

Chapter Two



Introduction HTML

HTML

- *HTML – HyperText Markup Language*
 - is the standard markup language used to create web pages.
- HTML describes the structure of a website semantically along with cues for presentation
 - its a markup language, rather than a programming language
- *HTML documents consist of a tree of elements and text.*
- *Each element is denoted in the source by a start tag,*
 - *such as "<body>", and an end tag, such as "</body>".*
- Web browsers can read HTML files and render them into visible or audible web pages.

Anatomy of HTML document

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>This is a web page.</p>
  </body>
</html>
```

- The `<!DOCTYPE html>` declaration defines this document to be HTML
- The `<html>` element is the root element of an HTML page
- The `lang` attribute defines the language of the document
- The `<meta>` element contains meta information about the document
- The `charset` attribute defines the character set used in the document
- The `<title>` element specifies a title for the document
- The `<body>` element contains the visible page content
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph

HTML Anatomy ...



- **Elements**

- Elements define the structure and content of objects within a page.
- Enclosed inside angel brackets `<` and `>` : E.g. `<a>` , `<p>`, `<div>`, `<span>`, `<strong>`, `<h1>`...
- meta information is placed in the head element
- content information is placed in the body element

- **Tags**

- The use of less-than and greater-than angle brackets surrounding an element creates what is known as a *tag*.
- Tags most commonly occur in pairs of opening and closing tags.
- Content of the elements always falls between the opening (e.g. `<p>`)and closing (e.g. `</p>`) tag:
E.g. `<a> ...content here...</p>`
- Tags have to be nested such that elements are all completely within each other, without overlapping:

- *Certain start tags and end tags can in certain cases be omitted and are implied by other tags.*

HTML Anatomy ...

Attribute

```
<p class="editor-note">My cat is very grumpy</p>
```



- Attributes
 - Attributes are properties used to provide additional information about an element.
 - The most common attributes include *id*, *class*, *src*, *href*
 - Attributes are defined within the opening tag, after an element's name.
 - Generally attributes include a name and a value.
 - The format for these attributes consists of the attribute name followed by an equals sign and then a quoted attribute value.
- E.g. `<a href="demo.html">simple</a>`

Classifying HTML elements

- **top-level elements:** html, head and body
- head elements: placed inside the head
 - Title, style, link, meta, base, script
 - do not display on the page
- **block-level elements:** flow elements that behave like paragraphs
 - occupy 100% of available width
 - stacked vertically with preceding and subsequent block elements
 - article, h1-h6, header, footer, section, p, figure, canvas,
 - pre, div, ul, ol, dl, table, form, video

Classifying HTML...

- **inline elements**

- phrasing elements that behave like words
- flow horizontally
- usually placed inside block elements
- a (anchor), audio, br, code, img, em, nav, samp
- span, strong, sub, sup, time, var

Syntax of HTML

- All tags begin with < and end with >
- tag name is given immediately following opening <
- unrecognized tags are ignored
- attributes are given following the tag name:
- <tag attribute1="value" attribute2="value" ...>
- ' can be used instead of "
- forgetting to close a quote can result in a blank page
- names and attributes are lowercase
- values are case sensitive
- attributes that have boolean values:
- true: attribute_name="attribute_name"
- false: attribute_name=""

```
<textarea readonly=""></textarea>  
<textarea readonly="readonly"></textarea>
```

HTML syntax ...

- unrecognized tags and attributes are ignored
- elements without closing tags are of the form `<img />`
 - ending `/` is optional for HTML5, needed for polyglot documents
- elements must be well-formed: no bad nesting
 - `<p>Stuff <strong></p></strong>`
- attributes can be in any order
- white space is allowed
 - between tag name and attributes
 - around the `=` sign
 - within attribute value (but should be avoided)

Tag	Description
h1, ..., h6	Header tag h1 to h6
p	paragraphs (Line changes at the end)
span	No line change after span
div	make division between contents
a	hyperlink
center	Move content to center
br	Line break (no closing tag)
hr	horizontal line (no closing tag)
pre	preserve formatting
table	insert table

HTML5

- HTML5 is a cooperation between the
 - World Wide Web Consortium (W3C) and
 - Web Hypertext Application Technology Working Group (WHATWG).
- WHATWG was working with web forms and applications,
- W3C was working with XHTML 2.0.
- In 2006, they decided to cooperate and create a new version of HTML.
-

HTML5: Ground Rules

- Some rules for HTML5 were established:
 - New features should be based on HTML, CSS, DOM, and JavaScript
 - Reduce the need for external plugins
 - Better error handling
 - More markup to replace scripting
 - HTML5 should be device independent
 - Dev process should be visible to the public

New features include:

- Reduce for external plugin
- Semantic elements: header, footer, section, article, others.
- canvas for drawing
 - paths of rectangles, arcs, lines, images
 - mouse events
- Better support for Local offline Storage (variation of cookies)
- audio & video elements
 - including drawing video on canvas
- form controls, like calendar, date, time, email, url, search, color, datetime, datetime-local, month, number, range, tel, week

New API's

- HTML Geolocation
- HTML Drag and Drop
- HTML Local Storage
- HTML Application Cache
- HTML Web Workers
- HTML SSE

Cont...

- All HTML5 elements can have the following:
 - id – uniquely identifies an element in the page
 - style – gives presentation style
 - class – style class, or space-separated list of style classes
 - title – title for the element. Can be used as tool-top display
 - hidden – prevents element from being displayed when set to true
 - contenteditable, draggable, dropzone, spellcheck, and more

- HTML5 defines eight new **semantic** elements.
- All these are **block-level** elements.
- To secure correct behavior in older browsers, you can set the CSS **display** property for these HTML elements to **block**:

