصوص).

نستخدم الكائن Document للتعامل مع صفحة الويب الحالية، أما إذا أردنا التعامل مع صفحة ويب غير الحالية نستخدم الصيغة <Window. Document>.

ويوضح الجدول رقم 2-4 أهم دوال (Functions) وسمات (Attributes) ومجموعات (Collections) الكائن Document.

أسئلة تقويم ذاتي



ما وظيفة الدوال والإجراءات التي يطلق عليها "معالجة أحداث"؟.

ما الفرق بين الأحداث OnMouseOver، OnMouseMove، OnMouseOut؟

مستخدما برنامج MS Front Page صمم النموذج التالي المسؤول عن استقبال بيانات اسطوانة (نصف قطر القاعدة وارتفاع الأسطوانة).

جدول 2-4 : وصف أهم دوال ، سمات ، ومجموعات الكائن Document

الدوال (Functions)	
Write(<Mess>)	تستخدم لكتابة رسالة من النوع HTML داخل صفحة الويب.
الصفات (Attributes)	
Bgcolor	تستخدم لتغيير لون خلفية صفحة الويب. مثال: Document.Bgcolor= "blue" ;
Fgcolor	تستخدم لتغير لون النص الذي يكتب داخل صفحة الويب.مثال: Document.Fgcolor= "yellow" ;
Title	تستخدم لتغيير نص عنوان صفحة الويب.
المجموعات (Collections)	
All	تعيد مجموعة تحتوي على جميع عناصر HTML داخل صفحة الويب.
Forms	تعيد مجموعة تحتوي على كل كائنات النوع form (نموذج)

الموجودة حاليا بترتيب ظهورهم داخل صفحة الويب.	
تعيد مجموعة من الكائنات التي من النوع image الموجودة حاليا وذلك بترتيب ظهورها داخل صفحة الويب.	Images

3.4 الكائن Navigator

عزيزي الدارس، يقدم هذا الكائن معلومات عن متصفح الإنترنت وبيئة التشغيل مثل: اسم المتصفح، جيل برنامج المتصفح، نظام التشغيل، بالإضافة إلى معلومات أخرى والتي تساعد المبرمج على كتابة شفرة مناسبة لمتصفح الإنترنت الحالي (مثل معرفة قابلية متصفح الإنترنت لتشغيل برامج Java أم لا) . يوضح الجدول رقم 3-4 أهم دوال (Functions) وسمات (Attributes) الكائن Navigator.

جدول 3-4 : وصف أهم دوال ، سمات ، ومجموعات الكائن Navigator

الدوال (Functions)	
Javaenabled( )	تستخدم هذه الدالة لمعرفة ما إذا كان المتصفح الحالي يستطيع أن ينفذ برنامج Java (Java Applet) أم لا ، حيث تعيد إحدى القيمتين (False, True)
الصفات (Attributes)	
AppName	تعيد نصاً يحتوي على اسم متصفح الإنترنت الحالي (هذه الصفة تقرأ فقط ولا يمكن تعديلها).
AppVersion	تعيد نصاً يحتوي على رقم جيل متصفح الإنترنت الحالي (هذه الصفة تقرأ فقط ولا يمكن تعديلها).
Conectionspeed	تعيد نصاً يحتوي على سرعة الاتصال الحالية.
Platform	تعيد نصاً يحتوي على بيئة نظام التشغيل الحالية، فعلى سبيل المثال يمكن أن تعيد win16 أو win32 حسب نوع نظام الويندوز الحالي.

4.4 الكائن IMG

يستخدم هذا الكائن للتعامل مع صورة أو لقطة فيديو، ويوضح الجدول رقم 4-4 أهم سمات (Attributes) الكائن IMG.

جدول4-4 : وصف أهم سمات الكائن IMG

الصفات Attributes	
Src	تحدد عنوان الملف (Url) الذي تحمل منه الصورة (صفة تقرأ وتعدل).
Href	تحدد العنوان المرتبط بهذه الصورة والذي ينتقل إليه الزائر عند الضغط على هذه الصورة.
Height	تحدد وتعيد ارتفاع الصورة وذلك بوحدة البكسل.
Width	تحدد وتعيد عرض الصورة وذلك بوحدة البكسل.
id	تعيد مؤشراً يشير إلى كائن Img والذي يستخدم للتعامل مع هذا الكائن.

5.4 الكائن Form

يمثل هذا الكائن حاوية لعناصر الإدخال والتي تمكن المستخدم من إرسال بيانات لخادم الويب للمعالجة، ومن خلال هذا الكائن يمكن التعامل مع هذه العناصر. فعلى سبيل المثال يمكن قراءة وتغيير صفات هذه العناصر للتأكد من صحة البيانات. يوضح الجدول رقم 5-4 أهم دوال (Functions) وسمات (Attributes) وأحداث (Events) الكائن Form.

جدول 4-5 : وصف أهم دوال، وسمات ، ومجموعات الكائن Form

الصفات (Attributes)	
Action	يحدد عنوان ملف ASP الذي سيتم استدعاؤه نتيجة إرسال النموذج ليستقبل بياناته ومعالجتها لدى الخادم.
Method	تحدد طريقة إرسال محتويات النموذج إلى الخادم حيث يوجد أسلوبان للإرسال: ▪ Get والتي تستخدم لإرسال محتويات النموذج مصاحبة للعنوان الذي تم كتابته مع سمة action. ▪ Post والتي تستخدم لإرسال محتويات النموذج بشكل منفصل عن عنوان سمة action.
Name	يحتوي عل اسم النموذج.
الدوال (Functions)	
Submit	تستخدم هذه الدالة لإرسال محتويات النموذج إلى الخادم وتصدر حدث OnSubnit.
Reset	تستخدم هذه الدالة لكي تعيد قيم محتويات النموذج إلى القيم الافتراضية المعرفة عند تصميم النموذج.
الأحداث (Events)	
OnReset	يصدر هذا الحدث عندما يضغط المستخدم على الزر Reset لإعادة قيم محتويات النموذج إلى القيم الافتراضية.
OnSubmit	يصدر هذا الحدث عندما يضغط المستخدم على الزر Submit لإرسال محتويات النموذج إلى الخادم.

6.4 الكائن Input

يمثل هذا الكائن أحد عناصر الإدخال التي تتواجد داخل النموذج مثل: reset, submit, radio button, textbox, checkbox, button . و تشترك معظم هذه العناصر في العديد من الصفات، والدوال، والأحداث، عدا العنصر صندوق التأشير (checkbox) والعنصر زر الاختيار (Radio Button) اللذين يحتويان على صفات خاصة بهم، لذا سيتم سرد الأشياء المشتركة للجميع أولاً ثم سرد الصفات الخاصة. يوضح الجدول رقم 6-4 أهم سمات (Attributes) وأحداث (Events) الكائن Input.

جدول 6-4 : وصف أهم دوال ، سمات ، ومجموعات الكائن Input

Attributes الصفات	
Form	تشير هذه الصفة إلى النموذج (الكائن Form) الذي يحتوي على هذا العنصر، وتعيد null إذا كان هذا العنصر لا يوجد داخل نموذج.
id	تستخدم هذه الصفة لإعادة مؤشر لهذا العنصر والذي يستخدم داخل شفرة Javascript للتعامل مع هذا الكائن.
Name	تستخدم هذه الصفة للتعامل مع الكائن (مثل استخدام سمة id) ، ولاحظ أن هذه الصفة لا تعبر عن عنوان العنصر أو محتوياته.
Disabled	تستخدم هذه الصفة لتفعيل أو عدم تفعيل عنصر ما حيث تأخذ هذه الصفة قيمة من اثنتين: True, False.
Value	تستخدم هذه الصفة لاسترجاع القيمة التي أدخلها المستخدم ، أو تشير أيضا إلى عنوان الأزرار.
Title	تستخدم هذه الصفة لإرشاد المستخدم إلى كيفية التعامل مع

عنصر ما، حيث يمكن إسناد رسالة إرشادية داخلها لكي تظهر للمستخدم عند الإشارة بالماوس على هذا العنصر.	
الأحداث (Events)	
يصدر هذا الحدث عندما يعطي التركيز لعنصر ما داخل النموذج (عندما يوضع مؤشر الكتابة داخل هذا العنصر).	On Focus

يتميز كل من العنصر صندوق التأشير (checkbox) والعنصر زر الاختيار (Radio Button) بصفات أخرى، ومن أهم هذه الصفات الصفة checked التي تم مناقشتها سابقا داخل موضوع التحقق من صحة بيانات النماذج.

7.4 الكائن Select

يستخدم هذا الكائن لتعريف قائمة منسدلة داخل نموذج، ومع أن هذا الكائن يحتوى على جميع الصفات والدوال والأحداث التي تنتمي للكائن Input إلا أنه يختص (أي الكائن Select) بصفات ودوال أخرى. يوضح الجدول رقم 7-4 أهم دوال (Functions) وسمات (Attributes) وأحداث (Events) ومجموعات (Collections) الكائن Select.

جدول 7-4 وصف أهم دوال و سمات، ومجموعات وأحداث الكائن Select

الصفات (Attributes)	
Length	تستخدم هذه السمة لإعادة عدد عناصر القائمة المنسدلة (عدد الاختيارات المتواجدة داخل مجموعة Options).
Multiple	تحدد هذه الصفة هل يمكن للمستخدم أن يختار عدة اختيارات أم اختيار واحد فقط، مع العلم أن القيمة الافتراضية لهذه الصفة تساوي False ، والتي تعني اختيار واحد فقط.
SelectedIndex	تستخدم هذه الصفة لتخصيص قيمة أو استرجاع قيمة فهرس الاختيار الحالي.
Size	تستخدم هذه الصفة لتخصيص عدد الصفوف التي تظهر للمستخدم من اختيارات أو استرجاعها.
الدوال (Functions)	
Add	تستخدم هذه الدالة لإضافة عنصر اختيار (يتم إنشاؤه من كائن Option) إلى اختيارات القائمة المنسدلة.
Item	تستخدم هذه الدالة لاسترجاع قيمة اختيار داخل القائمة المنسدلة وذلك بتمرير رقم الفهرس حيث يمكن استرجاع العنصر الأول مستخدما الفهرس صفر، ويمكن استرجاع العنصر الثاني مستخدما الفهرس واحد... وهكذا. مثال: لاسترجاع العنصر الأول وتخزينه داخل المتغير X. X= SelectObject.Item(0)

تابع جدول 7-4 وصف أهم دوال و سمات، ومجموعات وأحداث الكائن Select

لإزالة عنصر اختيار من قائمة منسدلة وذلك بتمرير فهرس هذا الاختيار.	Remove
المجموعات (Collections)	
تحتوي هذه المجموعة على كائنات من النوع Option ، و التي تظهر داخل القائمة المنسدلة.	Option
الأحداث (Events)	
يصدر هذا الحدث عندما يتغير الاختيار الحالي داخل القائمة المنسدلة (أي عندما يختار المستخدم اختياراً آخر غير الحالي).	On Change

تدريب (2)



أضف الدالة validate_form( ) إلى النموذج السابق لتتحقق من صحة البيانات المدخلة بواسطة المستخدم وإظهار رسالة خطأ عند إدخال قيمة فارغة لنصف قطر القاعدة، أو لارتفاع الأسطوانة.

أسئلة تقويم ذاتي



صمم صفحة ويب تحتوي على زر بعنوان "ضبط ألوان صفحة الويب" ، وعند الضغط على هذا الزر يُسْتَدْعَى إجراء يطلب من المستخدم إدخال لون خلفية الصفحة، ولون نص الصفحة، وعندئذ تتغير ألوان صفحة الويب الحالية بناء على هذه الاختيارات.

الخلاصة

في هذه الوحدة عزيزي الدارس، قمنا بدراسة البرمجة لدى عميل الويب حيث عرضنا كيف نتحقق من صحة البيانات المدخلة بواسطة المستخدم على عميل الويب قبل إرسالها إلى خادم الويب، كما استعرضنا أشهر الأحداث (Events) التي يمكن أن تحدث لدى عميل الويب وتحتاج من المبرمج إلى تطوير برامج فرعية تنفذ استجابة لتلك الأحداث، وأخيرا تعرفنا على كائنات مكتبة DHTML التي تساعد مطوري تطبيقات مواقع الويب على تطوير واجهة ذات طبيعة تفاعلية.

لمحة مسبقة عن الوحدة التالية

عزيزي الدارس، في الوحدة التالية سوف **نخصصها لشرح لغة VBScript، حيث سيتم التركيز على تعلم إمكانيات اللغة من أوامر (Instructions) وإجراءات ودوال (Procedures & Functions) ، ثم نشرح كيفية التطبيق داخل الوحدة التالية والتي تركز على البرمجة لدى الخادم (Server-Side Scripting).**

إجابات التدريبات

```
<script language="JavaScript">
   function validate_form()
   {
      valid = true;
      if(document.form1.txtN1.value=="")
         {
            alert(".يجب ان تكتب اسمك الاول");
            valid = false;
         }
      if(document.form1.txtN2.value=="")
         {
            alert(".يجب ان تكتب اسمك الاخير");
            valid = false;
         }

      if(document.form1.txtN1.value=="")
         {
            alert("يجب ان تكتب درجة المادة التى تدرسها حاليا");
            valid = false;
         }
      return valid
   }
</script>
```

المراجع

[1] عبدالكريم العقيل، "المخطط الأزرق لبناء صفحات ويب ديناميكية"، مكتبة جرير، الرياض، 2001م.

[2] د. أحمد أحمد شعبان دسوقى، أ. د. السيد محمود الربيعي، "تقنيات وبرمجة الإنترنت" مكتبة الرشد، الرياض، 2004م.

[3] د. عبد القادر عبد الله الفنتوخ، "الإنترنت للمستخدم العربي"، مكتبة العبيكان، الرياض، 2001م.

[4] د. إبراهيم فتحي معوض، "أساسيات برمجة الإنترنت"، معهد العالمية للحاسب والتقنية، 2005م.


الوحدة الخامسة
البرمجة لدى الخادم
لغة VBScript

محتويات الوحدة

الموضوع	رقم الصفحة
المقدمة	134
تمهيد	134
الأهداف	135
1. الإعلان عن المتغيرات (Variables Declaration)	136
2.التعابير الحسابية	138
3. استخدام الثوابت (Constants)	139
4. تعابير المقارنة	140
5. التعابير المنطقية	141
6. معاملات اللصق & , +	142
7. المصفوفات (Arrays)	143
1.7 استخدام الدالة ( ) Array	145
2.7 استخدام الدالة ( ) UBound	146
8. عبارات التحكم في سير البرنامج	147
9. أولا: العبارات الشرطية (Conditional Statements	147
1.9 عبارة IF. .Then	147
10. عبارة IF. .Then. .Else	148
11. عبارة If. .Then. .Else If	149
1.11 العبارة Select Case	151
12. ثانيا: عبارات التكرار (Repetitive Statements)	153
1.12 عبارة For. .Next	153

2.12 عبارة For Each. .Next	154
3.12 عبارة While. .wend	155
4.12 عبارة Do. .Loop while	156
13. الإجراءات والدوال (Procedures & Functions)	157
1.13 الإجراء الفرعي (Procedure)	158
2.13 الدوال (Functions)	159
3.13 استخدام عبارتي Exit sub & Exit	161
14. التعرف على دوال وإجراءات مكتبة لغة VBScript	163
15. دوال الوقت والتاريخ	163
1.15 دوال معالجة النصوص (String Manipulation Functions)	165
2.15 دوال التحقق من نوع البيانات	167
الخلاصة	169
لمحة مسبقة	169
اجابات التدريبات	170
المراجع	171

المقدمة

تمهيد

لكي تتمكن عزيزي الدارس من إنشاء موقع إنترنت ديناميكي (Dynamic Web Site) تتغير محتوياته من حين لآخر بناء على تغير مصدر بيانات الموقع داخل قاعدة بيانات الموقع ؛ يجب عليك أن تتعلم لغة برمجة نصية (Scripting Language) والتي تمدك بالعديد من الأوامر التي تمكنك من كتابة هذه البرامج، وبما أن لغة البرمجة الافتراضية لدى خادم الويب داخل تقنية Active Server Pages هي لغة VBScript (وتنطق في بي سكريبت)، سوف نخصص هذه الوحدة لشرح هذه اللغة، حيث سيتم التركيز على تعلم إمكانيات اللغة من أوامر (Instructions) وإجراءات ودوال (Procedures & Functions)، ثم نشرح كيفية التطبيق داخل الوحدة التالية والتي تركز على البرمجة لدى الخادم (Server-Side Scripting).تتميز لغة VBScript بسهولة تعلمها، حيث إنها تتشارك في العديد من خصائص لغة البيسيك (Basic) الشهيرة والتي تم تطويرها خصيصاُ للمبتدئين وقد تم تطوير لغة VBScript خصيصاً لاستخدامها في مستندات الويب، وهي تتميز بالسهولة والكفاءة والتوافق مع صفحات Active Sever Pages.

عند مقارنة لغة VBScript بلغات البرمجة النصية الأخرى، سنكتشف أن لغة VBScript لغة سهلة الاستخدام، فالعديد من اللغات الأخرى تتطلب منك أن تتبع قواعد صارمة لبناء الأوامر، مثل إضافة فصلة منقوطة ";" (Semicolon) في نهاية كل أمر، أو استخدام أسماء حساسة لحالة الأحرف (Case Sensitive)، وفي هذه اللغات، يمكن لخطأ إملائي بسيط أن يتسبب في توقف الأوامر عن العمل أو يسلك البرنامج سلوكاً غير صحيح، ولكن لغة VBScript لا تتطلب منك الالتزام بهذه القواعد الصارمة لبناء

العبارات والأوامر، وهذا يجعلها أسهل في الاستخدام من اللغات الأخرى، ومع ذلك، قد تؤدي هذه السهولة إلى صعوبة عملية اكتشاف الأخطاء في بعض الأحيان.

أهداف الوحدة



عزيزي الدارس، بنهاية دراسة هذه الوحدة ينبغي أن تكون قادرا على أن:

1. **تعرّف** كيفية الإعلان عن المتغيرات و الثوابت.
2. **تبين** مصطلحات المقارنة و المصطلحات المنطقية.
3. **تتعامل مع** عبارات التحكم في سير البرنامج.
4. تتعرف على دوال وإجراءات مكتبة لغة VBScript.

1. الإعلان عن المتغيرات

عزيزي الدارس، المتغير (Variable) هو اسم لمكان في الذاكرة يحتوي على قيمة فعلى سبيل المثال يمكن أن تستخدم المتغير myAge ليخزن القيمة 5، ومن الممكن أن تستخدم المتغيرات لإجراء أنواع مختلفة من العمليات الحسابية أو العمليات الأخرى. وللإعلان عن متغير لاستخدامه فيما بعد داخل الأوامر النصية، اكتب عبارة Dim ثم اتبعها باسم المتغير. توضح القواعد التالية الأسلوب الأمثل الذي يجب أن يتبعه المبرمج عند تسمية المتغيرات.

قواعد هامة لتسمية المتغيرات

- يجب أن تختار اسماً له معنى للمتغير، حتى تتمكن من التعرف على الوظيفة التي عرفته من أجلها فيما بعد.
- يمكن أن يصل طول اسم المتغير إلى 255 حرفاً.
- يجب أن يبدأ اسم المتغير بحرف أبجدي ولا يجب أن يحتوي على مسافات أو على نقاط.
- أسماء المتغيرات غير حساسة لحالة الأحرف (Not case sensitive).
- يفضل أن تستخدم عدة كلمات داخل اسم المتغير مع تكبير الحرف الأول من كل كلمة لكي تجعله واضحاً، مثل المتغير BirthDate والذي يستخدم لتخزين تاريخ ميلاد.

الصيغة العامة لتعريف المتغيرات

```
Dim <Var1> , <Var2>...
```

حيث يجب كتابة الكلمة المحجوزة Dim في بداية الإعلان عن المتغيرات ثم يليها أسماء المتغيرات، ويجب أن يفصل بينهم برمز الفاصلة ",". لاحظ أنه تم وضع اسم المتغير بين علامتي < > داخل الصيغة العامة للإشارة وأنه يجب استبدال هذا الاسم باسم المتغير الحقيقي عند التطبيق.

مثال :

يوضح المثال التالي الإعلان عن متغيرين: المتغير Name لتخزين اسم (سلسلة من الحروف)، والمتغير Age لتخزين عمر المستخدم (رقم).

```
Dim Name , Age
```

وبعد الإعلان عن المتغير، يجب أن تقوم بإسناد قيمة مبدئية له، والتي يمكن أن تكون قيمة عددية مثل الرقم 123 أو قيمة نصية مثل النص "Mohammed" ، ويجب أن توضع القيمة النصية داخل علامتي تنصيص.

مثال:

يوضح المثال التالي الإعلان عن متغيرين: المتغير Name لتخزين اسم (سلسلة من الحروف)، والمتغير Age لتخزين عمر المستخدم (رقم) ، ثم تخزين قيمة ابتدائية داخل كل منهم.

```
Dim Name, Age
Name= "Mohammed"
Age=5
```

لاحظ :

أن كل أمر من أوامر لغة VBScript يجب أن يكتب داخل سطر منفصل ، أي لا يمكن أن يكتب أكثر من أمر في سطر واحد، حيث لا توجد هناك علامة تفيد نهاية الأمر مثل لغة JavaScript والتي تستخدم الفاصلة المنقوطة ";" كعلامة نهاية الأمر.

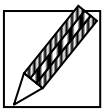
تدريب(1)



4. تتبع جزء البرنامج التالي واطبع ناتج المتغير Factor5.

```
Dim Elem , NumArr , Factor5
Factor5 = 1
NumArr = Array(1, 2, 3, 4, 5)
For each Elem in NumArr
        Factor5 = Factor5 * Elem
Nex
```

تدريب (2)



1. ما فائدة الدالة UCase(var)؟
2. ما فائدة الدالة Trim(Var)؟
3. ما فائدة الدالة Len(var)؟

2.التعابير الحسابية

عزيزي الدارس، يستخدم التعبير الحسابي لحساب علاقة حسابية تتكون من مجموعة من المتغيرات الرقمية (Numeric Variables) ومجموعة من الأرقام بينهما معاملات حسابية (Mathematical Operators)، حيث يمكن إسناد نتيجة معالجة التعبير الحسابي إلى متغير آخر أو طباعة الناتج مباشرة. يوضح الجدول رقم 1-5 وصفاً مختصراً عن المعاملات الحسابية وأسبقية تنفيذ كل منها، وإذا وجد معاملان لهما نفس أولوية التنفيذ، تتم المعالجة من اليسار إلي اليمين. أما إذا أردت أن تغير أسبقية التنفيذ (تقدم معالجة بعض المعاملات عن الآخرى) ، يجب أن تستخدم الأقواس ( ).

جدول 1-5 : وصف مختصر عن المعاملات الحسابية وأسبقية تنفيذها

المعامل	*الوصف*	*مثال*	*الأسبقية*
^	لحساب الأس	X^Y	1
-	لإعطاء إشارة سالبة للرقم	- 5	2
* ، /	لحساب الضرب والقسمة	5*2/3	3
Mod	لحساب باقي القسمة	5 Mod 2	4
+ ، -	لحساب الجمع والطرح	5+2-1	5

أسئلة تقويم ذاتي

1. أعد كتابة المصطلح الحسابي القادم بلغة VBScript مع تحديد أولوية كل معامل $\frac{5+3^2*7-4}{-5*2-1}$

3. استخدام الثوابت (Constants)

عزيزي الدارس، الثابت (Constant) هو متغير يحتوي على قيمة ثابتة، فمثلاً إذا أردت أن تستخدم ضريبة مبيعات قدرها 10% في عملية حسابية، فمن الممكن أن تضع هذه النسبة في ثابت باسم SALES_TAX ،ثم تستخدم هذا الثابت في أي مصطلح حسابي يحتوي على هذه النسبة، واستخدام الثوابت في هذه الحالة أسهل من استخدام القيمة مباشرة لأنها أوضح في المعنى وأسهل في التحديث. فعلى سبيل المثال، إذا تغيرت نسبة الضريبة فإنه يمكنك أن تُحدِّثُها في المكان الذي قمت فيه بتعريف الثابت الخاص بها فقط، ومن ثم سينعكس هذا التحديث على باقي أوامر البرنامج.

يجب أن تختار اسما له معنى وواضح للثوابت، حتى تتمكن من التعرف عليه بسهولة في المستقبل، ومن الأفضل أن تكتب أسماء الثوابت بحروف كبيرة حتى تتمكن من تمييزها

عن المتغيرات الأخرى بسهولة، وإذا كان اسم الثابت يتكون من عدة كلمات، يتم الفصل بينها بحرف الشرطة السفلية "_"، مثل اسم الثابت MAX_NUMBER_OF_USERS الذي يحتوي على القيمة العظمى لعدد المستخدمين.

الصيغة العامة

```
Const <consVar> = <Value>
```

مثال:

يوضح المثال التالي تعريف الثابت PI الذي يخزن القيمة التقريبية الثابتة 3.14.

```
Const PI = 3.14
```

وتحتوى لغة VBScript على العديد من الثوابت الجاهزة التي يمكن استخدامها مباشرة دون الإعلان عنها أو تحديد قيمتها، ولأن هذه الثوابت معرفة داخل اللغة، فإن لها قيمة محددة مسبقاً وبالتالي يمكنك استخدامها في جميع الأوامر النصية التي تكتبها دون الحاجة إلى تحديد قيمتها بنفسك، وفي الغالب يصف اسم الثابت الجاهز القيمة التي يحفظها، فعلى سبيل المثال إن الثابت vbBlack يحتوي على القيمة السداسية عشر للون الأسود، تبدأ معظم الثوابت الجاهزة بالحرفين vb مما يجعل تمييزها والتعرف عليها سهلاً.

4. تعابير المقارنة

عزيزي الدارس، تستخدم تعابير المقارنة بين قيمتين (أو متغيرين) وإصدار نتيجة منطقية (صواب True أو خطأ False)، يحتوي مصطلح المقارنة على معاملات مقارنة (Comparison Operators) مثل المعامل > (أكبر من)، و المعامل < (أصغر من). يوضح الجدول رقم 2-5 وصفاً مختصراً لهذه المعاملات.

جدول 2-5 : وصف مختصر عن معاملات المقارنة

المعامل	الوصف	مثال
>	يختبر إن كانت قيمة ما أكبر من قيمة أخرى	Y>X
<	يختبر إن كانت قيمة ما أصغر من قيمة أخرى	Y<X
>=	يختبر إن كانت قيمة ما أكبر أو تساوي قيمة أخرى	Y>=X
=<	يختبر إن كانت قيمة ما أصغر أو تساوي قيمة أخرى	Y=<X
=	يختبر إن كانت قيمة ما تساوي قيمة أخرى	Y=X
<>	يختبر إن كانت قيمة ما لا تساوي قيمة أخرى	Y<>X

5. التعابير المنطقية

عزيزي الدارس، تستخدم التعابير المنطقية (Logical Expressions) لإجراء عمليات منطقية حيث تتكون هذه التعابير من معاملات منطقية (Logical Operators) مثل: المعامل AND، والمعامل OR، وقيم منطقية (صواب True أو خطأ False). يوضح الجدول رقم 3-5 وصفاً مختصراً للمعاملات المنطقية وأسبقية تنفيذها.

جدول 5-5 : وصف مختصر عن المعاملات المنطقية وأسبقية تنفيذهما

المعامل	الوصف	مثال	الأسبقية
AND	تعطي نتيجة صواب True فقط إذا كانت بين قيمتين صواب True.	X AND Y	2
OR	تعطي نتيجة صواب True فقط إذا كانت بين إحدى قيمتين على الأقل صواب True.	X OR Y	3
NOT	تعكس القيمة التي تصاحبها إذا كانت True تجعلها False ، والعكس صحيح.	Not X	1

تستخدم كل من تعابير المقارنة والمصطلحات المنطقية مع الأوامر الشرطية، حيث تستخدم هذه التعابير كشرط تنفيذي لمجموعة من الأوامر.

6. معاملات اللصق & , +

يستخدم المعامل & للصق نصين بجوار بعضهم وإصدار نص جديد فعلى سبيل المثال يوضح البرنامج التالي كيف نكوِّنُ اسماً كاملاً (Full name) عن طريق لصق الاسم الأول (First name) على نص فارغ ، ثم لصق الناتج على الاسم الأخير (Last name) ، ثم إسناد الناتج إلى المتغير FullName .

```
Dim FirstName, LastName, FullName
FirstName = "Mohammed"
LastName="Ibrahim"
FullName = FirstName &  "  " & LastName
```

كما يمكن استخدام المعامل + لأداء نفس وظيفة المعامل & إذا تم وضعه بين قيم نصية (Strings)، حيث يمكن إعادة المثال العلوي هكذا:

```
Dim FirstName, LastName, FullName
FirstName = "Mohammed"
LastName="Ibrahim"
FullName = FirstName +  "  " + LastName
```

ولكن ليس من المفضل أن تستخدم المعامل + للصق النصوص بغرض عدم تداخل وظائف المعاملات، حيث إن الوظيفة الرئيسية للمعامل + هي جمع قيمتين رقميتين.

لاحظ عزيزي الدارس :

إذا كانت التعابير تحتوي بشكل عام على أكثر من نوع واحد من المعاملات ، ستقوم لغة VBScript بمعالجة التعابير بالترتيب التالي:

1) معالجة المعاملات الحسابية حسب أسبقيتها.
2) معالجة معاملات اللصق.
3) معالجة معاملات المقارنة.
4) معالجة المعاملات المنطقية.

تدريب (3)



ما نتيجة تنفيذ المصطلح التالي؟

(3 > 5) OR (5 < > “a”)

أسئلة تقويم ذاتي



ما نتيجة تنفيذ المصطلحات التالية؟

a) “Ali” + “ ” & “Hassan”

b) (5 > 2) And (-4 < > 4)

c) Not (20 * 2 < 30)

7. المصفوفات (Arrays)

عزيزي الدارس، تقوم المصفوفة (Array) بحفظ مجموعة من القيم المتعلقة ببعضها في كائن واحد. فعلى سبيل المثال، تستطيع المصفوفة حفظ أسماء كل يوم من أيام الأسبوع. ويتيح لك استخدام المصفوفات أن تتعامل مع عدة قيم في وقت واحد، يتم التعامل مع كل عنصر في المصفوفة برقم فهرس (Index) وحيد لا يتكرر.

الإعلان عن المصفوفة (Array Declaration)

عندما تقوم بالإعلان عن المصفوفة، يجب عليك أن تحدد فهرس آخر عنصر وليس عدد العناصر التي ستحتوي عليها المصفوفة، ولاحظ أن فهرس المصفوفة يبدأ من صفر وليس من 1، لذا إذا أعلنت عن مصفوفة بهذا الشكل days0fWeek(6) فإنها تستطيع أن تحفظ على سبعة عناصر وليس ستة عناصر.

الصيغة العامة

```
Dim <ArrayName> (<size>)
```

مثال:

يعلن المثال التالي عن مصفوفة ذات بعد واحد تسمى Num والتي يمكن أن تحتوي على ستة عناصر، حيث أسندنا الرقم 5 لنشير إلى أن العنصر الأخير يأخذ الفهرس 5.

```
Dim Num (5)
```

إذا كنت لا تعرف عدد العناصر التي ستضعها في المصفوفة، يمكنك إنشاء مصفوفة ديناميكية (Dynamic Array) ، وفي هذه الحالة لن تحتاج إلى تحديد فهرس آخر عنصر عند الإعلان عنها من البداية، ولتعريف مصفوفة ديناميكية ، استخدم الأمر Redim بدلا من الكلمة المحجوزة Dim، وعند الاحتياج إلى زيادة عدد العناصر في المصفوفة بعد ذلك، نستخدم العبارة: Redim Preserve. يجب استخدام الكلمة المحجوزة Preserve مع أمر Redim للحفاظ على البيانات المخزنة من قبل حيث يؤدي عدم استخدامها إلى إلغاء البيانات الحالية.

مثال:

يوضح المثال التالي تغيير حجم المصفوفة من عنصرين إلى أربع عناصر مع الحفاظ على البيانات المخزنة من قبل.

```
Dim day( )
Redim day(1)
day(0) = “Saturday”
day(1) = “Sunday”
Redim Preserve day(3)
day(2) = “Monday”
day(3) = “Tuesday”
```

1.7 استخدام الدالة Array ( )

عزيزي الدارس، تستخدم الدالة Array( ) لإنشاء مصفوفة ثم إسنادها إلى متغير حيث تستقبل هذه الدالة عناصر المصفوفة المراد إنشاؤها (مستخدما الفاصلة "," للفصل بين العناصر)، وعندئذ تنشأ الدالة Array( ) المصفوفة و إعادتها.

الصيغة العامة

```
Dim <ArrayNmae>
<ArrayNmae> = Array( <Array Members> )
```

مثال :

يوضح هذا المثال استخدام الدالة Array ( ) لإنشاء مصفوفة ذات بعد واحد والتي تحتوي على خمس عناصر رقمية.

```
Dim Num
Num= Array (50, 30, 30, 10, 33)
```

2.7 استخدام الدالة UBound ( )

تستخدم الدالة Ubound ( ) (اختصار للعبارة Upper Bound) لمعرفة قيمة فهرس آخر عنصر في المصفوفة، حيث يتمكن المبرمج من التعامل مع المصفوفة بشكل ديناميكي وكتابة شفرة عامة لا تتغير بتغير حجم المصفوفة، حيث يمكن تحديد حجم المصفوفة الحالي عن طريق زيادة الرقم 1 على القيمة العائدة من دالة UBound( ) (تذكر أن الفهرس يبدأ من صفر وليس من واحد).

الصيغة العامة

```
<Var> = UBound ( <ArrayName> )
```

مثال :

يوضح المثال التالي كيفية زيادة حجم المصفوفة Num بمقدار واحد حيث نسترجع قيمة فهرس آخر عنصر مستخدما الدالة Ubound ( ) ، ثم استخدام الأمر Redim لزيادة حجم المصفوفة Num بمقدار واحد.

```
NewSize= UBound( Num ) + 1
Redim Preserve Num( NewSize )
```

أسئلة تقويم ذاتي

1. ما فائدة الدالة Array( )؟
2. اكتب دالة getGrade( ) التي تستقبل درجة طالب في مادة ما (رقم صحيح) وتعيد تقديره طبقاً لقاعدة التحويل المعروفة.
3. أعد كتابة البرنامج التالي مستخدماً For…Next والدالة UBound.

8. عبارات التحكم في سير البرنامج

كما هو الحال مع معظم لغات البرمجة الأخرى، تحتوي لغة VBScript على عبارات عديدة تتيح لك التحكم في ترتيب تنفيذ الأوامر، يتم التحكم في ترتيب تنفيذ الأوامر باستخدام نوعين من العبارات: العبارات الشرطية (Conditional Statements)، وعبارات التكرار (Repetitive Statements) . يتم استخدام العبارات الشرطية لتنفيذ مجموعة معينة من الأوامر أو عدم تنفيذها وذلك طبقا ناتج شرط ما. أما عبارات التكرار تستخدم لتنفيذ مجموعة معينة من الأوامر عدداً محدداً من المرات أو إلى أن تنتفي صحة شرط معين، ويتم في الغالب الجمع بين العبارات الشرطية وعبارات التكرار معاً لتطوير برامج قوية ومتعددة الاستخدامات بأقل عدد من الأوامر.

9. أولا: العبارات الشرطية

(Conditional Statements)

1.9 عبارة IF. .Then

تستخدم هذه العبارة لتنفيذ مجموعة من الأوامر إذا تحقق شرطٌ ما (إذا كانت نتيجة اختباره صواب True) وإذا لم يتحقق هذا الشرط (إذا كانت نتيجة اختباره خطأ False)؛ فإن لغة VBScript ستتجاهل هذه الأوامر ولن تقوم بتنفيذها، وستستمر في تنفيذ العبارات التي تلي عبارة End If.

الصيغة العامة

```
If <condition> then
    <set  of statements>
End If
```

مثال :

يوضح المثال التالي استخدام عبارة IF/Then لإعادة قيمة Counter إلى صفر وذلك إذا وصلت القيمة إلى 99.

```
If  Counter = 99  then
    Counter = 0
End If
```

10. عبارة IF. .Then. .Else

تستخدم هذه العبارة لتنفيذ مجموعة من الأوامر (الموجودة بين الكلمة المحجوزة If والكلمة المحجوزة Else) إذا تحقق شرط ما، وتنفيذ مجموعة أخرى من الأوامر (الموجودة بين الكلمة المحجوزة Else والكلمة المحجوزة End IF) إذا لم تتحقق نتيجة هذا الشرط.

الصيغة العامة

```
If  <condition> then
    <Set 1 of statements>
Else
    <Set 2 of Statements>
End if
```

مثال :

يوضح المثال التالي استخدام عبارة F/Then/Else لإعادة قيمة المتغير Counter إلى صفر، وذلك إذا وصلت قيمته إلى 99 وزيادته بمقدار 1 إذا كانت قيمته أقل من ذلك.

```
If counter=99 then
   counter= 0
 Else
   counter = counter+1
End If
```

11. عبارة If. .Then. .Else If

تستخدم هذه العبارة لتنفيذ مجموعة من الأوامر إذا تحقق شرط ما، وإذا لم يتحقق، يتم اختبار الشرط الذي يليه، فإذا تحقق هذا الشرط يتم تنفيذ مجموعة الأوامر المصاحبة له، وإذا لم يتحقق، يتم اختبار الشرط الذي يليه، وهكذا. يجب ملاحظة أنه يتم اختبار الشروط بالترتيب أي إذا تحقق أحد الشروط الأولى، يتم تنفيذ مجموعة الأوامر المصاحبة له والخروج مباشرة (لا يتم التحقق من بقية الشروط).

لاحظ :

أن للمبرمج الحرية في إضافة أو عدم إضافة الجزء Else في آخر عبارة If. Then. .ElseIf ، حيث يتم تنفيذ الأوامر المصاحبة للكلمة المحجوزة Else إذا لم تتحقق جميع الشروط.

الصيغة العامة

```
If  <condition 1> then
    <Set 1 of statements>
Elseif <condition 2> then
     <Set 2 of Statements>
                :
Elseif <condition n> then
     <Set  n of Statements>
Else
     <Set  m of Statements>
End if
```

مثال :

يوضح المثال التالي كيفية تحويل درجة طالب في مادة دراسية (المخزنة في المتغير Score) إلى التقدير المناسب (المتغير Grade) بناء على قاعدة التحويل المعروفة، وإذا لم يحتوي المتغير Score على قيمة صحيحة يتم تخزين القيمة "error" في داخل المتغير Grade.

```
If Score <60 then
        Grade = "F"
ElseIf  Score >= 60 And Score <70  then
        Grade= "D"
ElseIf  Score >= 70 And Score <80  then
        Grade= "C"
ElseIf  Score >= 80 And Score <90  then
        Grade= "B"
ElseIf  Score >= 90 And Score < 100  then
        Grade= "A"
Else
        Grade= "error"
End If
```

1.11 العبارة Select Case

تقوم العبارة Select Case بتنفيذ مجموعة معينة من الأوامر حسب قيمة مصطلح معين، حيث تقوم لغة VBScript بحساب قيمة هذا المصطلح ثم مقارنته بالقيم المذكورة أمام عبارات Case. وعندما تتساوى إحدى قيم عبارات Case مع قيمة المصطلح، تقوم بتنفيذ الأوامر المصاحبة لعبارة Case. فإذا لم تجد لغة VBScript توافقاً في جميع عبارات Case المذكورة ، فإنها تتجاهل عبارات Case كلها وتبدأ في تنفيذ الأوامر المصاحبة للعبارة Case Else إن وجدت أو الانتقال مباشرة للأوامر التي تلي End Select.

لاحظ :

أن عبارة Select case حساسة لحالة الأحرف، ولذلك يجب أن تتوافق القيم النصية الناتجة عن المصطلح تماماً مع القيمة المحددة أمام عبارات Case.

الصيغة العامة

```
Select case <Expression>
        Case <value 1>
                <set 1 of statements>
           :
         Case < value n>
                <set n of statements>
         Case Else
                  <set m of statements>
End Select
```

مثال :

يوضح المثال التالي استخدام عبارة Select Case لاستدعاء إجراء اللغة المناسبة (عربي أو إنجليزي) وذلك بناء على قيمة متغير LangOption .

```
Select case LangOption
Case   1
        ArabicLang( )
 Case   2
        EnglishLang( )
Case else
        ErrorMess( )
End Select
```

12. ثانيا: عبارات التكرار (Repetitive Statements)

1.12 عبارة For. .Next

عزيزي الدارس، تتيح لك عبارة For/Next أن تكرر مجموعة معينة من الأوامر عدداً محدداً من المرات. فعلى سبيل المثال، ربما ترغب في إنشاء خمس مربعات نص على صفحة الويب، وبدلاً من كتابة الأوامر اللازمة لإنشاء مربع النص خمس مرات، يمكنك وضع هذه الأوامر داخل التكرار For/Next حيث يتم تنفيذها عدد المرات الذي تريده.

وعند إنشاء عبارة For/Next نستخدم متغيراً يعمل كعداد (Counter) للتكرار، وسوف تحدد القيمة التي سيبدأ منها العداد والقيمة التي سيتوقف عندها وقيمة الزيادة (أو قيمة النقص) في كل دورة للتكرار. ستقوم لغة VBScript بتنفيذ الأوامر مرة ثم تزيد (أو تقل) من قيمة العداد، وعندما تصل قيمة العداد إلى قيمة النهاية التي حددتها سينتهي عمل التكرار وتبدأ لغة VBScript في تنفيذ الأوامر التي تلي عبارة For/Next. وإذا لم تحدد مقدار الزيادة للقيمة المصاحبة للكلمة Step، فإن لغة VBScript ستزيد قيمة العداد بمقدار 1 في كل دورة للتكرار.

الصيغة العامة

```
For <Counter> = <Start> To <End> Step <Value>
        <set of statements>
Next
```

يوضح المثال التالي كيفية استخدام For/Next لحساب المتتالية الحسابية التالية:

1 + 2 + 3 + 4 + …….. +100

```
Dim i , s
s = 0
For i=1 to 100 Step 1
        S = s + i
Next
```

2.12 عبارة For Each. .Next

تستخدم عبارة For Each/Next للتعامل مع العناصر المحفوظة داخل مجموعة ما (Collection) مثل المصفوفات، حيث تقوم عبارة For Each/Next بتكرار معالجة كل عنصر في المجموعة بغض النظر عن عدد العناصر داخلها.

تمثل عبارة For Each/Next طريقة فعالة للوصول إلى المعلومات الموجودة في المجموعة عندما لا تعرف عدد العناصر التي تحتوي عليها هذه المجموعة. فعلى سبيل المثال، يمكنك استخدام عبارة For Each/Next للوصول إلى كل عنصر في مصفوفة ديناميكية.

عند استخدام هذه العبارة، لابد من تعريف المتغير لكي يمثل أحد عناصر المجموعة في التكرار الواحد، حيث سيتم استخدام هذا المتغير مع أوامر التكرار لمعالجة عناصر المجموعة.

الصيغة العامة

```
For Each <Element> in <Collection>
        <set of statements>
Next
```

مثال :

يوضح المثال التالي كيفية استخدام عبارة For Each/Next لتجميع عناصر مصفوفة رقمية وتخزينه داخل المتغير Sum.

```
Dim Elem , NumArr , Sum
Sum = 0
NumArr = Array(50, 10, 33, 40, 100)
For each Elem in NumArr
        Sum = Sum  + Elem
Next
```

3.12 عبارة While. .wend

تستخدم عبارة While/wend لتكرار معالجة مجموعة من الأوامر طالما نتيجة اختبار شرط التكرار صواب (True)، وتتوقف عن تكرار معالجة هذه الأوامر عندما تصبح نتيجة اختبار شرط التكرار خاطئة (False)، ويجب أن تلاحظ أن هذه العبارة تختبر أولا شرط التكرار ثم تنفذ الأوامر، أي ربما لا تنفذ هذه الأوامر أصلاً، وذلك إذا كانت نتيجة أول اختبار لشرط التكرار خاطئة.

الصيغة العامة

```
While <Condition>
        <set of statements>
wend
```

مثال:

يوضح المثال التالي كيفية استخدام While/wend لحساب المتتالية الحسابية التالية:

0 + 5+ 10+ 15 + …….. + 1000

```
Dim i , s
s = 0
i = 0
while  i < = 1000
        s = s + i
    i = i + 5
wend
```

4.12 عبارة Do. .Loop while

تستخدم عبارة Do/Loop while لتكرار معالجة مجموعة من الأوامر طالما نتيجة اختبار شرط التكرار صواب (True)، وتتوقف عن تكرار معالجة هذه الأوامر عندما تصبح نتيجة اختبار شرط التكرار خاطئة (False). ويجب أن تلاحظ أن هذه العبارة تنفذ الأوامر أولا ثم تختبر شرط التكرار ، ومعنى ذلك أن هذه الأوامر ستنفذ على الأقل مرة واحدة حتى لو كانت نتيجة أول اختبار لشرط التكرار خاطئة.

الصيغة العامة

```
Do
        <set of statements>
Loop while  <Condition>
```

أسئلة تقويم ذاتي



ما الفرق بين عبارة التكرار while .. wend
وعبارة التكرار Do .. loop while ؟

13. الإجراءات والدوال (Procedures & Functions)

تتكون كل من الدالة (Function) والإجراء (Procedure) من مجموعة من الأوامر والتي تقوم بأداء مهمة محددة عند استدعائها، حيث يمكن لكل من الدالة والإجراء استقبال بيانات (Parameters) عند استدعائها لإجراء المعالجة عليها، ومن الجدير بالذكر أن فكرة عمل واستخدام كل من الدالة والإجراء واحدة، والفرق الوحيد بينهما هو أن الدالة تعيد قيمة (نتيجة المعالجة) بعد أداء مهمتها أما الإجراء لا يعيد قيمة بعد أداء مهمته.

يتيح لك استخدام كل من الدوال والإجراءات تنظيم البرنامج إلى وحدات أصغر (دوال وإجراءات) والتي تتميز بسهولة إدارتها بالإضافة إلى سهولة عملية اكتشاف الأخطاء بها. هذا بالإضافة إلى إعادة استخدام كل من هذه الدوال والإجراءات أكثر من مرة في البرنامج الحالي أو مع برامج أخرى.