```
header, section, footer, aside,  
nav, main, article, figure {  
  display: block;  
}
```



HTML 5 migration

Typical HTML4	Typical HTML5
<code><div id="header"></code>	<code><header></code>
<code><div id="menu"></code>	<code><nav></code>
<code><div id="content"></code>	<code><section></code>
<code><div class="article"></code>	<code><article></code>
<code><div id="footer"></code>	<code><footer></code>

- Reading

- Www.w3school.com

-

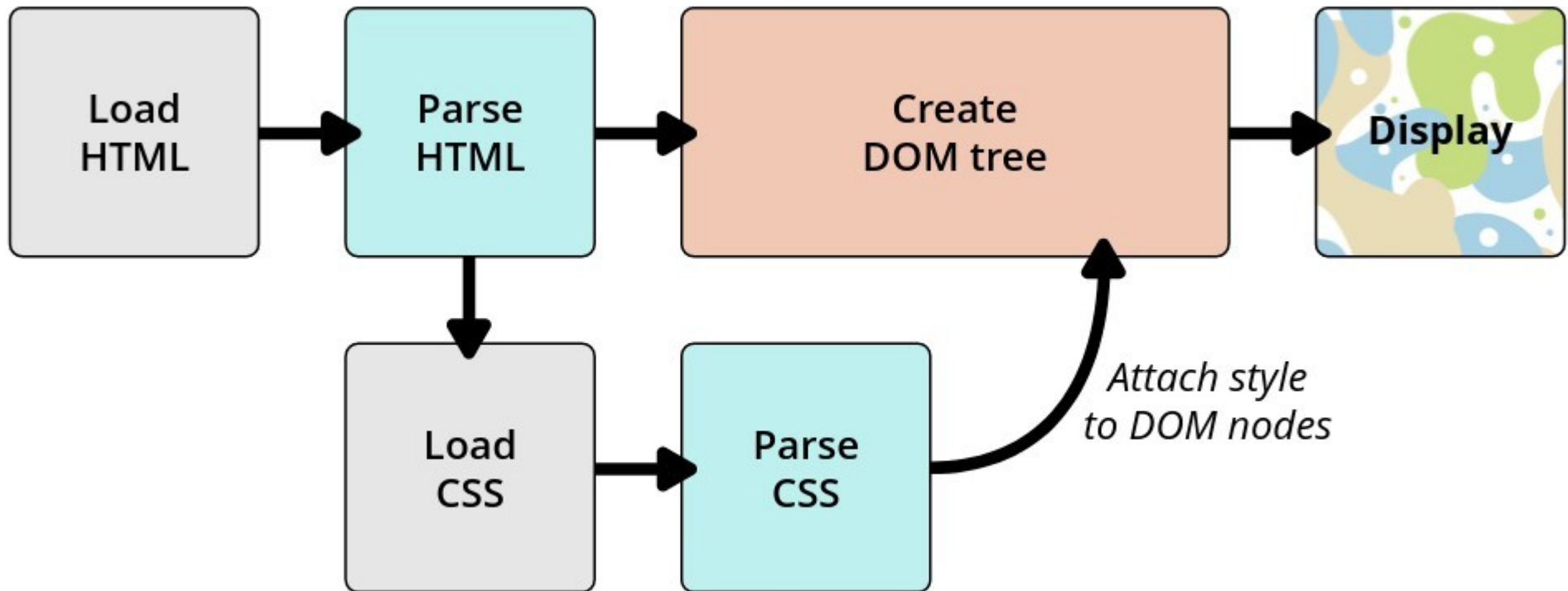
Chapter 3

Introduction to CSS

CSS

- **Cascading Style Sheets (CSS)**: is a simple mechanism for *adding style* (e.g. **fonts**, **colors**, **layouts**) to Web documents.
- Styles provide *powerful control* over the **presentation** of web pages.
- A style sheet consists of a set of **rules**.
- Each **rule** consists of one or more **selectors** and a **declaration block**.
- A **declaration block** consists of a list of **declarations** in curly braces (**{}**).
- Each **declaration** consists of a **property**, a **colon** (**:**), a **value**, then a **semi-colon** (**;**)

How CSS works



Style Sheet Syntax Explained

selector

property

value

rule