فعلى سبيل المثال، يمكن أن تكتب مجموعة من الأوامر لعرض رسالة تحذير عندما يدخل المستخدم بيانات غير صالحة في النموذج، و بدلاً من كتابة هذه الأوامر لعرض الرسالة في كل مرة تحتاج إليها، يمكنك أن تضعها في إجراء فرعي ثم تستدعي هذا الإجراء في كل مرة تحتاج إليه.

ويطلق على هذه النوعية من الدوال (والإجراءات) Programmer Defined Functions أي دوال معرفة من قبل المبرمج وليست معرفة من قبل مطوري لغة البرمجة، حيث يتم تعريفها بواسطة المبرمج عند عدم وجود دالة (أو إجراء) جاهزة تؤدي المهمة المطلوبة.

1.13 الإجراء الفرعي (Procedure)

كما ذكرنا سابقاً فإن الإجراء الفرعي يجمع مجموعة من الأوامر لأداء مهمة محددة ولا يعيد قيمة عند استدعائهِ، ويجب تعريف الإجراء مرة أولا قبل استدعائهِ (أي كتابة الأوامر المكونة للإجراء) والتي تنفذ بالتتالي عند استدعائهِ

الصيغة العامة لتعريف الأجراء

```
Sub <ProcedureName> ( [ <parameterList> ] )
        <set of statements>
End Sub
```

يبدأ تعريف الإجراء بالكلمة Sub، ثم اسم الإجراء (ProcedureName) ، ثم قائمة اختيارية من المدخلات (parameterList) ، ثم الأوامر المكونة للإجراء ، ثم كلمة نهاية الإجراء End Sub. لاحظ أن استخدام الأقواس [ ] داخل الصيغة العامة السابقة للتعبير عن الأشياء الاختيارية (فمثلا أن قائمة المعاملات اختيارية).

مثال :

يوضح المثال التالي تعريف إجراء يقوم بتحويل درجة سيلزس إلى درجة فهرنهيت حيث يستقبل قيمة درجة سيلزس celTemp، ثم يقوم بتحويلها إلى درجة FahrenTemp والذي يمكن طباعتها بعد ذلك.

```
Sub tempConvert (celTemp)
Dim FahrenTemp
fahrenTemp= celTemp* 9/5 + 32
                ‘ Then you can print fahrenTemp
End sub
```

استدعاء الإجراء الفرعي

بعد تعريف الإجراء الفرعي، ستحتاج إلى استدعائه. واستدعاء الإجراء يعني أن تطلب من لغة VBScript أن تقوم بتنفيذ الأوامر الموجودة داخله، ولكي يتم استدعاء الإجراء، يجب استخدام الكلمة المحجوزة Call، ثم اسم الإجراء.

الصيغة العامة لاستدعاء الأجراء

```
Call <procedureName> ( [ <parametersValues> ] )
```

توضح الصيغة العامة أن استدعاء إجراء فرعي يبدأ بالأمر Call ثم اسم الإجراء متبوعا بقائمة من المدخلات الاختيارية والتي سيتم إسنادها لقائمة المتغيرات المذكورة مع تعريف الإجراء داخل الأقواس والتي يجب فصلها بالفاصلة (,).

مثال :

يوضح المثال التالي استدعاء إجراء المثال السابق ( )tempConvert والذي يحول من درجة سيلزس إلى درجة فهرنهيت حيث مرر لهذا الإجراء القيمة 10 ومن المنتظر أن يطبع هذا الإجراء القيمة 50.

```
Call tempConvert(10)
```

2.13 الدوال (Functions)

كما ذكرنا سابقاً فإن الدالة تحتوي على مجموعة من الأوامر لأداء مهمة محددة وتعيد ناتج المعالجة (هذا هو الفرق الوحيد بينها وبين الإجراء). يجب تعريف الدالة قبل استدعائها (أي كتابة الأوامر المكونة للدالة) والتي ستنفذ بالتتالي عند استدعائها.

الصيغة العامة لتعريف الدالة

```
Function <FunctionName> ( [ <parameterList> ] )
        <set of statements>
        <FunctionName> = <processingResult>
End Function
```

يبدأ تعريف الدالة بالكلمة Function، ثم اسم الدالة (*FunctionName*)، ثم قائمة اختيارية من المدخلات (*parameterList*)، ثم الأوامر المكونة للدالة، ثم أمر إعادة نتيجة المعالجة (إسناد النتيجة النهائية لاسم الدالة)، ثم كلمة نهاية الدالة End Function.

مثال: يوضح المثال التالي تعريف دالة تقوم بنفس وظيفة الإجراء السابق، ولكن تعيد نتيجة تحويل درجة الحرارة (fahrenTemp) بدلا من طباعتها.

```
Function tempConvert (celTemp)
    Dim fahrenTemp
      fahrenTemp= celTemp* 9/5 + 32
    tempConvert = fahrenTemp
End Function
```

أسئلة تقويم ذاتي



7. ما الفرق بين الدالة والإجراء؟
8. اكتب الدالة Area(r) بلغة VBScript والتي تستقبل نصف قطر دائرة وتعيد المساحة

استدعاء الدالة

بعد تعريف الدالة، يتم استدعاؤها، أي أن تطلب من لغة VBScript أن تقوم بتنفيذ الأوامر الموجودة داخلها، ولكي يتم استدعاء الدالة، اكتب اسم المتغير الذي سيستقبل ناتج معالجة الدالة، ثم معامل الإسناد (=)، ثم اسم الدالة متبوعا بقائمة اختيارية من القيم الممررة إلى الدالة.

الصيغة العامة

```
<Var> = <functionName> ( [ <parametersValues> ] )
```

مثال :

يوضح المثال التالي استدعاء دالة المثال السابق ( )tempConvert التي تحول من درجة الحرارة سيلزيس إلى درجة الحرارة فهر نهيت وتعيد الناتج.

```
tempF = tempConvert(10)
```

لاحظ :

- يجب أن تتأكد من أن صحة هجاء اسم الإجراء (أو الدالة) عند استدعائه يجب أن يتوافق مع هجائه عند تعريفه.
- من الممكن أن تستدعي إجراءاً فرعياً (أو دالة) من داخل إجراء فرعي آخر. وعند استدعاء إجراء من داخل آخر، يجب أن تكون حريصاً ألا يدخل البرنامج في حلقة مفرغة بسبب استدعاء الإجراءات لبعضها بصورة مستمرة.

3.13 استخدام عبارتي Exit sub & Exit function

في بعض الحالات، لن ترغب في تنفيذ جميع الأوامر الموجودة في الإجراء الفرعي (أو الدالة)، وفي هذه الحالات يمكنك استخدام العبارة Exit Sub لإيقاف تنفيذ ما تبقى من أوامر في الإجراء والخروج منه فوراً (أو العبارة Exit function لإيقاف تنفيذ الدالة). فعلى سبيل المثال، افترض أنه يوجد لدينا إجراء فرعي يعرض بعض المعلومات من قاعدة

بيانات، فإذا حدث خطأ أثناء القراءة من قاعدة البيانات، يمكنك أن تعرض رسالة خطأ للمستخدم وتخرج من الإجراء فوراً.

وعندما تقابل لغة VBScript عبارة Exit Sub (أو عبارة Exit function مع الدالة)، فإنها ستتوقف عن تنفيذ الإجراء (أو الدالة) وتتجاهل بقية الأوامر التي تلي عبارة Exit Sub (أو عبارة Exit function مع الدالة) وتبدأ في تنفيذ الأوامر التي تلي استدعاء الإجراء (أو الدالة).

مثال :

يوضح المثال التالي الدالة Divide( ) والتي تستخدم لحساب ناتج قسمة المتغير X على المتغير Y ثم إعادة الناتج، ولكن يجب الانتباه إلى قيمة Y حيث يمكن أن تساوي صفراً، وهذا يؤدي إلى خطأ ولذلك تتم الاستفادة من الأمر Exit Function للخروج من الدالة عند حدوث هذه الحالة.

```
Function Divide (X , Y)
        Dim Z
        If Y = 0 then
                ‘ Print error message
                Exit function
        End if
        Z= x/y
        Divide = Z
End Function
```

14. التعرف على دوال وإجراءات مكتبة لغة VBScript

تتضمن لغة VBScript مجموعة من الدوال الجاهزة (معرفة من قبل مطوري اللغة) والتي تسمى Built-in functions، حيث تتنوع هذه الدوال والتي يمكن استدعاؤها للاستفادة من مهامها مثل دوال التاريخ والوقت، ودوال معالجة النصوص، ودوال التحقق من نوع البيانات، سيتم فيما يلي شرح هذه الدوال:

15. دوال الوقت والتاريخ

تتضمن مكتبة لغة VBScript مجموعة من الدوال الجاهزة التي تتيح لك العمل مع التاريخ والوقت على صفحة الويب، فعلى سبيل المثال ربما تحتاج إلى عرض التاريخ والوقت الحاليين على صفحة الويب، أو تقوم بعملية حسابية معقدة على التواريخ مثل حساب عدد الأيام التي مرت منذ آخر تحديث لصفحة الويب.

يوضح جدول رقم 4-5 قائمة بأهم دوال الوقت والتاريخ الشائعة في لغة VBScript، حيث يوضح الجدول الصيغة العامة للدالة، ومثال لكل دالة، ووصف الدالة

جدول 4-5 : وصف لأهم دوال الوقت والتاريخ في لغة VBScript

الصيغة العامة	الوصف	مثال
Date	تعيد التاريخ الحالي للنظام.	لاسترجاع التاريخ الحالي وتخزينه داخل المتغير MyDate: MyDate = Date
DateAdd(<interval>, <number>, <date>)	تضيف فترة زمنية (number) بناء على نوع الفترة (interval) إلى التاريخ الذي تحدده (date) ، فإذا كان نوع الفترة يُساوى "d" سيضيف أياماً، أما إذا ساوت "m" سيضيف شهوراً، أما إذا ساوت "yyyy" سيضيف سنين.	لإضافة 15 يوماً على التاريخ الحالي وتخزين الناتج بالمتغير aDate: aDate = DateAdd("d", 15, Date)
DateDiff(<interval>, <date1>, <date2>)	تحسب الفترة بين تاريخين وتعيد الناتج بناء على نوع الفترة (interval)، فإذا كان نوع الفترة يُساوى "d" سيعيد أياماً، أما إذا ساوت "m" سيعيد شهوراً، أما إذا ساوت "yyyy" سيعيد سنين.	لحساب الأيام بين التاريخ theDate والتاريخ الحالي، وتخزين الناتج بالمتغير Days: Days = DateDiff("d", Date, theDate)
Now	تعيد التاريخ والوقت الحالي للنظام.	تعيد التاريخ والوقت الحالي للنظام، وتخزين الناتج بالمتغير

:MyDateTime `MyDateTime = Now`		
تعود برقم اليوم الحالي. `Day(Date)`	تعود برقم اليوم في الشهر بين 1 و 31 .	`Day(<var>)`
تعود برقم الشهر الحالي. `Month(Date)`	تعود برقم الشهر في السنة بين 1 و 12 .	`Month(<var>)`
تعود برقم السنة الحالية. `Year(Date)`	تعود بالسنة لتاريخ ما.	`Year(<var>)`
تعود بالوقت الحالي. Time	تعود بالوقت الحالي.	`Time`
تعود بالساعة الحالية. Hour(Now)	تعود بالساعة بين 0 و 24 للمتغير var.	`Hour(<var>)`
تعود بالدقائق الحالية. `Minute(Now)`	تعود بالدقائق بين 0 و 59 للمتغير var.	`Minute(<var>)`
تعود بالثواني الحالية. Second(Now)	تعود بالثواني بين 0 و 59 للمتغير var.	`Second(<var>)`

1.15 دوال معالجة النصوص (String Manipulation Functions)

يمكن أن تحتوي سلسلة الحروف (String) على حروف، وأرقام، وحروف خاصة شائعة مثل الحروف @ , & , %، كما يمكن أيضا أن تكون خالية أو تحتوي على حروف لا تظهر فوق لوحة المفاتيح مثل رمز حقوق النشر ©.

تتيح لك دوال معالجة سلاسل الحروف أن تتعامل مع المعلومات الموجودة في هذه السلاسل، فعلى سبيل المثال يمكنك استخدام دوال سلاسل الحروف لكي تتعرف على

عدد الحروف التي أدخلها المستخدم في مربع نص يتواجد داخل نموذج ما، أو لتحويل سلسلة الحروف إلى حروف كبيرة (Capital Letters) أو صغيرة (Small Letters)، ومن الممكن أيضاً أن تستخدم دوال سلاسل الحروف لكي تستخلص البيانات من أحد العناصر الموجودة داخل النموذج. و أخيراً إنَّ استخدام دوال سلاسل الحروف يوفر عليك الوقت لأنه يتيح لك التحقق من صحة البيانات التي أدخلها المستخدم قبل إرسالها إلى خادم الويب.

يوضح جدول رقم 5-5 قائمة بأهم دوال معالجة النصوص الشائعة في لغة VBScript، حيث يوضح الجدول الصيغة العامة لكل دالة، ومثال لكل دالة، ووصف الدالة.

جدول 5-5 : وصف لأهم دوال معالجة النصوص في لغة VBScript

الصيغة العامة	الوصف	مثال
LCase(<var>)	تحول النص (var) إلى حروف صغيرة.	LCase (“GoodBye”) الناتج : goodbye
UCase(<var>)	تحول النص (var) إلى حروف كبيرة.	UCase (”GoodBye”) الناتج: GOODBYE
Rigth(<var>, <no>)	تستخلص عدداً من الحروف (no) بدءاً من الجانب الأيمن من السلسلة (var).	Right (“John Smith”,5) الناتج: Smith
Left(<var>, <no>)	تستخلص عدداً من الحروف (no) بدءاً من الجانب الأيسر من السلسلة (var).	Left (“John Smith”, 4) الناتج: Jonn
Len(<var>)	تحدد عدد الحروف داخل السلسلة (var)	Len (“John Smith”) الناتج: 10

Trim(<var>)	تزيل المسافات الخالية من بداية ونهاية النص (var).	Trim (“ John Smith “) الناتج: “Jone Smith”
InStr(<var1>, <var2>)	تحدد موقع سلسلة حروف (var2) داخل نص (var1) بدءاً من الطرف الأيسر للسلسلة.	Instr(“GOOD WOOD”, ”od”) الناتج: 3
Chr(<no>)	تحدد الحرف المقابل للرقم الذي تمرره لها (no).	Chr (65) الناتج: A
RePlace(<var1> , <var2>, <var3>)	تستبدل سلسلة حروف (var2) بأخرى (var3) داخل سلسلة الحروف الاصلية (var1).	RePlace(“My Pen”, “My”, “ ”) الناتج: Pen

2.15 دوال التحقق من نوع البيانات

تقدم لغة VBScript مجموعة من الدوال الجاهزة والتي تمكن المبرمج من التحقق من نوع بيانات المتغير الممرر لها، يوضح الجدول رقم 6-5 قائمة بأهم دوال التحقق من نوع البيانات الشائعة في لغة VBScript، حيث يوضح الجدول الصيغة العامة لكل دالة، ومثال لكل دالة، ووصف الدالة.

جدول 6-5 وصف لأهم دوال التحقق من نوع البيانات في لغة VBScript

الصيغة العامة	الوصف	مثال
IsArray (<var>)	تعيد قيمة True إذا كان البيان الفرعي للمتغير Var مصفوفة وتعيد False بخلاف ذلك.	MyArray(0) = "Sunday" MyArray(1) = "Monday" IsArray(MyArray) الناتج: True
IsNumeric(<var>)	تختبر إن كانت سلسلة الحروف الممررة لها (var) تحتوي على رقم أم لا.	IsNumeric (“12”)
IsDate(<var>)	تختبر إن كانت سلسلة الحروف الممررة لها (var) تحتوي على تاريخ أم لا.	IsDate (“3/27/00”) الناتج: True
IsEmpty (<var>)	تختبر إن كانت سلسلة الحروف الممررة لها (var) فارغة أم لا.	IsEmpty ("") الناتج: True

أسئلة تقويم ذاتي



1. عرف استخدام الدوال التالية:

1.Trim( )
2. Space()
3. UcaseS( )
4. IsDate( )
5. Time
6. Replace( )

الخلاصة

في هذه الوحدة عزيزي الدارس، قمنا بدراسة إنشاء موقع إنترنت ديناميكي (Dynamic Web Site) والذي تتغير محتوياته من حين لآخر بناء على تغير مصدر بيانات الموقع داخل قاعدة بيانات الموقع ، يجب عليك أن تتعلم لغة برمجة نصية (Scripting Language) والتي تمدك بالعديد من الأوامر التي تمكنك من كتابة هذه البرامج ، وبما أن لغة البرمجة الافتراضية لدى خادم الويب داخل تقنية Active Server Pages هي لغة VBScript (وتنطق في بي سكريبت) ، حيث تم التركيز على تعلم إمكانيات اللغة من أوامر (Instructions) وإجراءات ودوال (Procedures & Functions)

لمحة مسبقة عن الوحدة التالية

عزيزي الدارس، في وحدتنا التالية سوف نركز على المعالجة لدى الخادم (Server-Side Programming)، والتي تمكن المبرمج من تطوير تطبيقات التجارة الإلكترونية (E-commerce Applications) وغيرها من تطبيقات الإنترنت الهامة، وسوف نشرح في الوحدة القادمة كيفية التعامل مع قواعد البيانات، والتي لا يخلو أي تطبيق من الاحتياج إلى وجودها

إجابات التدريبات

تدريب (1)

Factor5 = 120

تدريب (2)

تحول النص الموجود داخل المتغير Var إلى حروف كبيرة Capital.

تزيل المسافات الخالية من بداية ونهاية النص الموجود داخل المتغير Var .

تعيد عدد حروف سلسلة الحروف الموجودة داخل المتغير Var .

تدريب (3)

False

المراجع

[1] عبدالكريم العقيل، "المخطط الأزرق لبناء صفحات ويب ديناميكية"، مكتبة جرير، الرياض، 2001م.

[2] د. أحمد أحمد شعبان دسوقى، أ. د. السيد محمود الربيعي، "تقنيات وبرمجة الإنترنت" مكتبة الرشد، الرياض، 2004م.

[3] د. عبد القادر عبد الله الفنتوخ، "الإنترنت للمستخدم العربي"، مكتبة العبيكان، الرياض، 2001م.

[4] د. إبراهيم فتحي معوض، "أساسيات برمجة الإنترنت"، معهد العالمية للحاسب والتقنية، 2005م.


الوحدة السادسة
البرمجة لدى الخادم
(Server-Side Programming

محتويات الوحدة

الموضوع	رقم الصفحة
المقدمة	175
تمهيد	175
الأهداف	175
1.المعالجة لدى الخادم ASP	176
1.1 خادم الويب IIS	177
2. محتويات ملف ASP	179
3. تعريف لغة المعالجة لدى خادم الويب	180
4. مكتبة ASP	181
5. الدالة ( )Redirect	186
6. الدالة ( )END	187
7. الكائن Request	192
8. الكائن Session	198
9. الكائن Application	203
10. تأمين متغيرات التطبيق	204
11. إزالة معلومات الجلسة والتطبيق	206
12. التعامل مع ملف Global.asa	207
الخلاصة	211
لمحة مسبقة	211
اجابات التدريبات	212
المراجع	214

المقدمة

تمهيد

ناقشنا في الوحدة الثالثة والرابعة كيفية المعالجة لدى عميل الويب (Client-Side Programming) والتي تستخدم لتغيير مظهر صفحات الويب بعد تحميلها، وتستخدم أيضا للتأكد من صحة البيانات المستقبلة من المستخدم قبل إرسالها لخادم الويب.

وسوف نركز في هذه الوحدة على المعالجة لدى الخادم (Server-Side Programming) والتي تمكن المبرمج من تطوير تطبيقات التجارة الإلكترونية (E-commerce Applications) وغيرها من تطبيقات الإنترنت الهامة، وسوف نشرح في الوحدة القادمة كيفية التعامل مع قواعد البيانات والتي لا يخلو أي تطبيق من الاحتياج إلى وجودها.

أهداف الوحدة



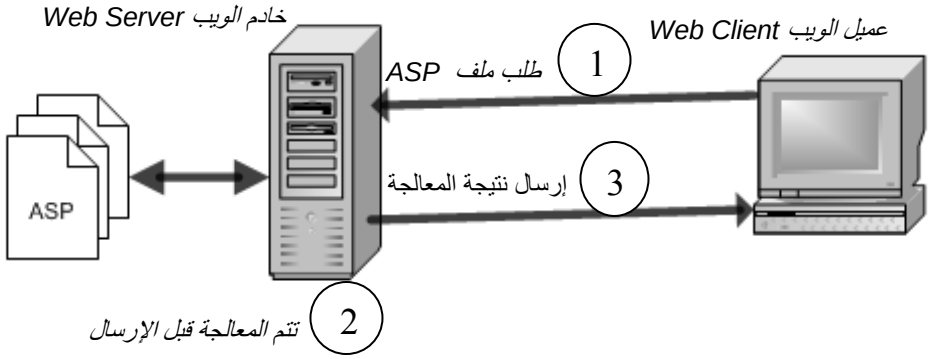
عزيزي الدارس، بنهاية دراسة هذه الوحدة ينبغي أن تكون قادرا على أن:

1. **تعرّف** لغة المعالجة لدى خادم الويب.
2. **تبين** تأمين متغيرات التطبيق.
3. **تتعامل مع** مكتبة ASP.
4. تتعرف على كيفية إزالة معلومات الجلسة والتطبيق.

1.المعالجة لدى الخادم ASP

عزيزي الدارس، تتم المعالجة لدى الخادم بواسطة تقنية ASP (اختصار لعبارة Active Server Pages) والتي تم تطويرها من قبل شركة Microsoft ، تقدم هذه التقنية مكتبة من الكائنات (Objects) والتي تمكن المبرمج من تطوير تطبيقات الإنترنت.

يتكون تطبيق الإنترنت (Web-Based Applications) من مجموعة من البرامج الفرعية والتي تخزن في ملفات من النوع ASP، حيث يحتوي كل ملف على برنامج فرعي يتم استدعاؤه عند الحاجة إليه، وعند استدعاء هذه الملفات بواسطة عميل الويب (متصفح الإنترنت)، ستتم معالجة هذه الملفات أولاً لدى خادم الويب بواسطة برنامج محرك تنفيذي للبرامج النصية (Scripting Engine) يوجد داخل خادم الويب كما هو واضح في شكل رقم 1-6.



شكل 1 -6 المعالجة لدى خادم الويب (Server-Side Scripting)

عزيزي الدارس، يجب أن تلاحظ الملاحظات التالية:

- يجب أن نطلق الاسم الممتد asp للملفات التي من النوع ASP لكي تعالج قبل إرسالها إلى عميل الويب (متصفح الإنترنت).
- يمكن أن تحتوي ملفات ASP على أكثر من نوع من شفر الأوامر:

شفرة ASP والتي تعالج لدى الخادم وإرسال الناتج لعميل الويب.

▪ شفرة HTML أو شفرة DHTML والتي ترسل كما هي لكي تفسر وتظهر بواسطة عميل الويب.

وبالرغم من أن لغة JavaScript يمكن أن تستخدم للمعالجة لدى خادم الويب ، ولكن تمثل لغة VBScript اللغة الافتراضية للمعالجة لدى خادم الويب بواسطة تقنية ASP ، ومن ثم تستخدم بكثرة لإنشاء مثل تلك التطبيقات.

1.1خادم الويب IIS

عزيزي الدارس، تتطلب معالجة أوامر ASP وجود خادم ويب على الجهاز الذي سيستضيف موقع تطبيق الإنترنت، حيث يجب عمل الخطوات اللازمة لاستضافة هذا الموقع على خادم الويب قبل استخدامه بواسطة زائري الموقع (راجع الملحق " أ " لمعرفة خطوات استضافة الموقع على خادم الويب)، وكما ذكرنا سابقاً يوجد نوعان من خادم الويب اللذان يعملان تحت بيئة Windows:

▪ خادم ويب Personal Web Manager (PWS) والذي يعمل تحت بيئة Windows Me وما أقل ، ولذلك لم يعد يستخدم لعدم استخدام نظم تشغيل أقل من Windows 2000.

▪خادم ويب Internet Information Services (IIS) والذي يعمل تحت بيئة Windows 2000 وما أعلى، ولكثرة استخدام نظام التشغيل Windows 2000 ، ونظام التشغيل Windows XP ، تم تخصيص الملحق (أ) لتوضيح تفاصيل كل من عملية تركيب خادم IIS وكيفية عمل استضافة لمواقع الويب عليه.

خطأ شائع. . .

يحاول كثير من مطوري مواقع الويب أن يجرب ملفات ASP دون عمل استضافة للموقع الذي يريد أن يختبره، ومن ثم يحاول أن يستدعيها بفتحها مباشرة من خلال عميل الويب عن طريق القائمة File ثم الأمر Open مثلما نعمل عند تجربة ملفات المواقع

الثابتة. فكيف تنفذ أوامر ASP دون استضافة هذه المواقع على خادم الويب المسئول عن تنفيذها؟

تتم معالجة أوامر ASP على خادم الويب ثم إرسال نتائج المعالجة على شكل رموز HTML إلى مستعرض الويب لكي يفسرها ثم يعرضها للمستخدم، يستطيع المستخدمون استعراض رموز HTML لصفحة ASP ولكنهم لن يتمكنوا من رؤية أوامر ASP الموجودة بها لأنها لم تعد موجودة حيث تم تنفيذها ولا يوجد الآن سوى نتيجة هذا التنفيذ.

وبعد عمل استضافة لموقع الويب الذي تم تطويره، يجب اختبار تنفيذ صفحات ASP المخزنة داخل هذا الموقع قبل نشرها على شبكة الإنترنت، ولاختبار هذه الصفحات، يجب زيارة الموقع مستخدما عميل الويب (متصفح الإنترنت) ثم استدعاء هذه الملفات حيث يتم كتابة عنوان صفحة ASP المراد اختبارها داخل قائمة عنوان متصفح الإنترنت ثم الضغط على مفتاح Enter ، يوضح الأمر التالي الصيغة العامة لكتابة عنوان صفحة ASP المراد استدعاؤها من خادم الويب (راجع الملحق (أ)).

```
HTTP::///<ServerName>/ <VirtualDirName>/<FileName>
```

- يمثل <ServerName> اسم خادم الويب الذي يستضيف موقع الإنترنت، وإذا كان خادم الويب يوجد على نفس الجهاز الذي يحتوي على متصفح الإنترنت فإنه يمكن استخدام الاسم Localhost والذي يشير إلى عنوان خادم الويب المعرف على الجهاز الحالي.
- يمثل <VirtualDirName> الاسم المستعار للموقع الذي تم استخدامه عند عمل استضافة للموقع على خادم الويب.
- يمثل <FileName> اسم الملف المراد عرضه على متصفح الإنترنت سواء أكان ملف ASP أو ملف HTML.

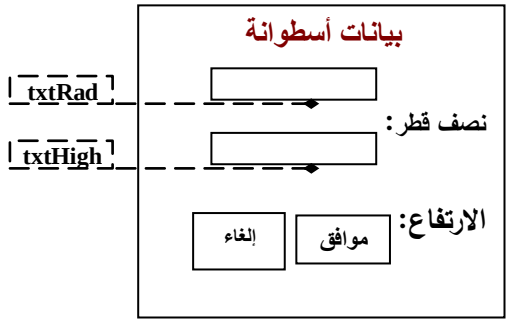
تدريب (1)

1. اكتب شفرة ASP المسؤولة عن استقبال بيانات النموذج التالي وطباعة ناتج حساب حجم اسطوانة، علما بأن معادلة حساب حجم الأسطوانة هي:

Volume = 3.14 * High * rad^2

ملاحظة:

- المتغير Rad يمثل نصف قطر قاعدة الأسطوانة، والمتغير High يمثل ارتفاع الأسطوانة.
- التزم بأسماء الحقول النصية الموضحة على النموذج.



2. محتويات ملف ASP

عزيزي الدارس، يمكن أن يحتوي ملف ASP على نوع شفرة أو أكثر ، وذلك حسب الحاجة، وفيما يلي يتم سرد هذه الشفرات:

- شفرة ASP : يجب أن يحتوي ملف ASP على أوامر من النوع ASP لكي تنفذ على خادم الويب قبل إرسالها، وإن لم تحتوي على أوامر ASP فمن الأفضل أن تعطى اسماً ممتداً *.htm لرفع أداء الموقع حيث لا يتم عمل مسح شامل لمحتويات الملف بواسطة خادم الويب لتنفيذ ما بداخله من أوامر ASP ويرسل مباشرة إلى عميل الويب، يتم تحديد أوامر ASP بين العلامات <% %> في أي مكان داخل الملف حيث تنفذ هذه الأوامر من قبل الخادم ووضع ناتج التنفيذ موضع هذه الأوامر. ويمكن إدراج أوامر ASP في أي

مكان تريده داخل صفحة الويب. وإذا كانت الأوامر ستعرض معلومات في نافذة مستعرض الويب لدى المستخدم، يجب إدراجها بين علامتي <BODY> </BODY> لأننا نعلم أن متصفح الإنترنت يعرض ما بداخل هذه العلامات فقط داخل صفحة الويب.

- شفرة HTML : ربما يحتوي ملف ASP على شفرة HTML والتي ترسل كما هي إلى عميل الويب دون أي تغيير.
- دوال وإجراءات تستدعى لدى الخادم: يمكن أن يحتوي ملف ASP على دوال وإجراءات تعرف من قبل المبرمج لاستدعائها وتنفيذها على خادم الويب فقط. وذلك بإضافة هذه الدوال والإجراءات داخل علامة Script، ولكن يجب أن تكون الصفة Runat تساوى Server التي تحدد مكان تنفيذ هذه الدوال (Script Runat = "> Server" <).
- دوال وإجراءات تستدعى لدى العميل: يمكن أن يحتوي ملف ASP على دوال وإجراءات تعرف من قبل المبرمج لاستدعائها وتنفيذها على عميل الويب، وذلك بإضافة علامة < Script > دون إضافة السمة Runat لأن القيمة الافتراضية لها تساوي Client (أي أن عميل الويب يمثل الجهة الافتراضية لتنفيذ هذه الدوال).

3. تعريف لغة المعالجة لدى خادم الويب

عزيزي الدارس، تسمح تقنية ASP للمبرمج أن يطور ملفات ASP مستخدما لغتين: لغة VBScript وهي اللغة الافتراضية، ولغة JavaScript، وهما نفس اللغات التي يمكن تنفيذهما لدى عميل الويب، حيث كل من خادم الويب وعميل الويب يحتويان على محرك التنفيذ لكل من هاتين اللغتين.

يجب على مطوري صفحات ASP أن يخبر خادم الويب أي اللغات سوف تستخدم في كتابة شفرة ASP وذلك باستخدام الصيغة التالية ، حيث يجب أن يكتب هذا الأمر في بداية صفحة ASP.

Language= “LanguageName” %>@< %

فعلى سبيل المثال ، إذا لم يكتب المبرمج هذا الأمر سيمرر خادم الويب جميع أوامر ASP المتواجدة بين <% %> إلى محرك التنفيذ الخاص بلغة VBScript لأنها هي اللغة الافتراضية، أما إذا أراد أن يستخدم لغة JavaScript، فيجب عليه كتابة الأمر التالي في بداية الصفحة.

```
< % @Language= "JavaScript" %>
```

4. مكتبة ASP

عزيزي الدارس، كما ذكرنا سابقا فإن تقنية ASP تقدم مكتبة من الكائنات (Objects) التي تحتوي على دوال (Methods)، وصفات (Attributes)، ومجموعات (Collections) والتي تمكن المبرمج من أن يطور تطبيقات ويب قوية مثل تطبيقات التجارة الإلكترونية. تحتوي مكتبة ASP على خمس كائنات رئيسية:

1. كائن Response والذي يستخدم لإرسال نتائج المعالجة والرسائل من خادم الويب إلى عميل الويب.
2. كائن Request والذي يستخدم لاستقبال بيانات لدى خادم الويب من عميل الويب، وذلك عن طريق النماذج وغيرها.
3. كائن Session والذي يستخدم لتعريف مجموعة من المتغيرات على مستوى كل مستخدم ، ومن ثم يمكن تخزين بيانات كل مستخدم على حده.
4. كائن Application والذي يستخدم لتعريف مجموعة من المتغيرات العامة على مستوى جميع المستخدمين مثل متغير عدد الزوار الكلي للموقع.
5. كائن Server والذي يستخدم لإنشاء متغيرات من الكائنات (Objects)، واستقبال معلومات عن بيئة عميل المستخدم.

سنوضح فيما يلي أهم هذه الكائنات من خلال شرح أهم الدوال والصفات والمجموعات التي تحتويها تلك الكائنات.

1- كائن Response

عزيزي الدارس، يقدم الكائن Response مجموعة من الدوال (Methods)، والصفات (Attributes) التي تمكن المبرمج من التحكم في المعلومات المرسلة لعميل الويب. وفيما يلي سنوضح هذه الدوال والصفات.

الدالة ( )Write

تستخدم الدالة ( )Write لإرسال معلومات ورسائل من خادم الويب إلى عميل الويب.

الصيغة العامة

```
<% Response.write (“<Message>”)  %>
```

عزيزي الدارس لاحظ التالي:

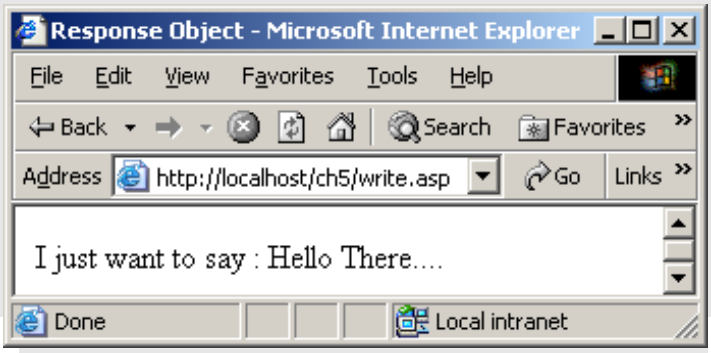
- أنه تم الفصل بين الكائن Response والدالة ( )Writeبالمعامل "." حيث يجب اتباع هذه القاعدة عند استدعاء أحد عناصر أي كائن مثل الدالة، الصفة ، والمجموعة.
- يجب أن تكون الرسالة المرسلة لعميل الويب نصية وتحدد بعلامتي التنصيص " ".

مثال:

يوضح المثال التالي (شكل رقم 6-2) استخدام الدالة ( )Write لإرسال رسالة ترحيب “Hello There” للمستخدم والتي تلصق بجوار عبارة ” :I Just want to Say“.

```
<html>
<head>
<title>Response Object</title>
</head>

<body>
I just want to say :  <% Response.Write(" Hello
There....") %>
</body>
</html>
```



شكل 6-2 : مثال لاستخدام دالة ( )Write لإظهار رسالة ترحيبية

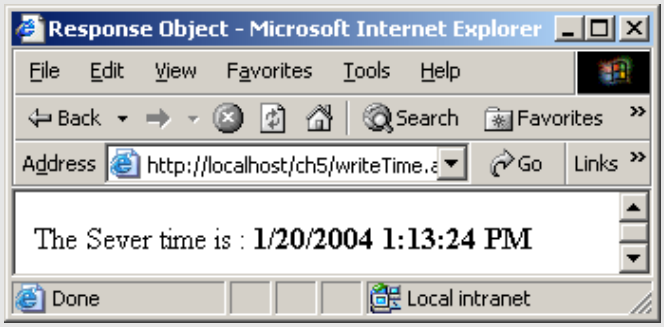
مثال :

يوضح الشكل رقم 3-6 استخدام الدالة ( )Write لإرسال التاريخ والوقت الحالي لخادم الويب وإظهارهما أمام المستخدم.

```
<html>
<head>
<title>Response Object</title>
</head>

<body>
The Sever time is :  <% Response.Write("<b>" & now() & "</b>") %>
</body>

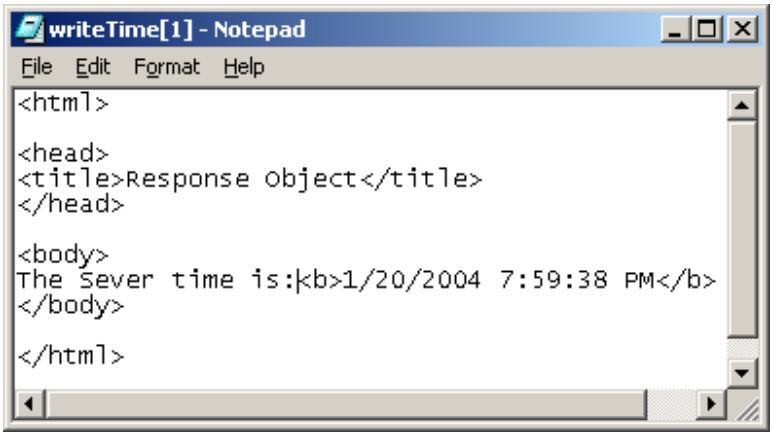
</html>
```



شكل 6-3 : مثال لاستخدام دالة( )Write لإرسال وقت وتاريخ خادم الويب

وهنا يخطر لنا سؤال: هل ستظهر للمستخدم شفرة ASP عند تصفحه لشفرة المصدر (Source Code) لملف ASP؟

كما ذكرنا سابقا أن شفرة ASP تنفذ على الخادم وتوضع محلها نتيجة المعالجة وبالتالي عندما يتصفح المستخدم شفرة مصدر الملف الحالي لا يجد شفرة ASP وتظهر له شفرة HTML وناتج تنفيذ شفرة ASP وكأن هذه الشفرة الظاهرة الآن مصدرها ملف HTML طبيعي، يوضح شكل رقم 4-6 إظهار شفرة المصدر للمثال السابق.



شكل 4-6 : شفرة المصدر والتي تحتوي على شفرة HTML فقط

عزيزي الدارس، لاحظ التالي:

- تم استخدام الدالة ( )Write (إحدى دوال مكتبة ASP) مع الدالة ( )now (إحدى دوال لغة VBScript) معاً في المثال السابق لإظهار تاريخ ووقت الخادم.
- تم لصق علامة   قبل وبعد استدعاء الدالة ( )now لإظهار الناتج بخط أسود عريض.

أسلوب آخر لإرسال معلومات لعميل الويب

يمكن استخدام المعامل "=" فقط لإرسال المعلومات إلى عميل الويب حيث يعمل المعامل "=" وظيفة الدالة ( )Write، توضح الصيغة العامة التالية كيفية إرسال نتيجة مصطلح (Expression) إلى عميل الويب.

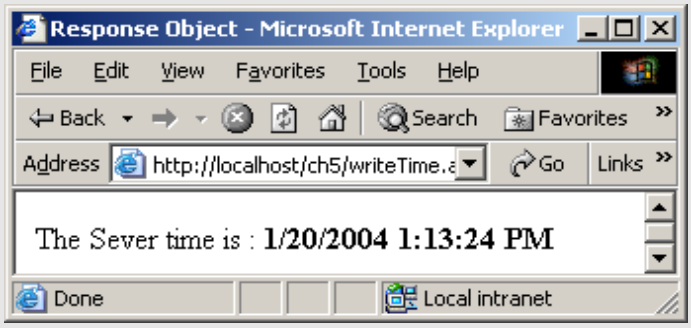
```
<%= Expression%>
```

فعلى سبيل المثال، يمكن إعادة كتابة المثال السابق مستخدما هذا الأسلوب كما هو موضح في الشكل رقم 5-6.

```
<html>
<head>
<title>Response Object</title>
</head>

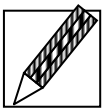
<body>
The Sever time is :  <% = "<b>" & now() & "</b>" %>
</body>

</html>
```



شكل 5- 6: استخدام المعامل = لإرسال بيانات إلى خادم الويب

تدريب (2)



ما فائدة الأوامر التالية؟

<% = "" & now() & "" %>

< % @Language= “JavaScript” %>

5. الدالة Redirect( )

تستخدم الدالة Redirect( ) لإعادة توجيه متصفح الإنترنت لزيارة عنوان صفحة ASP المستدعاة حالياً سواء أكان هذا العنوان لصفحة داخل نفس الموقع الحالي أو عنوان موقع آخر، وعند تنفيذ هذا الأمر، سيتم تجاهل بقية أوامر ASP التالية له.

الصيغة العامة

```
Response.Redirect (<URL>)
```

يمثل <URL> العنوان الجديد المراد الانتقال إليه، ويمكن كتابة هذا الموقع بشكل نسبي للفهرس الحالي أو بشكل مطلق والذي يبدأ بعنوان خادم الويب وينتهي باسم صفحة الويب.

مثال :

يوضح المثال التالي إعادة توجيه المستخدم إلى موقع البحث المناسب بناء على قيمة مستقبلة منه (SearchEngineValue)، فإذا كانت هذه القيمة تساوي "Yahoo" فيتم التوجيه إلى موقع Yahoo، وهكذا ، أما إذا لم تحتوِ هذه القيمة على اسم محرك فيتم توجيه المستخدم إلى الصفحة الرئيسية للموقع الحالي.

```
Select Case <searchEngineValue>
Case "Yahoo"
        Response.Redirect ("www.yahoo.com")
Case "Google"
        Response.Redirect ("www.google.com")
        .. ..
                .. ..
Case Else
        Response.Redirect ("index.htm")
End Select
```

عزيزي الدارس، لاحظ التالي:

1- أن جميع عناوين البحث مكتوبة بشكل مطلق.

2- أن عنوان الصفحة الرئيسية للموقع الحالي مكتوب بشكل نسبي (أي أن الصفحة الحالية والصفحة الرئيسية تتواجدان داخل نفس المجلد).

6. الدالة END( )

تتيح لك الدالة( )End أن توقف أوامر المعالجة في صفحة ASP وتمنعها من توليد أي مخرجات إضافية، تستخدم الدالة End في الغالب لإنهاء أوامر المعالجة عند وقوع الأخطاء أو الأحداث غير المتوقعة في الصفحة، فعلى سبيل المثال ربما ترغب في السماح للمستخدمين الذين تم تسجيلهم فقط بالوصول إلى موقعك، وفي هذه الحالة يمكنك أن تكتب أوامر التحقق من اسم المستخدم وكلمة المرور ثم تستخدم عبارة Response.End( ) لإنهاء الصفحة إذا لم يكن المستخدم مسجلاً لديك، مع العلم أن عبارة Response.End( ) تؤثر على صفحة ASP التي توجد بها فقط، فإذا كانت هناك صفحات ASP أخرى تتم معالجتها على الخادم، فإنها لن تتأثر بذلك.

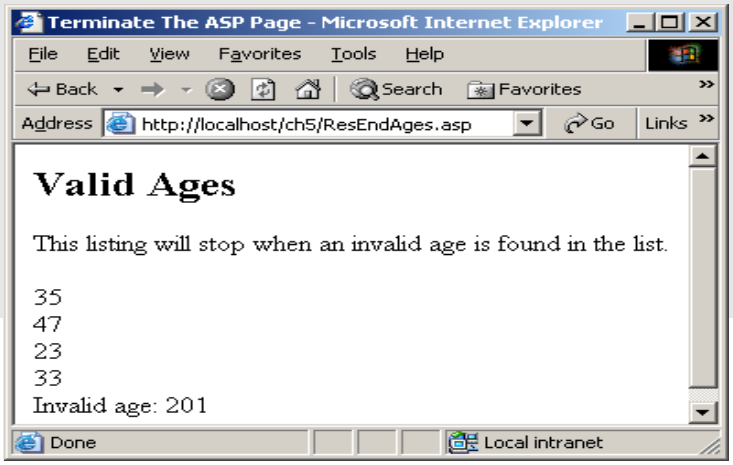
الصيغة العامة

```
Response.End( )
```

مثال :

يوضح المثال التالي (شكل رقم 6-6) استخدام الدالة End للتوقف عن التجول داخل عناصر مصفوفة ذات بعد واحد تحتوي على أعمار عدة أشخاص، وعندما تجد عمراً غير منطقي (أكبر من 120) تظهر رسالة خطأ تفيد ذلك.

```
<HTML>
<BODY>
<H2>Valid Ages</H2>
This listing will stop when an invalid age is found in the list.
<P>
<%
 , oneAg Dim  ages
ages = Array(35,47,23,33,201,52,30)
for each oneAg in ages
          If oneAg > 120 Then
              Response.Write("Invalid age: " & oneAge)
              Response.End
          Else
             Response.Write(oneAge & "<BR>")
          End If
next
%>
<P>
</body>
</html>
```



شكل 6-6 : استخدام الدالة End( ) للتوقف عن المعالجة عند وجود خطأ منطقي

الصفة Expires

تتيح الصفة Expires أن تحدد عدد الدقائق التي سيحتفظ فيها مستعرض الويب بمحتويات صفحة ASP في الذاكرة المؤقتة الخاصة به حيث يجب أن تختلف فترة انتهاء الصلاحية التي تحددها للصفحة حسب محتوياتها.

تستخدم معظم مستعرضات الويب الشهيرة ذاكرة مؤقتة (Cash Memory) لحفظ صفحات الويب التي يزورها المستخدم، وعندما يقوم المستخدم بعرض إحدى الصفحات مرة أخرى، فإن المستعرض سيبحث في الذاكرة المؤقتة لديه ليرى إن كانت الصفحة موجودة بها أم لا، فإذا كانت الصفحة موجودة، فإن المستعرض سيفحص الصفحة ليرى إن كانت فترة صلاحيتها ما زالت سارية أم لا، فإذا كانت فترة صلاحية الصفحة لم تنتهي بعد، فإن المستعرض سيعرض الصفحة الموجودة في الذاكرة المؤقتة بدلاً من قراءتها من مصدرها الأصلي مرة أخرى بغرض إسراع عملية العرض، أما إذا كانت فترة صلاحية الصفحة قد انتهت، فإن المستعرض سيعيد قراءة الصفحة من مصدرها الأصلي مرة أخرى. إن حفظ الصفحات التي زارها المستخدم في الذاكرة المؤقتة يوفر الوقت اللازم لقراءة هذه الصفحات من مصدرها الأصلي، ولكنه في نفس الوقت قد يحرم المستخدمين من رؤية أحدث إصدار من الصفحة.

يجب استخدام عبارة Response.Expires قبل أي أوامر داخل الصفحة ، فإذا تم كتابة أي أوامر قبل عبارة Response.Expires، يؤدي عميل الويب إلى عرض رسالة خطأ.