```
<style type="text/css">
```

```
{ body {
```

```
    background-color: #000000;
```

```
}
```

```
h1 {
```

```
    font-family: Georgia, "Times New Roman", Times, serif;
```

```
    font-size: 32px;
```

```
    color: #3099D3;
```

```
    text-align: center;
```

```
}
```

```
</style>
```

Creating CSS

- Create a style sheet file and apply CSS to multiple webpages, a single page, or an individual HTML element.
- Three methods:
 - External Style Sheets (preferred choice)
 - Embedded Style Sheets
 - Inline Styles (least desirable)

CSS scope

- Create a style sheet file and apply CSS to multiple webpages, a single page, or an individual HTML element.
- Three methods:
- External Style Sheets (preferred choice)
- Embedded Style Sheets
- Inline Styles (least desirable)
-

Example: inline

```
<!-- css.html -->

<!DOCTYPE html>
<html>
<head>
  <title>CSS Tutorial</title>

</head>
<body>

  <h3 style="color:blue"> Heading 1 </h3>
  <h3 style="color:blue"> Heading 3 </h3>
  <h3 style="color:blue"> Heading 3 </h3>

</body>
</html>
```

Example: Embedded CSS

```
<!-- css.html -->

<!DOCTYPE html>
<html>
<head>
  <title>CSS Tutorial</title>

  <style type="text/css">
    h3.h3_blue{      /*change color to blue*/
      color: blue;
    }

    h3.h3_red{       /*change color to red*/
      color:red;
    }
  </style>
</head>
<body>

  <h3 class='h3_blue'> Heading 1 </h3>
  <h3 class='h3_blue'> Heading 3 </h3>
  <h3 class='h3_blue'> Heading 3 </h3>

  <h3 class='h3_red'> Heading 1 </h3>
  <h3 class='h3_red'> Heading 3 </h3>
  <h3 class='h3_red'> Heading 3 </h3>

</body>
</html>
```

Example: External

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Tutorial</title>
  <link rel="stylesheet" type="text/css" href="asset/css/my_css.css">
</head>
<body>

  <h3 class='h3_blue'> Heading 1 </h3>
  <h3 class='h3_blue'> Heading 3 </h3>
  <h3 class='h3_blue'> Heading 3 </h3>

  <h3 class='h3_red'> Heading 1 </h3>
  <h3 class='h3_red'> Heading 3 </h3>
  <h3 class='h3_red'> Heading 3 </h3>

</body>
</html>
```

```
/* asset/css/my_css.css */

h3.h3_blue{
  color: blue;
}

h3.h3_red{
  color:red;
}
```

Basic CSS Selectors

- There are three types of selectors in CSS,
- **Element** : can be selected using it's name
 - e.g. 'p', 'div' and 'h1' etc.
- **Class** : can be selected using '.className' operator
 - e.g. '.h3_blue'.
- **ID** : can be selected using '#idName'
 - e.g. '#my_para'.

Example: selector

```
<!-- css.html -->

<!DOCTYPE html>
<html>
<head>
  <title>CSS Selectors</title>
  <link rel="stylesheet" type="text/css" href="asset/css/my_css.css">
</head>
<body>
  <h3>CSS Selectors</h3>

  <p class='c_head'> Paragraph with class 'c_head' </p>
  <p id='i_head'> Paragraph with id 'i_head' </p>

</body>
</html>
```

```
/* asset/css/my_css.css */

/*element selection*/
h3 {
  color: blue;
}

/*class selection*/
.c_head{
  font-family: cursive;
  color: orange;
}

/*id selection*/
#i_head{
  font-variant: small-caps;
  color: red;
}
```

More selectors

TABLE 2.1. LIST OF SELECTORS

Selectors	Description
h1, p, span etc.	element selector
.className	class selector
#idName	id selector
*	Universal selector (selects everything)
h1.className	select h1 with class 'className'
h1#idName	select h1 with id 'idName'
p span	descendant selector (select span which is inside p)
p > span	child selector ('span' which is direct descendant of 'p')
h1, h2, p	group selection (select h1, h2 and p)
span[my_id]	select 'span' with attribute 'my_id'
span[my_id=m_span]	select 'span' with attribute 'my_id=m_span'

Precedence

- **Precedence**
 - **Local > Internal > External**
- **Selector Priority level :**
 - **ID (highest priority)**
 - **Class**
 - **Element**