الصيغة العامة

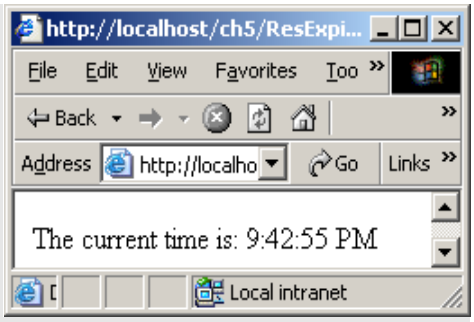
```
<% Response.Expires= <minutes> %>
```

- تحدد < minutes > عدد الدقائق الذي سيحتفظ بها متصفح الإنترنت بالصفحة في الذاكرة الخاصة به (Cash Memory) والتي بعدها يعيد تحميلها من خادم الويب.

مثال :

يوضح المثال التالي كيف يمكنك تحديد فترة انتهاء الصلاحية بصفر إذا كنت لا ترغب في تخزين صفحة ASP في الذاكرة المؤقتة للمستعرض. إن استخدام هذا الأمر بهذا الشكل مفيد بالنسبة لصفحات ASP التي تتغير محتوياتها باستمرار.

```
<% Response.Expires = 0 %><%
Response.Expires = 0 %>
<HTML>
<BODY>
The current time is: <%= Time()%>
</BODY>
</HTML>
```



شكل 6-7 : استخدام الصفة Expires لاسترجاع هذه الصفحة من الخادم باستمرار

الصفة IsClientConnected

تقوم بعض صفحات ASP بمهام تستغرق وقتاً طويلاً لمعالجتها مثل العمليات الحسابية المطولة أو استعادة البيانات من قواعد البيانات، وعند إنشاء صفحة ASP تقوم بمهمة تحتاج إلى وقت طويل، يجب استخدام عبارة Response.IsClientConnected لكي تجعل خادم الويب يتحقق من أن المستخدم مازال متصلاً به قبل أن ينهي خادم الويب المعالجة وعرض البيانات.

فعلى سبيل المثال، إذا كانت صفحة ASP تتطلب معلومات من ملفين لقاعدة بيانات، فإنك قد تطلب من الخادم قراءة المعلومات من الملف الأول، ثم تتحقق إن كان المستخدم مازال متصلاً بالخادم قبل أن تقوم بقراءة المعلومات من الملف الثاني حيث يمكن أن يساعد هذا على تجنب الوقت والموارد المهدرة دون داع.

```
<% Response.IsClientConnected  %>
```

عادة تستخدم هذه العبارة مع الجمل الشرطية للتحقق من استمرار اتصال المستخدم بصفحة الويب الحالية.

مثال :

يوضح المثال التالي (شكل رقم 8-6) استخدام الصفة IsClientConnected للتأكد من استمرارية اتصال المستخدم بالخادم

```
<HTML>
<BODY>

<H2>Welcome ...</H2>
Please wait while the page loads.
<%
If Response.IsClientConnected = true Then
          Response.Write("<Br> Thank you for your
patience.")
Else
          Response.End
End If
%>
<a href="index.htm"><Br>Click here</a>
 to transferee to the main page.
```



شكل 8-6 : استخدام الصفة IsClientConnected للتأكد من استمرارية اتصال المستخدم

عزيزي الدارس لاحظ التالي:

- لا يمكن تعديل هذه الصفة لأنها تقرأ فقط .
- لا تعمل عبارة Response.IsClientConnected مع بعض الإصدارات القديمة من برامج خادم الويب.

7. الكائن Request

يستخدم الكائن Request لاستقبال بيانات من المستخدم بغرض معالجتها على الخادم ثم إرسال نتيجة المعالجة إلى عميل الويب مستخدماً الكائن Response ، كما يمكن استقبال البيانات من المستخدم بغرض تخزينها في قاعدة بيانات وهكذا، يحتوي الكائن Request على مجموعات (Collections) والتي ستحتوي على البيانات المرسلة من عميل الويب، وتختلف هذه المجموعات بناء على الأسلوب المستخدم في إرسال البيانات، ومن أهم هذه المجموعات:

- المجموعة Form التي تحتوي على البيانات المرسلة من عميل الويب في نموذج يعتمد على الأسلوب Post (method= "post").
- مجموعة Querystring التي تحتوي على البيانات المرسلة من عميل الويب في نموذج يعتمد على الأسلوب Get (method= "Get")، أو تحتوي على البيانات المصاحبة لرابطة الاستدعاء (Hyperlink).

التعرف على المجموعات

يمكن النظر للمجموعة على أنها مصفوفة ذات بعد واحد تحتوي على مجموعة متغيرات (العناصر)، ويمكن التعامل مع هذه العناصر من خلال فهرس كل عنصر حيث يوجد لكل عنصر فهرس رقمي (مثل فهرس المصفوفة).

أما عنصر المجموعة يتكون من زوج مرتب والذي يتكون من اسم العنصر، وعادة قيم مسندة لهذا العنصر، ولإيضاح ذلك، يوضح شكل رقم 9-6 التركيب الداخلي للمجموعة حيث تتكون هذه المجموعة من خمس عناصر بداية من الفهرس صفر إلى الفهرس 4.

فهرس	اسم عنصر	قيم العنصر
0	(<Element Name1>,	Val1)
1	(<Element Name2>,	Val2, val3)
2	(<Element Name3>,	Val4)
3	(<Element Name4>,	Val5)
4	(<Element Name5>,	Val1)

شكل 9-6 :التركيب الداخلي للمجموعة: فهرس، واسم عنصر، وقيم العنصر

المجموعة Form

يوجد أسلوبان لاستقبال بيانات حقل إدخال معرف داخل نموذج: مستخدماً اسم الحقل أو مستخدماً فهرس الحقل داخل المجموعة Form ، يوضح الأمر التالي الصيغة العامة لاستقبال بيانات حقل إدخال.

```
Request.form ("<element>" [ OR <index>])
```

- يمثل المصطلح <element> اسم حقل الإدخال المعرف داخل النموذج والمراد استقبال بياناته.
- يمثل المصطلح <index> فهرس حقل الإدخال المعرف داخل النموذج والمراد استقبال بياناته.

عزيزي الدارس لاحظ

- أن مجموعة Form تستقبل فقط النموذج المرسل بطريقة post والمعرفة داخل الصفة method داخل علامة form.

- أنه يجب ربط شفرة ASP التي ستستقبل بيانات النموذج بكتابة اسم ملف ASP مع الصفة Action داخل علامة form.

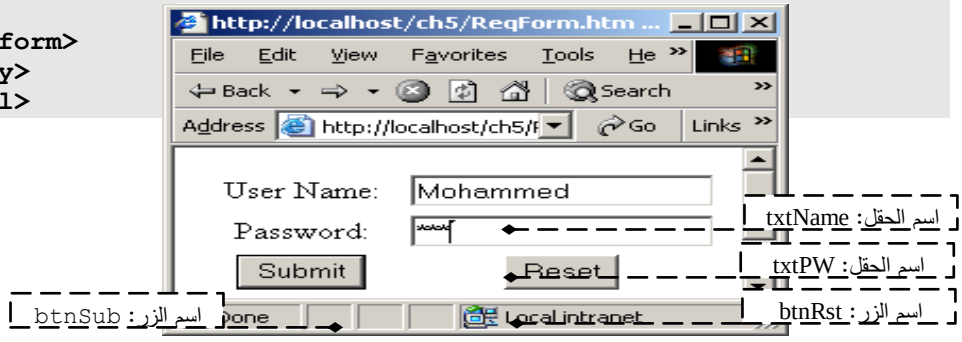
مثال: يوضح المثال التالي (شكلا رقم 10-6 ورقم 11-6) كيفية استقبال بيانات مستخدم (اسم المستخدم وكلمة المرور) من خلال نموذج، ثم إعادة طباعة هذه البيانات بعد إعادة تشكيلها، وفى النهاية عرض جميع عناصر النموذج من خلال التجول داخل مجموعة form وطباعة كل من اسم عنصر المجموعة والقيمة المصاحبة له.

يوضح شكل رقم 10-6 عرض النموذج داخل متصفح الإنترنت وبداخله البيانات المدخلة من قبل المستخدم، ويوضح شكل رقم 11-6 كل من شفرة ASP المسؤولة عن استقبال بيانات هذا النموذج وناتج تنفيذ هذه الشفرة والتي تظهر داخل متصفح الإنترنت.

ReqForm.htm

```
<html>
<body>
   <form method="POST" action="acceptDataForm.asp">
  User Name:
  <input type="text" name="txtName" size="20">
  Password:
  <input type="password" name="txtPW" size="20">
  <input type="submit" value="Submit" name="btnSub"></td>
  <input type="reset" value="Reset" name="btnRst"></td>

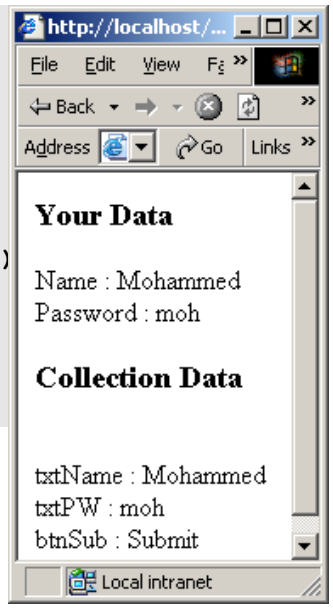
   </form>
</body>
</html>
```



شكل 10-6 : النموذج و شفرة HTML المناظرة له

acceptDataForm.asp

```
<%
Dim nm , pw , elem
nm = request.form("txtName")
pw = request.form("txtPW")
Response.write("<H3>Your Data</H3>")
Response.write( " Name :" & nm )
Response.write("<Br> Password : " & pw)
Response.write("<H3>Collection Data</H3>")
for each elem in request.form
    Response.write("<Br>" & elem & " : ")
    Response.write(request.form(elem))
next
%>
```



شكل 11-6 : شفرة Asp المسؤولة عن استقبال النموذج ونتيجة المعالجة

المجموعة QueryString

يتشابه استخدام المجموعة QueryString مع استخدام المجموعة Form ولكن تستخدم الثانية فقط عند استقبال بيانات نموذج مرسل بطريقة post (method = "post") وهى الطريقة الأكثر شيوعاً، أما مجموعة QueryString تستخدم في الحالات التالية:

- تستخدم لاستقبال بيانات نموذج مرسل بطريقة Get (method= "Get")، حيث ترسل البيانات مصاحبة لعنوان ملف ASP المرافق لصفة Action.
- تستخدم لاستقبال بيانات مصاحبة لرابطة استدعاء ملف من النوع ASP حيث تضاف علامة الاستفهام "؟" بعد اسم ملف ASP ثم مجموعة عناصر (على شكل <VarName>= <Value>) ويفصل بينهم بالمعامل "&".

```
<% Request.QueryString ( "<element>" [OR<index>])
```

- يمثل المصطلح <element> اسم أحد حقول الإدخال المعرف داخل النموذج أو اسم متغير داخل مجموعة المتغيرات المصاحبة لرابطة الاستدعاء.
- يمثل المصطلح <index> فهرس أحد حقول الإدخال المعرف داخل النموذج أو فهرس اسم متغير داخل مجموعة المتغيرات المصاحبة لرابطة الاستدعاء.

مثال :

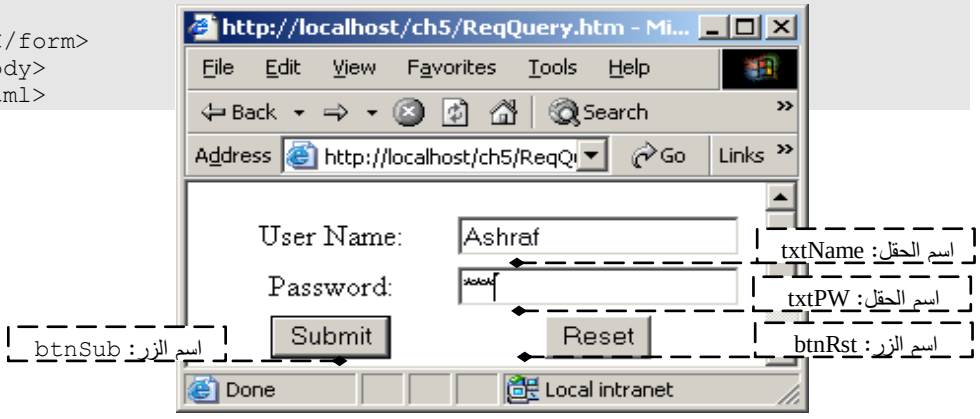
يوضح المثال التالي إعادة للمثال السابق مع الفارق أنه تم إرسال البيانات بطريقة "Get" بدلاً من طريقة "Post"، وتم استخدام مجموعة QueryString بدلاً من مجموعة Form. و لغرض التنوع، تم استخدام فهرس عنصر المجموعة بدلاً من اسم المتغير ولاحظ استخدام الصفة Count والتي تعيد حجم عناصر المجموعة الحالية.

يوضح شكل رقم 12-6 كلا من شفرة HTML الخاصة بالنموذج ولقطة من عرض النموذج داخل متصفح الإنترنت وبداخله البيانات المدخلة من قبل المستخدم. و يوضح شكل رقم136 كل من شفرة ASP المسؤولة عن استقبال بيانات هذا النموذج وناتج تنفيذ هذه الشفرة والتي تظهر داخل متصفح الإنترنت.

ReqQuery.htm

```
<html>
<body>
   <form method="Get" action="acceptDataQuery.asp">
  User Name:
  <input type="text" name="txtName" size="20">
  Password:
  <input type="password" name="txtPW" size="20">
  <input type="submit" value="Submit" name="btnSub"></td>
  <input type="reset" value="Reset" name="btnRst"></td>

   </form>
</body>
</html>
```



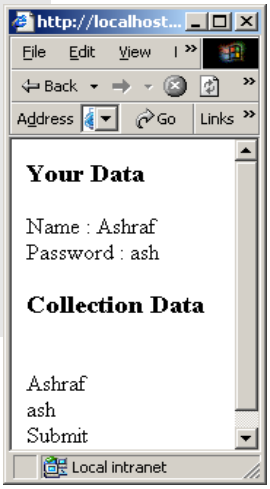
شكل 6-12 : النموذج و شفرة HTML المناظرة له

acceptDataQuery.asp

```
<%
Dim nm , pw , i

nm = request.QueryString("txtName")
pw = request.QueryString("txtPW")

Response.write("<H3>Your Data</H3>")
 & nm & "<Br> Password:" & pw) "Response.write("Name:
Response.write("<H3>Collection Data</H3>")
for i = 1 to request.QueryString.count
   Response.write("<br>" & request.QueryString(i))
next
%>
```



شكل 6-13 : شفرة Asp المسؤولة عن استقبال النموذج ونتيجة المعالجة

8. الكائن Session

من أهم فوائد تقنية ASP هو فكرة الكائن Session والذي يمكن المبرمج من تعريف مجموعة من المتغيرات على مستوى المستخدم (أي ينشئ لكل مستخدم نسخة من الكائن Session لكي يحتوي على معلومات خاصة بهذا المستخدم)، ومن هنا أتت فكرة تطبيق الإنترنت حيث أصبح لكل مستخدم جلسة عمل مختلفة عن المستخدمين الآخرين.

عندما يتحرك المستخدم بين الصفحات في موقعك، قد يطلب الموقع منه إدخال بعض المعلومات الشخصية مثل اسمه أو كلمة المرور الخاصة به، ومن الممكن أن تنشأ متغيرات جلسة (Session Variables) لحفظ المعلومات التي أدخلها المستخدم وبالتالي تجعلها متوفرة لكل الصفحات التي يستعرضها المستخدم، حيث يوفر هذا الأسلوب على المستخدم الحاجة إلى إدخال نفس المعلومات عند عرض كل صفحة في الموقع.

تبدأ الجلسة (Session) عندما يطلب المستخدم عرض صفحة ASP من موقع الويب الخاص بك، وتنتهي الجلسة عندما لا يقدم المستخدم طلباً بعرض صفحات أخرى لفترة معينة من الوقت، وفي الغالب تكون 20 دقيقة، حيث يتم بعدها التخلي عن الجلسة.

يمكن استخدام متغيرات الجلسة مع مستعرضات الويب التي تدعم ملفات الكعك (Cookies) فقط. إذا كان المستخدم يستعمل مستعرضاً قديماً، أو إذا كان قد أوقف دعم ملفات الكعك؛ فإن معلومات الجلسة ستضيع بمجرد أن يطلب المستخدم عرض صفحة أخرى.

إذا قامت إحدى صفحات ASP في موقع الويب الخاص بك بإنشاء متغيرات جلسة لأحد المستخدمين، فإن صفحات ASP الأخرى التي يعرضها المستخدم تستطيع أن تقرأ وتعالج القيم المحفوظة داخل هذه المتغيرات، ولن يتمكن المستخدمون الذين يزورون الموقع حاليا أن يتعاملوا مع معلومات الجلسة الخاصة بمستخدم آخر.

الصيغة العامة لإعطاء قيمة لمتغير جلسة

```
Session("<VarName>")= <Value>
```

توضح هذه الصيغة كيفية تخصيص قيمة لمتغير جلسة حيث يشير المصطلح <VarName> إلى اسم المتغير، ويشير المصطلح <Value> إلى القيمة المراد تخصيصها لهذا المتغير.

الصيغة العامة لاسترجاع قيمة من متغير جلسة

```
Session("<VarName>")
```

توضح هذه الصيغة كيفية إسناد قيمة من متغير جلسة بغرض طباعتها أو إسنادها لمتغير مؤقت.

مثال :

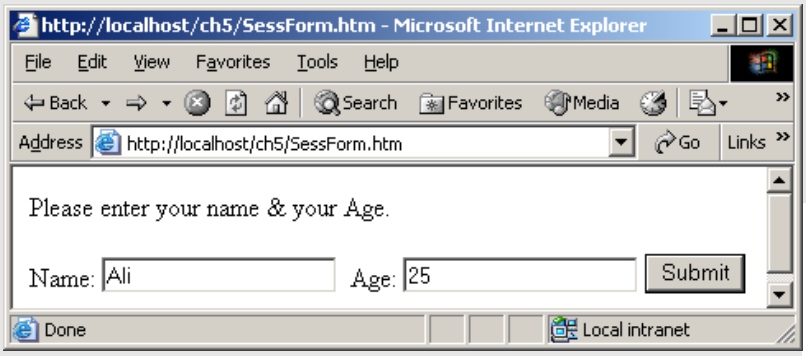
يوضح المثال التالي استقبال بيانات مستخدم (الاسم، والعمر) وتخزينها داخل متغيرات الجلسة الحالية لاستخدامهما داخل صفحات ASP الأخرى، وفيما يلي نوضح خطوات تنفيذ هذا السيناريو:

1. تصميم نموذج لاستقبال البيانات من المستخدم ثم إرسالها إلى خادم الويب.
2. تم استقبال هذه البيانات بواسطة الملف SessionVar.asp الذي خزن هذه البيانات داخل كل من متغير الجلسة userName ، ومتغير الجلسة userAge .
3. ثم إعادة توجيه عميل الويب لصفحة SessDisply.asp التي بدورها أعادت طباعة هذه البيانات مما يؤكد أن صفحات ASP المختلفة تستطيع أن تقرأ وتعدل متغيرات الجلسة.

يوضح شكل رقم 14-6 كل من شفرة HTML للنموذج وشكل النموذج عند استعراضه بواسطة متصفح الإنترنت. ويوضح شكل رقم 15-6 كلاً من شفرة ASP لملف SessionVar.asp، وملف SessDisply.asp وعرض نتيجة التنفيذ.

SessForm.htm

```
Please enter your name &amp; your Age.
<FORM ACTION="sessionvar.asp" METHOD="POST">
Name: <INPUT TYPE="text" NAME="txtName" size="20">
Age: <INPUT TYPE="text" NAME="txtAge" size="20">
<INPUT TYPE="submit" VALUE="Submit">
</FORM>
```



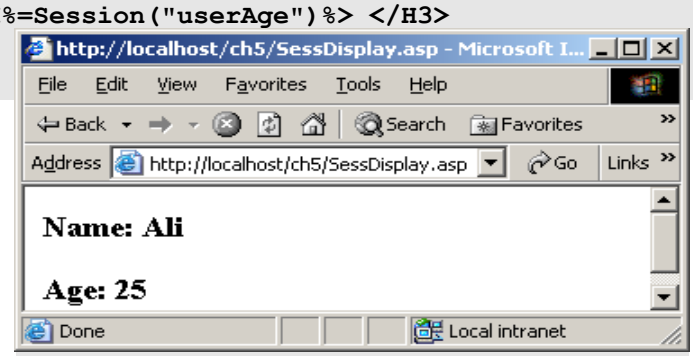
شكل 6-14 : النموذج و شفرة HTML المناظرة له

sessionvar.asp

```
<%
Session("userName")=Request.Form("txtName")
Session("userAge")=Request.Form("txtAge")
response.redirect("SessDisplay.asp")
%>
```

SessDisplay.asp

```
<H3>Name: <%=Session("userName")%> </H3>
<H3>Age:  <%=Session("userAge")%> </H3>
```



شكل 6-15 : شفرة Asp المسؤولة عن استقبال النموذج ونتيجة المعالجة

الصفة TimeOut

عزيزي الدارس، تتيح لك الصفة Timeout أن تحدد بالدقائق طول الفترة الزمنية التي يظل خادم الويب محتفظاً فيها بمعلومات الجلسة بعد آخر صفحة ASP يطلبها المستخدم من موقع الويب.

يتم الاحتفاظ بمعلومات الجلسة على خادم الويب في الوضع الافتراضي لمدة عشرين دقيقة، وتكون هذه المعلومات متاحة لكل صفحات ASP التي يعرضها المستخدم في موقع الويب، كما ستكون معلومات الجلسة متاحة لصفحات ASP حتى إذا قام المستخدم بزيارة موقع ويب آخر ثم عاد إلى موقعك في خلال مهلة الانتظار المحددة، أما إذا عاد المستخدم إلى موقعك بعد مهلة الانتظار المحددة ، فإن الجلسة ستنتهي وتلغى معلومات الجلسة من الذاكرة، ويمكنك أن تجعل موقع الويب الخاص بك أكثر أمناً باستخدام عبارة Session.Timeout عن طريق تغيير مدة الانتظار (مدة بقاء الجلسة في حال عدم تفاعل المستخدم).

الصيغة العامة

```
<% Session.Timeout [ = <value> ] %>
```

- تستخدم الصيغة السابقة لتخصيص القيمة <Value> للصفة Timeout أو لاسترجاع قيمة الصفة Timeout ، حيث إن الأقواس [ ] تعني أن ما بداخلها اختياري.

مثال . يوضح المثال التالي (شكل رقم 6-16) شفرة ASP وناتج تنفيذها، حيث يتم استرجاع قيمة الصفة Timeout وإعادة إخراجها بشكل منسق.

ASP Code

```
<%
Response.Write("The session timeout is currently ")
Response.Write(Session.Timeout & " minutes.")
%>
```

Result

The session timeout is currently 20 minutes.

شكل 6-16 : شفرة Asp المسؤولة عن استرجاع مهلة الانتظار و إعادة طباعتها

الدالة Abondon( )

تتيح لك الدالة Abondon( ) أن تقوم فوراً بإنهاء الجلسة لأحد المستخدمين وتمسح معلومات الجلسة الخاصة به (متغيرات الجلسة)، حيث يؤدي هذا إلى تحرير ذاكرة وموارد خادم الويب مما يزيد من كفاءة أدائه.

إن إنهاء الجلسة بهذا الشكل يفيد عندما يحدث خطأ أو عندما يقوم المستخدم بإجراء معين يبين أنه لم يعد يحتاج إلى معلومات الجلسة مثل تسجيل خروجه من موقع الويب. وإذا لم تقم بإنهاء الجلسة؛ فإن خادم الويب سيظل يحتفظ بالمعلومات الخاصة بها في ذاكرته حتى تنتهي مدة الانتظار.

وبشكل عام تدمر محتويات الجلسة بواسطة خادم الويب في حالتين فقط، عند مرور الفترة الزمنية المعرفة داخل الصفة Timeout ، وعند استدعاء الدالة Abondon ( ).

الصيغة العامة

```
<% Session.abondon( ) %>
```

تستخدم هذه الصيغة لإنهاء الجلسة الحالية للمستخدم الحالي.

مثال:

يوضح المثال التالي (شكل رقم 6-17) استخدام دالة Abondon( ) لإنهاء جلسة مستخدم داخل ملف SessAbondon.asp ، ثم تعريف متغير جلسة MyName

وإعطاء قيمة له واسترجاعها ثم إعادة توجيه عميل الويب إلى الملف testSess.asp الذي يحاول استرجاع هذا المتغير ولكن لا يجده ، مما يدل على التالي:

- يمكن التعامل مع متغيرات الجلسة بعد تنفيذ دالة ( )Abandon داخل نفس الملف.
- تدمر الجلسة تماماً بعد إنهاء جميع أوامر ASP الموجودة داخل الملف الذي اصدر دالة ( )Abandon.

Secabondon.asp

```
<%
Session.Abandon
Session("MyName") = "Mary"
Response.Write(Session("MyName"))
Response.redirect("testSes.asp")
%>
```

testSes.asp

```
<%
Response.Write(Session("MyName"))
%>
```

شكل 17-6 : شفرة Asp لاستخدام دالة Abondon لإنهاء الجلسة الحالية

9. الكائن Application

عزيزي الدارس، يستخدم الكائن Application لإمكانية تعريف متغيرات مشتركة على مستوى جميع المستخدمين، حيث يمكنك إنشاء متغيرات لحفظ معلومات التطبيق مثل عداد الزوار أو رسالة ترحيب، وتستطيع كل صفحات ASP في التطبيق أن تصل إلى المعلومات التي تم حفظها في متغيرات التطبيق. فعلى سبيل المثال إذا أنشأت متغير تطبيق لحفظ عداد الزوار؛ فإنك تستطيع أن تعرض عدد الزوار الذين استخدموا التطبيق في أسفل كل صفحة فيه.

يبدأ التطبيق عندما يقوم أول مستخدم بطلب صفحة ASP من التطبيق وينتهي عندما يتم إيقاف خادم الويب عن العمل أو عندما تتم إعادة تشغيله.

الصيغة العامة لإعطاء قيمة لمتغير تطبيق

```
Application("<VarName>") = <Value>
```

توضح هذه الصيغة كيفية تخصيص قيمة لمتغير تطبيق حيث يشير <VarName> إلى اسم المتغير، ويشير المصطلح <Value> إلى القيمة المراد تخصيصها لهذا المتغير.

الصيغة العامة لاسترجاع قيمة من متغير تطبيق

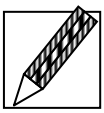
```
Application("<VarName>")
```

توضح هذه الصيغة كيفية استرجاع قيمة من متغير تطبيق وطباعتها أو إسنادها لمتغير مؤقت.

10. تأمين متغيرات التطبيق

عزيزي الدارس، إن جميع صفحات ASP لديها الحق أن تتعامل مع متغيرات التطبيق، وهذا يعني أن كل صفحة ASP في البرنامج تعرض لمستخدم ما قد تقوم بتعديل قيمة أحد متغيرات التطبيق، ومن ثم إذا حاولت عدة صفحات ASP تعديل نفس متغير التطبيق في نفس الوقت، فإن التعديلات التي أجرتها إحدى هذه الصفحات هي التي سيتم قبولها، وسوف يتم تجاهل تعديلات بقية الصفحات، ولحل هذه المشكلة نعرض في المواضيع التالية كلاً من الصفة Lock والصفة Unlock المسؤولتان عن تأمين متغيرات التطبيق أثناء تعديلهما.

تدريب (3)



مستخدماً كل من كائن Session، وكائن Application اكتب الإجراءات المطلوبة داخل ملف Global.asa لإنشاء متغير عام CurrentUsers يحتوي على العدد الحالي للزوار، ومتغير آخر يحتوي على العدد الكلي للزوار TotUsers.

الصفة Lock

عزيزي الدارس، تتيح لك الصفة Lock أن تقوم بتأمين متغيرات التطبيق أثناء تعديلها بواسطة إحدى صفحات ASP، وفي أثناء تأمين المتغيرات، إذا حاولت إحدى صفحات ASP الوصول إلى متغير تطبيق تم تأمينه؛ فإن خادم الويب سيوقف معالجة هذه الصفحة بصورة مؤقتة حتى يتم إلغاء تأمين المتغيرات.

الصفة Unlock

كما تتيح لك الصفة Unlock أن تقوم بإلغاء تأمين المتغيرات، وإذا لم تستخدم هذه العبارة؛ فإن تأمين المتغيرات سيلغى تلقائياً عندما تنتهي المعالجة.

يجب وضع عبارتي Application.Lock، Application.Unlock قبل وبعد الأوامر التي تقوم بتعديل قيم متغيرات التطبيق مباشرة على التوالي.

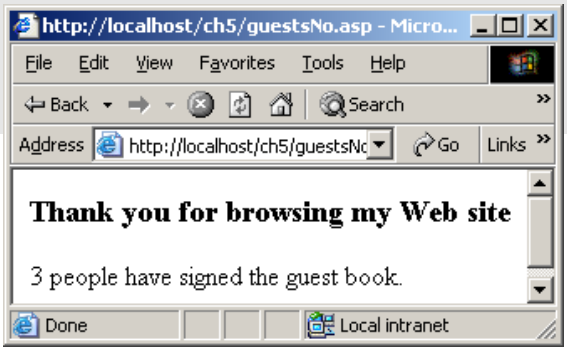
مثال:

يوضح المثال التالي (شكل رقم 18-6) استخدام كل من الصفة Lock و الصفة Unlock لتأمين تعديل متغير عدد زوار الموقع ، ثم استخدام هذا المتغير لطباعة الرسالة المناسبة.

```
<HTML>
<BODY>
<H3>Thank you for browsing my Web site</H3>
<%
Application.Lock
Application("guests")=Application("guests") + 1
Application.unLock

If Application("guests") = 1 Then
          strPeople=" person has "
Else
          strPeople=" people have "
End If
Response.Write(Application("guests") & strPeople)
Response.Write("signed the guest book.")
%>

<BODY>
</HTML>
```



شكل 18-6 : استخدام الصفة Lock و الصفة Unlock لتأمين تعديل متغير عدد زوار الموقع

11. إزالة معلومات الجلسة والتطبيق

عزيزي الدارس، تقوم متغيرات الجلسة و التطبيق بحفظ المعلومات داخل المتغيرات أثناء استخدامها ومن الممكن أن تزيل أحد أو كل هذه المتغيرات، ولكن تذكر إن إزالة متغيرات الجلسة أو التطبيق لا تعني إعادة تعيين قيمتها إلى صفر مثلا، وإنما ستقوم بمسح المتغير نفسه من الذاكرة، ويظل هكذا إلى أن يتم تعريفه مرة أخرى.

الصيغة العامة لإزالة متغير

<ObjName>.Contents.Remove("<VarName>")

- يمثل المصطلح <ObjName> الكائن Session أو الكائن Application.
- يمثل المصطلح <VarName> اسم المتغير المراد إلغاؤه.

مثال:

يستخدم المثال التالي لإلغاء متغير الجلسة UserName من الذاكرة.

```
Session.contents.Remove (“UserName”)
```

مثال:

يستخدم المثال التالي لإلغاء متغير التطبيق guests من الذاكرة.

```
Application.contents.Remove("guests")
```

12. التعامل مع ملف Global .a sa

عزيزي الدارس، يستخدم ملف Global.asa لكي يحتوي على الإجراءات الفرعية التي تتم معالجتها بواسطة خادم الويب عند بداية ونهاية جلسة مستخدم (Session) أو عند بداية ونهاية التطبيق (Application)، حيث يحتوي كل تطبيق على ملف Global.asa واحد فقط، والذي يجب أن يسمى بهذا الاسم، وأن يتم حفظه على المجلد الرئيسي للتطبيق، وتشير الحروف "asa" الموجودة داخل امتداد الملف إلى اختصار العبارة Active Server Application.

يجب الالتزام بتسمية الإجراءات الأربعة المذكورة عاليه بأسماء معينة لأن الخادم سيستدعيها بشكل تلقائي عند بداية ونهاية الجلسة وعند بداية ونهاية التطبيق حيث سيتم استدعاء الإجراء ( )Application_OnStart عند بداية التطبيق، واستدعاء الإجراء( ) Application_OnEnd عند نهاية التطبيق، واستدعاء الإجراء ( )Session_OnStart عند بداية الجلسة، واستدعاء الإجراء (Session_ OnEnd) عند نهاية الجلسة، كما يجب وضع هذه الإجراءات بين علامتي <Script> ، و </Script>، وإسناد القيمة “sever” إلى السمة RUNAT داخل علامة <Script>

والتي تشير إلى أن هذه الإجراءات سيتم معالجتها على خادم الويب وليس العميل، وتوضح النقاط التالية سيناريو استدعاء هذه الإجراءات.

- يقوم خادم الويب بتنفيذ الأوامر الموجودة داخل إجراء ( )Application_OnStart عندما يطلب أول مستخدم للموقع صفحة من الموقع، وفي الغالب يستخدم هذا الإجراء الفرعي لإنشاء متغيرات التطبيق التي سيتم استخدامها طوال عمل التطبيق.
- يقوم خادم الويب بتنفيذ الأوامر الموجودة داخل الإجراء الفرعي ( )Application_OnEnd عندما ينتهي التطبيق، وهو ما يحدث في الغالب عندما يتم إيقاف تشغيل خادم الويب، وفي الغالب يستخدم هذا الإجراء الفرعي لتنفيذ الأوامر الخاصة بإعادة ترتيب الأمور داخل التطبيق قبل إنهائه، مثل إزالة متغيرات التطبيق من الذاكرة، وكتابة المعلومات إلى ملفات السجل، وإغلاق الملفات المفتوحة.
- تبدأ الجلسة Session عندما يطلب مستخدم جديد عرض صفحة من التطبيق، وعندما تبدأ الجلسة، فإن خادم الويب سيقوم تلقائياً بالبحث في الملف Global.asa عن إجراء فرعي باسم ( )Session_OnStart، ويقوم بتنفيذ الأوامر الموجودة به قبل معالجة الصفحة التي طلبها المستخدم، وفى الغالب يستخدم هذا الإجراء لفتح اتصال جديد بقاعدة البيانات أو التأكد من أن المستخدم قد سجل دخوله إلى الموقع بصورة صحيحة.

وعند انتهاء الجلسة، سيقوم خادم الويب بتنفيذ الأوامر الموجودة بالإجراء
( )Session_OnEnd تنتهي الجلسة عندما تنتهي مهلة الانتظار الخاصة بها أو عندما يتم التخلي عنها بتنفيذ الدالة ( )Abondon.

مثال :

يستخدم المثال التالي (شكل رقم 19-6) لتعريف متغير يحتوي على عدد الزوار الكلي لزوار الموقع (countUsers)، ومتغير يخزن عدد الزوار الحاليين(currentUsers)، وفيما يلي سيناريو تنفيذ هذا المثال.

- عند بداية التطبيق، تسند قيمة ابتدائية لكل من المتغيرين بقيمة صفر .
- عند بداية جلسة جديدة، تزاد قيمة كل من المتغيرين بقيمة واحد.

- عند إنهاء جلسة، تنقص قيمة المتغير currentUsers بقيمة واحد.

شكل 19-6 : استخدام الملف Global.asa لتعريف و تعديل عدد الزوار الكلي والحالي للموقع

Global.asa

```
<SCRIPT LANGUAGE="VBScript" RUNAT="Server">
Sub Application_OnStart
    Application("countUsers")=0
    Application("currentUsers")=0
End Sub



Sub Session_OnStart
 Application.lock

Application("countUsers")=Application("countUsers")
+1

Application("currentUsers")=Application("currentUsers
") +1
 Application.unlock
End Sub

Sub Session_OnEnd
   Application.lock

Application("currentUsers")=Application("currentUsers
")-1
   Application.unlock
End Sub

</SCRIPT>
```

أسئلة تقويم ذاتي



1. ضع علامة √ أمام العبارة الصحيحة وعلامة X أمام العبارة الخاطئة؟
   - تنتمي الدالة Redirect إلى الكائن Request.
   - يستخدم الأسلوب Get لإرسال النموذج في مجموعة Form .
   - يستخدم الكائن Application لتعريف متغيرات على مستوى جميع المستخدمين.
   - تستخدم الدالة Remove مع الكائن Session لإنهاء الجلسة بين المستخدم وخادم الويب.
   - يجوز تخزين الملف Global.asa داخل فهرس فرعي داخل الفهرس الرئيسي لموقع الويب.
   - يجوز كتابة كل من شفرة HTML وشفرة ASP داخل ملف امتداده asp.*.
2. تستخدم المجموعة querystring للبحث في جدول في قاعدة البيانات.

الخلاصة

في هذه الوحدة عزيزي الدارس، قمنا بدراسة المعالجة لدى الخادم (Server-Side Programming) والتي تمكن المبرمج من تطوير تطبيقات التجارة إلكترونية (E-commerce Applications) وغيرها من تطبيقات الإنترنت الهامة، وتعرفنا أنه تتم المعالجة لدى الخادم بواسطة تقنية ASP (اختصار لعبارة Active Server Pages) والتي تم تطويرها من قبل شركة Microsoft ، تقدم هذه التقنية مكتبة من الكائنات (Objects) والتي تمكن المبرمج من تطوير تطبيقات الإنترنت.

يتكون تطبيق الإنترنت (Web-Based Applications) من مجموعة من البرامج الفرعية والتي تخزن في ملفات من النوع ASP حيث يحتوي كل ملف على برنامج فرعي يتم استدعاؤه عند الحاجة إليه، وعند استدعاء هذه الملفات بواسطة عميل الويب (متصفح الإنترنت)، ستتم معالجة هذه الملفات أولاً لدى خادم الويب بواسطة برنامج محرك تنفيذي للبرامج النصية (Scripting Engine) يوجد داخل خادم الويب

لمحة مسبقة عن الوحدة التالية

عزيزي الدارس، قبل أن تتمكن صفحة ASP من الوصول إلى قواعد البيانات يجب إنشاء اتصال بقاعدة البيانات، حيث يجب إنشاء "اسم مصدر بيانات" Data Source Name (تختصر DSN) لقاعدة البيانات المخزن بها بيانات الموقع ، ثم نستخدم DSN الذي أنشأناه داخل صفحات ASP والذي يحتوي على كل من نوع قاعدة البيانات التي تريد أن تتصل بها وعلى مكان قاعدة البيانات وعلى كل من اسم المستخدم وكلمة المرور المطلوبين للتعامل مع قاعدة البيانات.وهذا ماسنعرفه في الوحدة التالية إن شاء الله.

إجابات التدريبات

```
<%
   Dim   rad,  high,   volume
    rad=Request.Form("txtrad")
    high =Request.Form("txtHigh")
    volume = 3.14* rad^2 *  high
   Response.Write(“The volume is: ” & volume)
%>
```

< % @Language= “JavaScript” %>

يستخدم هذا الأمر لإخبار خادم الويب أن اللغة التي ستستخدم في كتابة شفرة ASP هي لغة JavaScript.

<% = "" & now() & "" %>

الإجابة:

يستخدم هذا الأمر لاسترجاع التاريخ والوقت الحالي لخادم الويب وإظهارهما أمام المستخدم.

```
<SCRIPT LANGUAGE="VBScript" RUNAT="Server">
        Sub Application_OnStart
                 Application(" TotUsers")=0
                Application("currentUsers")=0
        End Sub
        Sub Session_OnStart
              Application.lock
                Application(" TotUsers")=Application(" TotUsers") + 1

Application("currentUsers")=Application("currentUsers") + 1
             Application.unlock
        End Sub

        Sub Session_OnEnd
        Application.lock
          Application("currentUsers")=Application("currentUsers") -
1
             Application.unlock
        End Sub
</SCRIPT>
```

المراجع

[1] عبدالكريم العقيل، "المخطط الأزرق لبناء صفحات ويب ديناميكية"، مكتبة جرير، الرياض، 2001م.

[2] د. أحمد أحمد شعبان دسوقى، أ. د. السيد محمود الربيعي، "تقنيات وبرمجة الإنترنت" مكتبة الرشد، الرياض، 2004م.

[3] د. عبد القادر عبد الله الفنتوخ، "الإنترنت للمستخدم العربي"، مكتبة العبيكان، الرياض، 2001م.

[4] د. إبراهيم فتحي معوض، "أساسيات برمجة الإنترنت"، معهد العالمية للحاسب والتقنية، 2005م.


الوحدة السابعة
التعامل مع قواعد البيانات

محتويات الوحدة

الموضوع	الصفحة
المقدمة	217
تمهيد	217
الأهداف	218
1. إنشاء اسم مصدر البيانات (Data Source Name)	219
2. ما هي ADO	223
3.إعداد الاتصال مستخدما الكائن Connection	224
4. معالجة البيانات مستخدما الكائن Record set	228
1.4 الدالة ( )Open	229
2.4 التركيب الداخلي لكائن Recordset	230
5. استرجاع وعرض البيانات	232
6. البحث عن معلومات	234
7. تعديل البيانات	235
8. حذف السجلات	237
9. إضافة سجل جديد	241
الخلاصة	245
إجابات التدريبات	246
مسرد المصطلحات	247
المراجع	248

المقدمة

تمهيد

عزيزي الدارس، من أهم خصائص تقنية ASP هي قدرتها على الاتصال بقواعد البيانات التي تقوم بحفظ وإدارة مجموعات كبيرة من البيانات في شكل جداول، حيث نستطيع تطوير موقع متغير المحتوى يتكون من مجموعة صفحات ASP لجعل هذه المعلومات متاحة للمستخدمين الذين يزورون الموقع ومجموعة صفحات أخرى مسؤولة عن تعديل البيانات داخل قاعدة البيانات.

وبدلاً من حفظ المعلومات في ملفات نصية أو في صفحات ويب ساكنة، نستطيع إعداد صفحات ASP لقراءة هذه المعلومات وتنسيقها، وعندما يزور المستخدم هذه الصفحات؛ فإن المعلومات التي ستعرض داخل هذه الصفحات سيتم إنشاؤها من أحدث المعلومات المتوفرة في قاعدة البيانات.

سيتم إيضاح كيفية تطوير تطبيق ويب يتعامل مع قاعدة بيانات بشرح مثال متكامل والذي سيشار إليه من بداية الوحدة إلى نهايتها، حيث سنوضح من خلاله العمليات الأساسية المطلوبة في مثل تلك التطبيقات: البحث، الإضافة، الحذف، والتعديل، حيث سيتم الاستعانة بقاعدة بيانات خاصة بمكتبة (C:\WebDev\Library.mdb) والتي تحتوي على الجدول "Book" و الجدول "Category". تم تطوير هذه القاعدة بالبرنامج MS Access

أهداف الوحدة



عزيزي الدارس، بنهاية دراسة هذه الوحدة ينبغي أن تكون قادرا على أن:

1. تُتمكن من إنشاء اسم مصدر البيانات.
2. تعرف ما هي ADO.
3. تتعامل مع إعداد الاتصال مستخدما الكائن Connection.
4. تتعرف على كيفية معالجة البيانات مستخدما الكائن Recordset .
5. تستطيع استرجاع وعرض البيانات .
6. تتمكن من البحث عن معلومات .
7. تدرك كيفية البحث عن معلومات .
8. تعرف كيفية حذف وإضافة السجلات.

1. إنشاء اسم مصدر البيانات

عزيزي الدارس، بعد أن تقوم بإنشاء قاعدة البيانات، يجب أن تعين لها اسم مصدر بيانات Data Source Name (تختصر DSN) الذي يقوم بحفظ المعلومات التي تخبر صفحات ASP بكيفية الوصول إلى قاعدة البيانات.

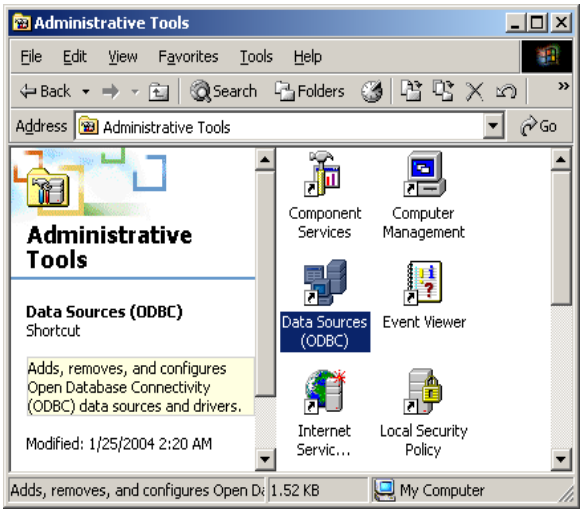
سيتم إنشاء اسم مصدر البيانات مرة واحدة فقط لقاعدة البيانات، حيث لا نحتاج إلى إنشاء اسم جديد في كل مرة نقوم فيها بتغيير أو تحديث هيكل قاعدة البيانات، ويجب أن يتم إنشاء اسم مصدر البيانات على خادم الويب الذي يستضيف قاعدة البيانات وصفحات ASP التي ستتصل بهذه القاعدة.

مثال:

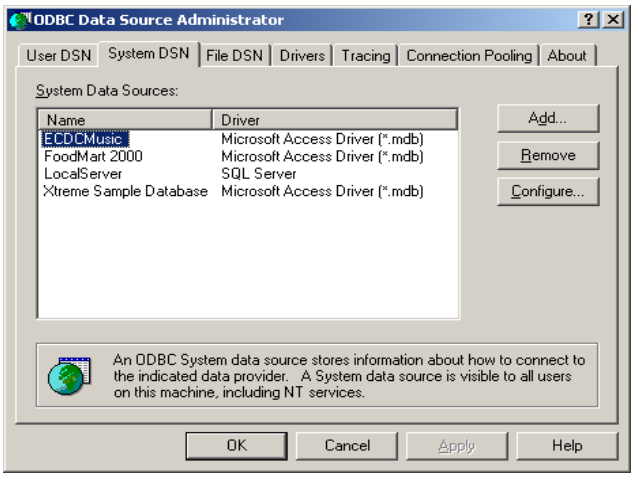
توضح الخطوات التالية تفاصيل كيفية إنشاء اسم مصدر بيانات لقاعدة البيانات library.mdb المذكورة عاليا، مع العلم أن بيئة التشغيل للجهاز المستضيف لقاعدة البيانات هو Windows 2000.



1) في لوحة التحكم انقر نقراً مزدوجاً فوق رمز Administrative Tool



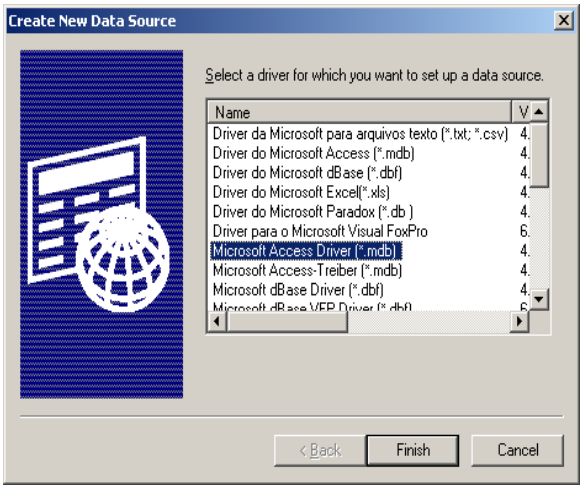
2) انقر نقراً مزدوجاً فوق رمز Data Sources.



3) سيظهر مربع حوار ODBC Data Source Administrative.

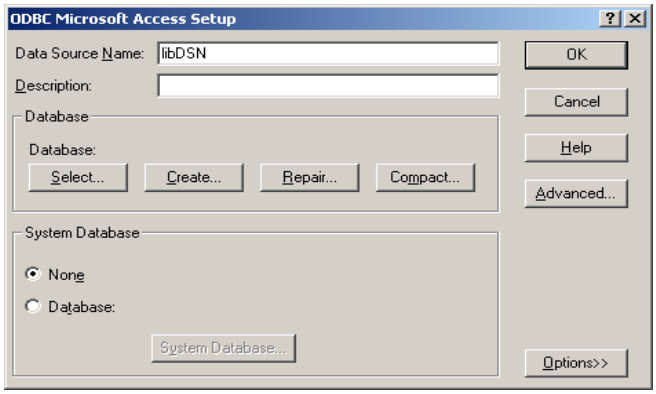
4) انقر علامة التبويب System DSN.

5) انقر Add لإنشاء اسم مصدر البيانات.



6) انقر Microsoft Access Driver.

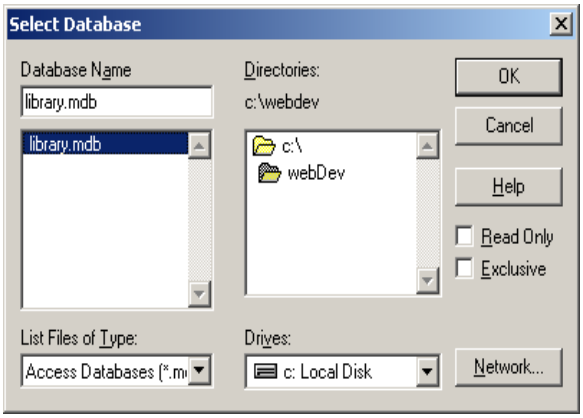
7) انقر زر Finish.



8) سيظهر مربع حوار ODBC Microsoft Access setup.

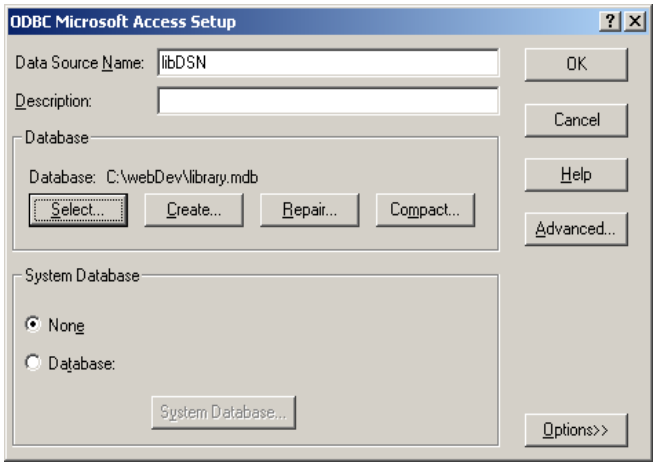
9) اكتب اسم مصدر البيانات الذي تريد استخدامه لقاعدة البيانات (libDSN مثلا).

10) انقر زر Select لعرض مربع حوار Select Database.

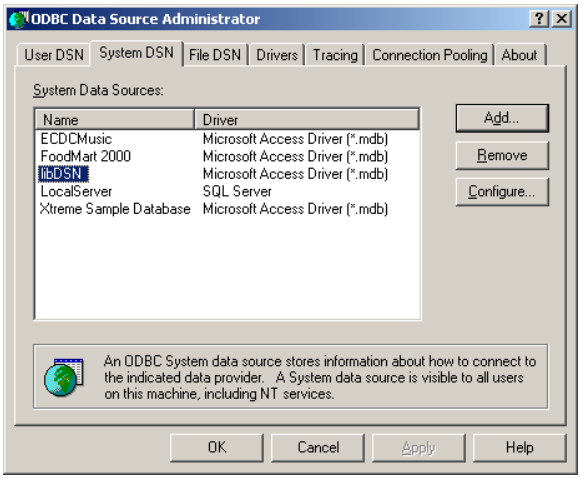


11) اختر قاعدة البيانات التي تريد إنشاء اسم مصدر بيانات لها (library.mdb).

12) انقر زر OK في مربع حوار Select Database.



13) انقر زر OK في مربع حوار ODBC Microsoft Access Setup



14) سيظهر اسم مصدر البيانات الجديد.

15) انقر زر OK لإغلاق مربع حوار ODBC Data Source Administrator.

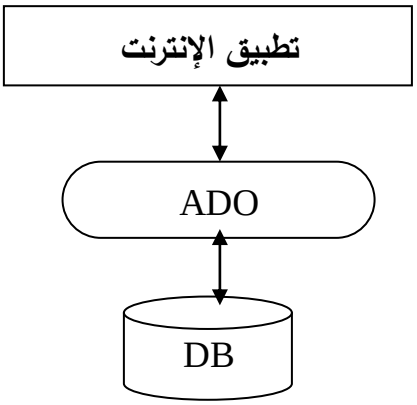
سؤال تقويم ذاتي

؟	عرف المصطلح DSN. وما فائدة استخدامه؟

2. ما هي ADO

عزيزي الدارس، ADOهي اختصار للعبارة ActiveX Data Object، وهي إحدى التقنيات الهامة المقدمة من شركة Microsoft لتسهيل عملية إدارة وصيانة البيانات داخل قاعدة البيانات. تتكون ADO من مجموعة من الكائنات (Objects) التي تقدم لمطوري التطبيقات بصورة عامة وتطبيقات الإنترنت بصورة خاصة أسلوباً سهلاً وكفء للاستعلام، وللإضافة، وللتعديل، ولحذف بيانات في قاعدة بيانات.

من أهم الكائنات المقدمة داخل مكتبة ADO هو الكائن Connection، والذي يستخدم لإنشاء عملية الاتصال بقاعدة البيانات، والكائن Recordset الذي يستخدم لاستقبال وتخزين مجموعة سجلات ولإجراء عمليات الاستعلام، والإضافة،



شكل 7-1 : علاقة الربط بين مكون ADO، وتطبيق الإنترنت، وقاعدة البيانات

و يوضح شكل رقم 7-1 علاقة الربط بين مكون ADO وتطبيق الإنترنت و قاعدة البيانات.

3. إعداد الاتصال مستخدما الكائن Connection

عزيزي الدارس، بعد إنشاء اسم مصدر البيانات، يمكنك إعداد الاتصال بين صفحات ASP وقاعدة البيانات التي تريد أن تحتوي على بيانات الموقع الخاص بك، ولكي تقوم بإعداد الاتصال بقاعدة البيانات، يجب تعريف نسخة (متغير) من كائن Connection وإعطاء هذه النسخة اسماً، وبعد ذلك، ستتمكن من استخدام دوال و صفات الكائن Connection للاتصال بقاعدة البيانات.

الصيغة العامة لتعريف نسخة من Connection

```
Set <ConInstance> = Server.CreateObject ("ADODB.Connection")
```

تستخدم هذه الصيغة لإنشاء نسخة من الكائن Connection المعرف داخل مكتبة ADO حيث:

- يمثل الأمر Set أحد أوامر لغة VBScript الذى يستخدم لإسناد كائن إلى متغير.
- يمثل <ConInstance> اسم النسخة المراد إنشاؤها من الكائن Connection.

- يمثل Server أحد كائنات مكتبة ASP.
- تمثل الدالة ( )CreateObject أحد الدوال المعرفة داخل الكائن Server، وتستخدم لإنشاء كائن وإرجاعه بناءً على نوع المعامل التي تستقبله (في هذه الحالة المعامل هو "ADODB.Connection").
- تستخدم العبارة "ADODB.Connection" لإخبار الدالة ( )CreateObject إنشاء نسخة من الكائن Connection المعرف داخل مكتبة ADO.

مثال:

يوضح المثال التالي كيفية إنشاء النسخة Con من الكائن Connection لتطبيق المكتبة.

```
Set Con = Server.CreateObject("ADODB.Connection")
```

يحتوي الكائن Connection على العديد من الصفات والدوال الهامة والتي تمكن المبرمج من إجراء عملية الاتصال بقاعدة البيانات، وفيما يلي سيتم شرح أهم هذه الصفات والدوال.

الصفة ConnectionTimeout

عزيزي الدارس، يجب استخدام الصفة ConnectionTimeout لتحديد مهلة الانتظار لإتمام عملية الاتصال، حيث تحدد هذه المهلة الفترة الزمنية (بالثواني) التي سينتظرها خادم الويب عند محاولة فتح الاتصال قبل أن يصدر رسالة خطأ.

الصيغة العامة

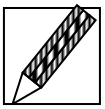
```
<ConInsturce>.ConnectionTimeout = <seconds>
```

- يمثل <ConInsturce> نسخة معرفة من الكائن Connection.
- تمثل <Seconds> الفترة الزمنية (بالثواني) التي سينتظرها خادم الويب لإتمام عملية الاتصال قبل أن يصدر رسالة خطأ.

مثال: يوضح المثال التالي كيفية إسناد 60 ثانية لصفة ConnectionTime للمتغير Con الخاص بتطبيق المكتبة.

```
Con.ConnectionTimeout = 60
```

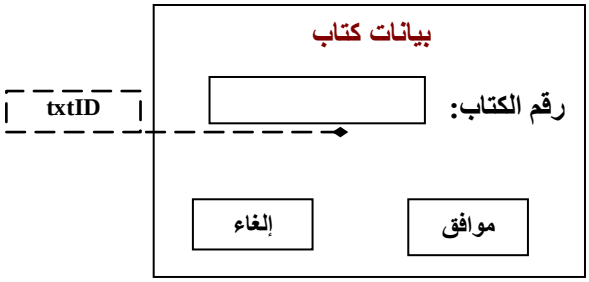
تمرين (1)



1. توجد لديك قاعدة بيانات مكتبة ، وتحتوي هذه القاعدة على عدة جداول منها جدول Book والذي يستخدم لتخزين بيانات الكتب .

يحتوى جدول Book على الحقول التالية

- ID يمثل رقم الكتاب.
- Name يمثل اسم الكتاب
- Price يمثل سعر الكتاب.



افترض أنك أنشأت اسم مصدر بيانات مستخدماً برنامج ODBC وأعطيت له اسم libDSN. مستخدما النموذج الموضح ، اكتب شفرة ASP المسؤولة عن استقبال رقم الكتاب والبحث عنه ، وإظهار بياناته إن وجد أو رسالة عدم توافره إن لم يتم إيجاده.

الدالة Open( )

بعد تعريف اسم مصدر البيانات وتعريف نسخة من الكائن Connection و تخصيص مهلة الانتظار لإتمام عملية الاتصال له، يجب تنفيذ عملية الاتصال وذلك عن طريق الدالة Open( ) حيث تستخدم الدالة Open( ) اسم مصدر البيانات لقاعدة البيانات التي تريد إجراء الاتصال بها.

الصيغة العامة

```
<ConInsturce >.Open  " DSN  =  <DataSourceName>"
```

- يمثل <DataSourceName> اسم مصدر البيانات الذي تم إنشاؤه.
- يمثل <ConInsturce> نسخة معرفة من الكائن Connection.

مثال:

يوضح المثال التالي كيفية استخدام الدالة Open( ) لتنفيذ عملية الاتصال بقاعدة البيانات Library.mdb، حيث مرر إليها اسم مصدر البيانات libDSN المعرف سابقا.

```
Con.Open "DSN = libDSN"
```

الدالة Close( )

عندما تنتهي معالجة البيانات داخل صفحة ASP والتي سبقها فتح اتصال مع قاعدة البيانات، فإنه يجب إغلاق الاتصال مستخدما الدالة Close، ثم يجب بعدها أن تسند الكلمة Nothing للنسخة المعرفة من الكائن Connection مستخدما الأمر Set وذلك لتحرير ذاكرة الخادم.

الصيغة العامة

```
<ConInsturce>.Close()
set  <ConInsturce>  =  Nothing
```

- يمثل <ConInsturce> نسخة معرفة من الكائن Connection.

مثال:

عزيزي الدارس، يوضح المثال التالي الخطوات السابقة مجتمعة لتعريف نسخة من الكائن Connection وتخصيص مهلة الانتظار لإتمام عملية لاتصال له وتنفيذ عملية الاتصال ومعالجة البيانات (هذا الجزء فارغ الآن) و كيفية استخدام الدالة Close لتنفيذ عملية إغلاق الاتصال مع قاعدة البيانات Library.mdb، وأخيرا تحرير ذاكرة الخادم التي تم تخصيصها للنسخة Con.

```
Set  Con = Server.CreateObject("ADODB.Connection")
Con.ConnectionTimeout  =  60
Con.Open  "DSN = libDSN"
        'معالجة البيانات
Con.close()
Set  Con= Nothing
```

عزيزي الدارس لاحظ :

يمكنك الوصول إلى أكثر من قاعدة بيانات واحدة في أوامر ASP عن طريق إنشاء عدة نسخ من الكائن Connection.

4. معالجة البيانات مستخدما الكائن Recordset

بعد تأسيس الاتصال بقاعدة البيانات، يمكن استخدام الكائن Recordset لقراءة المعلومات من قاعدة البيانات، حيث يستخدم الكائن Recordset عند العمل مع عدد كبير من السجلات التي يتم تحديثها باستمرار، ولاستخدام الكائن Recordset يجب إنشاء نسخة من الكائن وحفظها داخل متغير.

الصيغة العامة لتعريف نسخة من الكائن Recordset

```
Set  <RS>  = Server.CreateObject ("ADODB.Recordset")
```

تستخدم هذه الصيغة لإنشاء نسخة من الكائن Recordset المعرف داخل مكتبة ADO حيث:

- يمثل الأمر Set أحد أوامر لغة VBScript ويستخدم لإسناد كائن إلى متغير.
- يمثل المصطلح <RS> اسم النسخة المراد إنشاؤها من الكائن Recordset.
- يمثل المصطلح Server أحد كائنات مكتبة ASP.
- تمثل الدالة CreateOb0ject أحد الدوال المعرفة داخل الكائن Server وتستخدم لإنشاء كائن وإرجاعه بناء على نوع المعامل التي تستقبله (في هذه الحالة المعامل هو "ADODB. Recordset").
- تستخدم العبارة "ADODB.Recordset" لإخبار الدالة CreateObject إنشاء نسخة من الكائن Recordset المعرف داخل مكتبة ADO.

مثال:

لقراءة بيانات جدول Book المخزن داخل قاعدة بيانات library.mdb ، يجب إنشاء نسخة rsBook من الكائن Recordset المعرف داخل مكتبة ADO كما هو واضح من المثال التالي.

```
Set rsBook = Server.createobject ("ADODB.Recordset")
```

1.4 الدالة Open( )

تستخدم الدالة Open( ) لقراءة وإسناد البيانات من قاعدة البيانات إلى النسخة المعرفة من الكائن Recordset.

الصيغة العامة

```
<RS>.open "<DataSource>" , <activeConn> , [<CursorType> , <LockType> ]
```

- يمثل <RS> النسخة المعرفة من الكائن Recordset والتي ستستقبل البيانات وتخزنها في الذاكرة.
- يمثل <DataSource> مصدر البيانات الموجود في قاعدة البيانات الذي ربما يكون اسم جدول في قاعدة البيانات أو اسم استعلام معرف داخل قاعدة البيانات، أو عبارة SQL من النوع Select لاسترجاع البيانات من جدول أو أكثر من قاعدة البيانات.
- يمثل <activeConn> النسخة المعرفة من الكائن Connection بغرض الاتصال بقاعدة البيانات.
- يحدد <CursorType> نوع المؤشر الذي سيستخدم مع الكائن Recordset، حيث يختلف نوع المؤشر بناء على القيمة المسندة لهذا المعامل، مع العلم أن هناك مجموعة قيم معرفة داخل ADO، وأن القيمة الافتراضية (إذا لم تكتب شيئاً صراحةً) تستخدم لفتح مجموعة السجلات بغرض القراءة فقط أي لا يمكن التعديل فيها.
- <LockType> يحدد نوع تأمين قاعدة البيانات وإذا لم يكتب صراحة ستفتح قاعدة البيانات للقراءة فقط.

مع العلم أننا سوف نتعرض لكل من موضوع أنواع المؤشر، وموضوع تأمين قاعدة البيانات لاحقا في هذه الوحدة بشكل أكثر تفصيلا.

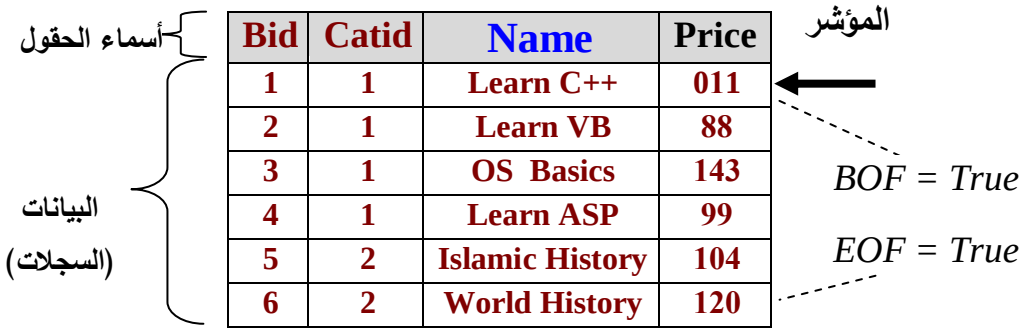
مثال:

يوضح المثال التالي كيفية استرجاع البيانات من الجدول Book المخزن في قاعدة البيانات library.mdb وإسنادها داخل النسخة rsBook (المعرفة من الكائن Recordset).

```
rsBook.Open "Book" , Con
```

2.4 التركيب الداخلي لكائن Recordset

عزيزي الدارس، وجدت أنه من المفضل أن أعرض التركيب الداخلي للكائن Recordset ممثلاً في عرض التركيب الداخلي للنسخة rsBook قبل عرض مثال تفصيلي لاسترجاع وعرض بيانات الجدول Book، وذلك لإيضاح مفهوم المؤشر وبعض الصفات والدوال الهامة للكائن Recordset والتي لا غنى عنها في مثل تلك الأمثلة. يوضح الشكل رقم 2-7 تصوراً رسومياً للتركيب الداخلي للنسخة rsBook بعد إسناد بيانات جدول Book لها، حيث يتضح لنا التالي:



شكل 2-7 : التركيب الداخلي للكائن Recordset

- يتكون الكائن المعرف من Recordset من جزءين:
- أسماء الحقول والتي تشير إلى أسماء حقول الجدول الأصلي (Book).
- البيانات وهي تمثل مجموعة السجلات المخزنة في هذا الجدول.
- وجود مؤشر يحدد موقع السجل الحالي والذي يشير إلى أول سجل عند تنفيذ الدالة Open (أي في بداية تحميل البيانات).
- يشتمل المتغير المعرف من الكائن Recordset على الصفة BOF (Begin Of File) والتي تحتوي على قيمة منطقية، حيث تحتوي على True عندما يشير المؤشر إلى أول سجل فقط وعلى False، عندما يشير المؤشر إلى أي سجل آخر.
- يشتمل المتغير المعرف من الكائن Recordset على الصفة EOF (End Of File) والتي تحتوي على قيمة منطقية، حيث تحتوي على True عندما يشير المؤشر إلى أخر سجل فقط، وعلى False عندما يشير المؤشر إلى أي سجل آخر.
- تسمى السجلات التي تقوم بقراءتها من قاعدة البيانات وتخزينها داخل نسخة الكائن Recordset في الغالب باسم "مجموعة السجلات" ، ولكي تصل إلى كل سجل في هذه المجموعة يمكنك إنشاء تكرار مثل التكرار While Do واستخدام الصفة EOF لكي توقف التكرار بعد الوصول إلى آخر سجل في هذه المجموعة.
- للوصول إلى كل قيمة كل حقل في السجل، يمكنك استخدام أسماء الحقول من الجدول في قاعدة البيانات.

▪هناك عدة دوال في كائن Recordset تتيح لك التنقل بين السجلات المختلفة (نقل المؤشر من سجل إلى آخر) في مجموعة السجلات وهذا يعتمد على نوع المؤشر المستخدم. يوضح الجدول التالي (شكل رقم 3-7) أهم هذه الدوال.

اسم الدالة	الوصف
Movenext	التحرك إلى السجل التالي في مجموعة السجلات.
MovePrevious	التحرك إلى السجل السابق في مجموعة السجلات.
MoveFirst	التحرك إلى أول سجل في مجموعة السجلات.
MoveLast	التحرك إلى أخر سجل في مجموعة السجلات.
Move (2)	التحرك إلى الأمام بمقدار سجلين في مجموعة السجلات.
Move (-3)	التحرك إلى الخلف بمقدار ثلاث سجلات في مجموعة السجلات.

شكل 3-7 : أهم دوال التجول للكائن Recordset

5. استرجاع وعرض البيانات

عزيزي الدارس، يوضح المثال التالي (شكل رقم 4-7) جميع الأوامر اللازمة لاسترجاع بيانات الجدول Book وعرضها للمستخدم، وفيما يلي سيتم سرد سيناريو تنفيذ هذا المثال والذي يتكون من أربع مراحل (مفترضا أنه قد تم إنشاء اسم مصدر البيانات كما هو واضح في بداية الوحدة):

1. إنشاء وفتح نسخة Con من الكائن Connection وربطه باسم مصدر بيانات.
2. إنشاء وفتح نسخة rsBook من الكائن Recordset ، ثم استرجاع البيانات.
3. عرض البيانات من النسخة rsBook مستخدما التكرار while .
4. غلق وتحرير الذاكرة لكل من النسخة rsBook ، والنسخة Con.

```
<html>
<body>
<%
Dim con , rsBook
Set con =Server.CreateObject("ADODB.Connection")
con.ConnectionTimeout=40
con.Open "DSN=libDSN"

Set rsBook=Server.CreateObject("ADODB.Recordset")
rsBook.open "Book" , con

while Not rsBook.eof
Response.write("<B>Name: </B>" & rsBook("bName") & " ")
Response.write("<B>Price: </B>" & rsBook("Price") & "<Br>")
rsBook.movenext
Wend

rsBook.Close
con.Close
Set rsBook= Nothing
Set con = Nothing
%>
</body>

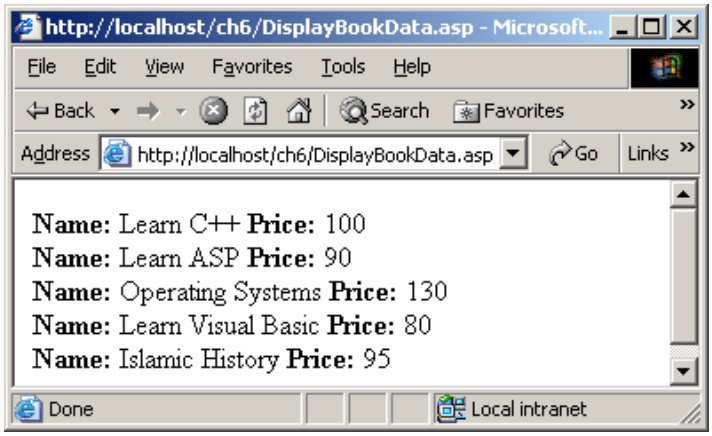
</html>
```

1) إنشاء و فتح كائن Connection

2) إنشاء و فتح كائن Recordset

3) استرجاع وعرض البيانات

4) غلق كل من كائن Connection و كائن Recordset



شكل 4-7 : استرجاع بيانات الجدول Book وعرضها للمستخدم

6. البحث عن معلومات

لعمل برنامج فرعي للبحث عن بيانات وعرض ناتج البحث يتم استخدام نفس الأوامر المذكورة في المثال السابق عدا أمر rsBook.Open، حيث يتم استخدام عبارة SQL من النوع Select بدلاً من اسم الجدول "Book" كمصدر للبيانات المطلوب استرجاعها من قاعدة البيانات. إذا أردنا أن نبحث عن الكتب التي لا تزيد أسعارها عن قيمة معينة، يمكن عمل نموذج يستقبل هذه القيمة من المستخدم ثم إستدعاء ملف ASP والذي ينشئ عبارة Select تتضمن هذه القيمة مع الجزء where.

يوضح الشكل رقم 5-7 كلاً من شفرة HTML المكونة للنموذج وشكل النموذج الذي سيستقبل قيمة البحث، و يوضح الشكل رقم 6-7 كلاً من شفرة ASP المسؤولة عن البحث في قاعدة البيانات وعرض نتيجة البحث.

searchForm.htm

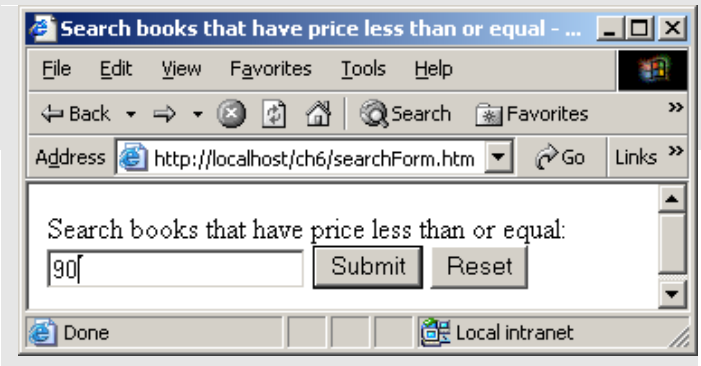
```
<html>

<body>

<form method="POST" action="search.asp">
  <p>Search books that have price less than or equal:
  <input type="text" name="txtPrice" size="20">
  <input type="submit" value="Submit" name="B1">
  <input type="reset" value="Reset" name="B2"></p>
</form>

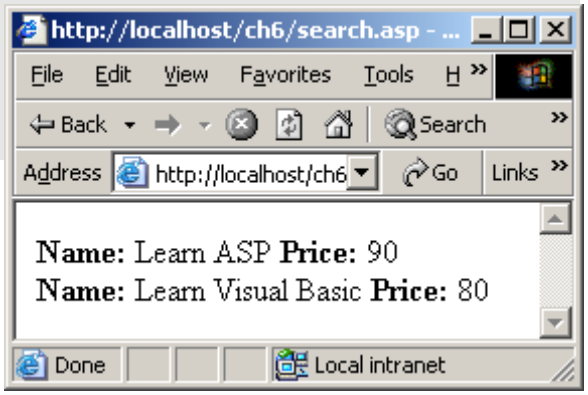
</body>

</html>
```



شكل 5-7 : شفرة HTML المكونة للنموذج وشكل النموذج

```
search.asp
<html>
<body>
<%
Dim con , rsBook , selectSql
reqPrice = request("txtPrice")
selectSql = "select * from Book where price <=" & reqPrice
Set con =Server.CreateObject("ADODB.Connection")
con.ConnectionTimeout=40
con.Open "DSN=libDSN"
Set rsBook=Server.CreateObject("ADODB.Recordset")
rsBook.open selectSql , con
while Not rsBook.eof
Response.write("<B>Name:</B>" & rsBook("bName") & " ")
Response.write("<B>Price:</B>" & rsBook("Price") &
"<Br>")
rsBook.movenext
Wend
rsBook.Close
con.Close
Set rsBook= Nothing
Set con = Nothing
%>
</body>
```



شكل 6-7 : شفرة ASP المسؤولة عن البحث و نتيجة البحث

7. تعديل البيانات

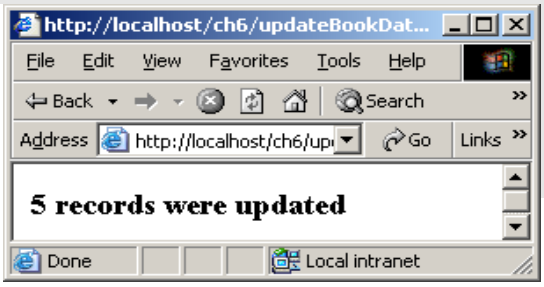
عزيزي الدارس، بمجرد إنشاء مجموعة سجلات، يمكنك أن تعدل أي سجل بها، مع العلم إن التعامل مع السجلات الموجودة ضمن مجموعة السجلات Recordset يعد أكثر كفاءة من العمل مباشرة مع السجلات داخل الجدول.

وكما ذكرنا سابقا عند تنفيذ الدالة ( )Open المنتمية للكائن Recordset ، يتم تحديد نوع المؤشر Cursor Type ونوع التأمين Lock Type فعلى سبيل المثال، إذا

كان نوع المؤشر للقراءة فقط، فلن تتمكن من تعديل أو حذف سجلات من قاعدة البيانات وإذا حاولت هذا فسيحدث خطأ، مع العلم أنه يجب إسناد القيمة 2 لكل من نوع المؤشر Cursor Type ونوع التأمين Lock Type عند التعديل أو الحذف أو الإضافة، كما هو واضح من المثال التالي. وعندما تنتهي من تنفيذ كل التعديلات التي تجريها على مجموعة السجلات Recordset ، يجب أن نستدعي الدالة ( )Update المنتمية للكائن Recordset لحفظ التعديلات التي أجريتها إلى قاعدة البيانات. كما يجب أن تلاحظ إذا توقفت صفحة ASP عن المعالجة نتيجة خطأ، أو إذا توقف خادم الويب عن العمل قبل أن تتم معالجة الدالة ( )Update، فإن التعديلات التي أجريتها لن تتم.

مثال: يوضح المثال التالي (شكل رقم 7-7) كيفية زيادة سعر الكتاب بنسبة10% لجميع الكتب المسجلة داخل الجدول Book.

```
<html>
<body>
<%
Dim con , rsBook , i
Set con =Server.CreateObject("ADODB.Connection")
con.ConnectionTimeout=40
con.Open "DSN=libDSN"
Set rsBook=Server.CreateObject("ADODB.Recordset")
rsBook.open "Book" , con, 2, 2
i = 0
while Not rsBook.eof
 rsBook("Price") = rsBook("Price") * 1.1
 rsBook.update
 rsBook.movenext
 i = i + 1
Wend
Response.write("<H3>" & i & " records were updated </H3>")
rsBook.Close
con.Close
Set rsBook= Nothing
Set con = Nothing
%>
</body>
</html>
```



شكل 7-7 : شفرة ASP المسؤولة عن التعديل (زيادة سعر الكتاب بنسبة 10%)

8. حذف سجلات

عزيزي الدارس، يوجد أكثر من أسلوب لحذف سجل من قاعدة بيانات، وسأعرض فيما يلي أهم أحد هذه الأساليب.

الدالة Delete( ) للكائن Recordset

تستخدم الدالة Delete لحذف سجل من مجموعة سجلات داخل نسخة معرفة من الكائن Recordset ، أو لحذف جميع سجلات هذه النسخة.

الصيغة العامة

```
<RS>.Delete [ <DelParm> ]
```

- يمثل <RS> النسخة المعرفة من الكائن Recordset والتي ستستقبل البيانات وتخزنها داخل الذاكرة.
- يمثل <DelParm> مقداراً ثابتاً معرفاً داخل ADO والذي يحدد أسلوب الحذف حيث يوجد هناك احتمالان:

1. إذا كان المراد حذف السجل الحالي فقط ، فعندئذ لا نكتب شيئاً.
2. إذا كان المراد حذف جميع السجلات المتواجدة داخل <RS> ، فعندئذ نكتب الثابت "adAffectAll".

مثال:

يوضح المثال التالي كيفية استخدام الدالة Delete( ) لحذف سجل من الجدول Book، وإليك سيناريو تنفيذ هذا المثال:

1. يتم استقبال اسم الكتاب من المستخدم من خلال نموذج.
2. إنشاء نسخة rsBook من الكائن Recordset تحتوي على سجل هذا الكتاب مستخدما العبارة Select.
3. اختبار محتويات rsBook.

4. إذا احتوت على الكتاب، سيتم حذفه.

5. إذا لم تحتوي على الكتاب ، يتم إظهار رسالة بذلك.

يوضح الشكل رقم 8-7 كلاً من شفرة HTML المكونة للنموذج وشكل النموذج الذي سيستقبل اسم الكتاب، و يوضح الشكل رقم 9-7 كلاً من شفرة ASP المسؤولة عن حذف بيانات الكتاب من قاعدة البيانات.

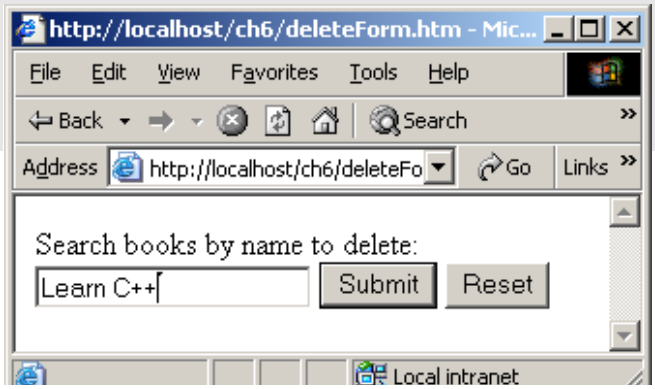
```
<html>

<body>

<form method="POST" action="delete.asp">
  <p>Search books by name to delete:
  <input type="text" name="txtName" size="20">
  <input type="submit" value="Submit" name="B1">
  <input type="reset" value="Reset" name="B2"></p>
</form>

</body>

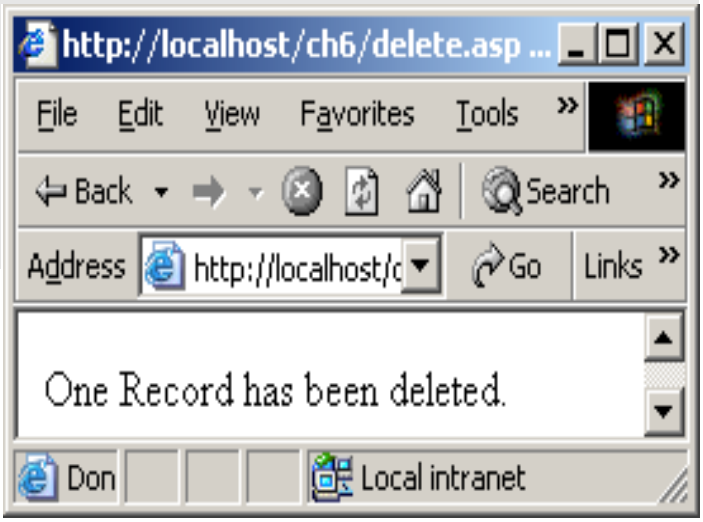
</html>
```



شكل 8-7 : شفرة HTML المكونة للنموذج وشكل النموذج

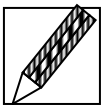
```
<html>

<body>
<%
Dim conLib , rsBook , selectSql
reqName = request("txtName")
selectSql = "select * from Book where bName ='" & reqName &"';"
Set conLib =Server.CreateObject("ADODB.Connection")
conLib.ConnectionTimeout=40
conLib.Open "DSN=libDSN"
Set rsBook=Server.CreateObject("ADODB.Recordset")
rsBook.open selectSql , conLib , 2, 2
if  rsBook.eof and rsBook.bof then
Response.write("There is no books matching your
criteria.")
else
rsBook.delete
rsBook.update
Response.write("One Record has been deleted.")
end if
rsBook.Close
conLib.Close
Set rsBook= Nothing
Set conLib = Nothing
%>
</body>
</html>
```



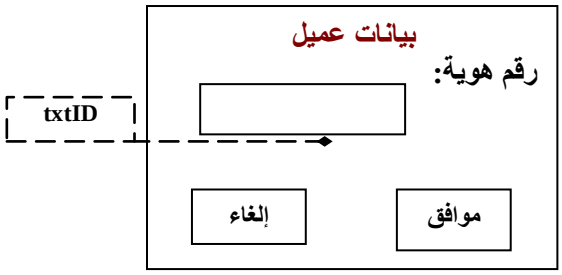
شكل7-9 : شفرة ASP المسؤولة عن حذف الكتاب إن وجد

تمرين (2)



توجد لديك قاعدة بيانات بنك، وتحتوي هذه القاعدة على عدة جداول، منها جدول Client والذي يستخدم لتخزين بيانات العملاء، ويحتوي جدول Client على الحقول التالية:

- ID يمثل رقم هوية العميل.
- Name يمثل اسم العميل.
- Phone يمثل رقم تليفون العميل.
- AccountNo يمثل رقم حساب العميل.



افترض أنك أنشأت اسم مصدر بيانات مستخدماً برنامج ODBC وأعطيت له اسم bankDSN. مستخدما النموذج الموضح، اكتب شفرة ASP المسؤولة عن استقبال رقم العميل والبحث عنه ثم حذفه إن وجد أو رسالة عدم توافره إن لم يتم الحصول عليه.

9. إضافة سجل جديد

عزيزي الدارس، يمكن إضافة سجل جديد في أحد جداول قاعدة البيانات عن طريق استخدام الدالة ( )addNew والتي تستخدم لإضافة سجل فارغ في نسخة معرفة من الكائن Recordset ، ثم بعد ذلك يمكن تخزين البيانات الجديدة داخل حقول هذا السجل، ثم تأكيد الإضافة عن طريق استدعاء الدالة Update.

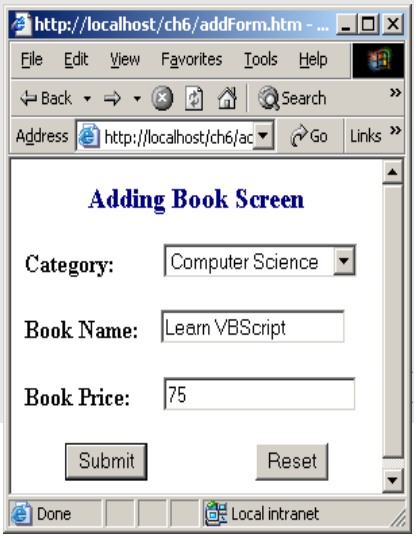
مثال:

يوضح المثال التالي كيفية استخدام الدالة ( )addNew لإضافة سجل إلى الجدول Book ، وفيما يلي سيناريو تنفيذ هذا المثال:

1. يتم استقبال بيانات كتاب جديد من المستخدم (من خلال نموذج).
2. إنشاء نسخة rsBook من الكائن Recordset.
3. إضافة البيانات المستقبلة من النموذج إلى النسخة rsBook.
4. إظهار رسالة تفيد نجاح عملية الإضافة.

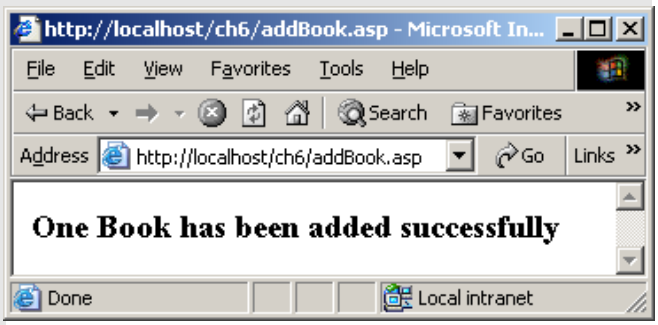
يوضح الشكل رقم 10-7 كلاً من شفرة HTML المكونة للنموذج وشكل النموذج الذي سيستقبل بيانات الكتاب، و يوضح الشكل رقم 11-7 كلاً من شفرة ASP المسؤولة عن إضافة بيانات الكتاب إلى قاعدة البيانات .

```
<html>
<body>
<form method="POST" action="addBook.asp">
  <p align="center"><b>
  <font size="4" color="#000080">
Adding Book Screen</font></b></p>
  <p><b>Category:  </b> <select size="1" name="txtCat">
    <option value="1">Computer Science</option>
    <option value="2">History</option>
    <option selected value="0">-- Select Category --</option>
  </select></p>
  <p><b>Book Name:</b>
  <input type="text" name="txtName">
  </p>
  <p><b>Book Price:</b>
  <input type="text" name="txtPrice">
  </p>
  <p align="center">
<input type="submit" value="Submit"
name="B1">
<input type="reset" value="Reset"
name="B2"></p>
</form>
```



شكل 10-7 : شفرة HTML المكونة للنموذج وشكل النموذج

```
<html>
<body>
<%
Dim con , rsBook , reqCat , reqName , reqPrice
reqCat = request("txtCat")
reqName = request("txtName")
reqPrice = request("txtPrice")
Set con =Server.CreateObject("ADODB.Connection")
con.ConnectionTimeout=40
con.Open "DSN=libDSN"
Set rsBook=Server.CreateObject("ADODB.Recordset")
rsBook.open "Book" , con , 2 , 2
rsBook.addNew
rsBook("cID")=reqCat
rsBook("bName")=reqName
rsBook("Price")=reqPrice
rsBook.update
Response.write("<H3> One Book has been added successfully
</H3>")
rsBook.Close
con.Close
Set rsBook= Nothing
Set con = Nothing
%>
</body>
</html>
```



شكل 7-11 : شفرة ASP المسؤولة عن إضافة الكتاب الجديد

قبل أن تتمكن صفحة ASP من الوصول إلى قواعد البيانات، يجب إنشاء اتصال بقاعدة البيانات، حيث يجب إنشاء "اسم مصدر بيانات" Data Source Name (تختصر DSN) لقاعدة البيانات المخزن بها بيانات الموقع ، ثم نستخدم DSN الذي أنشأناه داخل صفحات ASP، والذي يحتوي على كل من نوع قاعدة البيانات التي تريد أن تتصل بها وعلى مكان قاعدة البيانات وعلى كل من اسم المستخدم وكلمة المرور المطلوبين للتعامل مع قاعدة البيانات

أسئلة تقويم ذاتي



ما وظيفة العبارات التالية؟

- ConnectionTimeout
- EOF
- AddNew
- BOF
- Write
- Redirect
- Expires

الخلاصة

عزيزي الدارس، تعرفنا في هذه الوحدة عى أهم خصائص تقنية ASP ،وهي قدرتها على الاتصال بقواعد البيانات التي تقوم بحفظ وإدارة مجموعات كبيرة من البيانات في شكل جداول، حيث نستطيع تطوير موقع متغير المحتوى يتكون من مجموعة صفحات ASP لجعل هذه المعلومات متاحة للمستخدمين الذين يزورون الموقع ومجموعة صفحات أخرى مسؤولة عن تعديل البيانات داخل قاعدة البيانات.

وبدلاً من حفظ المعلومات في ملفات نصية أو في صفحات ويب ساكنة، نستطيع إعداد صفحات ASP لقراءة هذه المعلومات وتنسيقها، وعندما يزور المستخدم هذه الصفحات فإن المعلومات التي ستعرض داخل هذه الصفحات سيتم إنشاؤها من أحدث المعلومات المتوفرة في قاعدة البيانات.

إجابات التدريبات

```
<%
Dim con , rsBook , selectSql , reqID
reqID = request("txtID")
selectSql = "select * from Book where ID =" & reqID
Set con =Server.CreateObject("ADODB.Connection")
con.Open "DSN=libDSN"
Set rsBook=Server.CreateObject("ADODB.Recordset")
rsBook.open selectSql , con
If rsBook.eof And rsBook.bof then
   Response.write(" The book is not found”)
else
   Response.write("ID:" & rsBook("ID") & " ")
   Response.write("Name:" & rsBook("Name") & "<Br>")
   Response.write("Price:" & rsBook("Price") & " ")
   Response.write("Quantity:" & rsBook("AvailableQty ") &
"<Br>")
end if
rsBook.Close
con.Close
Set rsBook= Nothing
Set con = Nothing
%>
```

```
<%
Dim con , rsClient , selectSql , reqID
reqID = request("txtID")
selectSql = "select * from Client where ID = " & reqID
Set con =Server.CreateObject("ADODB.Connection")
con.Open "DSN=bankDSN"
Set rsClient =Server.CreateObject("ADODB.Recordset")
rsClient.open selectSql , con, 2, 2
if  rsClient.eof and rsBook.bof then
      Response.write("No clients were found.")
else
      rsClient.delete
      rsClient.update
      Response.write("One Client has been deleted.")
end if
rsClient.Close
con.Close
Set rsClient = Nothing
Set con = Nothing
%>
```

مسرد المصطلحات

اسم مصدر البيانات Data Source Name

(تختصر DSN) الذي يقوم بحفظ المعلومات التي تخبر صفحات ASP بكيفية الوصول إلى قاعدة البيانات.

ADO ActiveX Data Object

هي إحدى التقنيات الهامة المقدمة من شركة Microsoft لتسهيل عملية إدارة وصيانة البيانات داخل قاعدة البيانات، تتكون ADO من مجموعة من الكائنات (Objects) التي تقدم لمطوري التطبيقات بصورة عامة وتطبيقات الإنترنت بصورة خاصة أسلوباً سهلاً وكفء للاستعلام، وللإضافة، وللتعديل، ولحذف بيانات في قاعدة بيانات.

المراجع

[1] عبدالكريم العقيل، "المخطط الأزرق لبناء صفحات ويب ديناميكية"، مكتبة جرير، الرياض، 2001م.

[2] د. أحمد أحمد شعبان دسوقى، أ. د. السيد محمود الربيعي، "تقنيات وبرمجة الإنترنت" مكتبة الرشد، الرياض، 2004م.

[3] د. عبد القادر عبد الله الفنتوخ، "الإنترنت للمستخدم العربي"، مكتبة العبيكان، الرياض، 2001م.

[4] د. إبراهيم فتحي معوض، "أساسيات برمجة الإنترنت"، معهد العالمية للحاسب والتقنية، 2005م.