

IT 111 – INTRO TO COMPUTING

CHAPTER 06

BASIC HTML & CSS

MARK IAN M. MUKARA

IT FACULTY

HTML

- The standard markup language for creating Web pages.
- **HTML** stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup.
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

HTML VERSIONS

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

HTML TAGS

Four basic HTML tags:

1. <HTML>
2. <HEAD>
3. <TITLE>
4. <BODY>

HTML TAGS

- **<!DOCTYPE>**: Tells the browser which version of HTML the document uses
- **<html>**: The root element that contains all other elements
- **<head>**: Includes metadata like the page title, character encoding, and links to external resources
- **<title>**: Sets the title of the page, which is displayed in the browser's title bar or tab
- **<link>**: Used to link external resources like stylesheets
- **<style>**: Contains inline CSS or references an external stylesheet
- **<script>**: Contains JavaScript code or references an external script file
- **<body>**: Contains all the visible content of the web page, including text, images, and other elements

HTML TAGS

- **<h1> to <h6>**: Headings that represent hierarchical sections of the content, with **<h1>** being the highest level and **<h6>** the lowest
- **<p>**: Defines a paragraph of text.
- ****: Embeds images on the page.
- ****: Defines an unordered (bulleted) list.
- ****: Defines an ordered (numbered) list.
- ****: Defines list items within **** or ****.
- **<div>**: A generic container for grouping and styling content.
- ****: A generic inline container for styling a portion of text.
- **<table>**: Defines a table.
- **<hr>**: Creates a horizontal line to separate content.
- **
**: Inserts a line break.

HTML TAGS

HTML tags are keywords (tag names) surrounded by angle brackets:

```
<tagname>content</tagname>
```



Example:

```
<p>My First Paragraph.</p>
```

- HTML tags normally come in pairs like `<p>` and `</p>`
- The first tag in a pair is the start tag, the second tag is the end tag
- The end tag is written like the start tag, but with a forward slash inserted before the tag name

SAMPLE HTML DOC

```
<!DOCTYPE html>  
  
<html>  
<head>  
<title>Page Title</title>  
</head>  
<body>  
  
<h1>My First Heading</h1>  
<p>My first paragraph.</p>  
  
</body>  
</html>
```

Example Explained

- The `<!DOCTYPE html>` declaration defines this document to be HTML5
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about 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

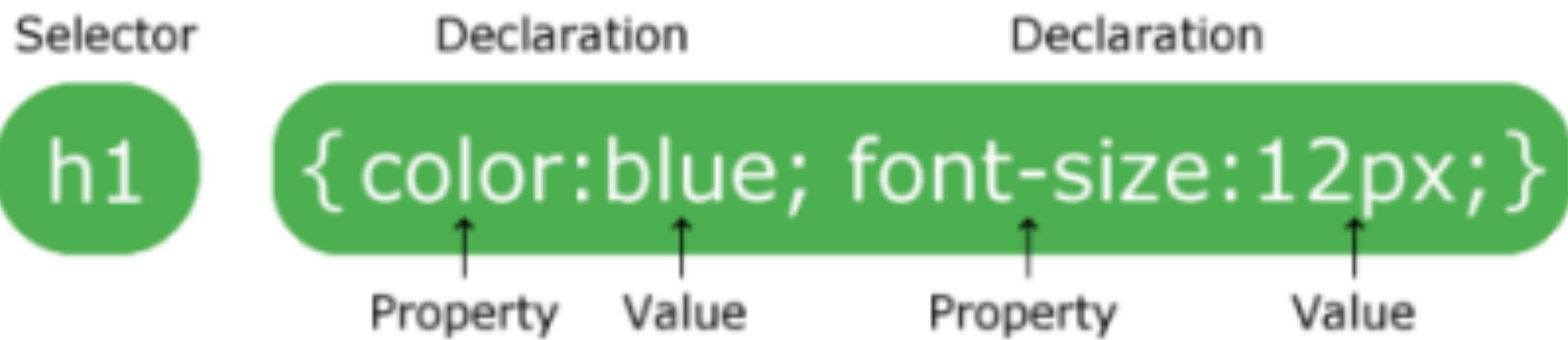
CSS

» Cascading Style Sheet

» Simple design language that describes the style of an HTML document. Through CSS, each element in the HTML document can be modified according to its color, size, backgrounds, layout, etc. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files.

CSS SYNTAX

A CSS rule-set consists of a selector and a declaration block:



CSS SYNTAX

HTML document with CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: red;
  text-align: center;
}
</style>
</head>
<body>

<p>Hello World!</p>
<p>These paragraphs are styled with CSS.</p>

</body>
</html>
```

OUTPUT

Hello World!

These paragraphs are styled with CSS.

3 WAYS TO INSERT CSS

I Internal CSS

An internal style sheet may be used if one single page has a unique style. Internal styles are defined within the HTML page:

```
<head>
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```

3 WAYS TO INSERT CSS

1 Inline CSS

An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<h1 style="color:blue; margin-left:30px;">This is a heading</h1>
```

3 WAYS TO INSERT CSS

I External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!

Each page must include a reference to the external style sheet file inside the `<link>` element. The `<link>` element goes inside the `<head>` section:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

3 WAYS TO INSERT CSS

1 External CSS

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension.

Here is how the "mystyle.css" looks:

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

3 WAYS TO INSERT CSS

1 External CSS

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Note: Do not add a space between the property value and the unit (such as margin-left:20px). The correct way is: margin-left:20px;

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

CSS SELECTORS

CSS SELECTORS

CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

- **element Selector**
- **id Selector**
- **class Selector**
- **grouping Selector**

- **ELEMENT SELECTORS**

The element selector selects elements based on the element name.

You can select all elements on a page like this (in this case, all elements will be center-aligned, with a red text color):

```
p {  
    text-align: center;  
    color: red;  
}
```

- **ID SELECTORS**

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element should be unique within a page, so the id selector is used to select one unique element.

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

- **ID SELECTORS**

The style rule below will be applied to the HTML element with id="para1":

```
#para1 {  
    text-align: center;  
    color: red;  
}
```

Note: An id name cannot start with a number.

• CLASS SELECTORS

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

In the example below, all HTML elements with class="center" will be red and centeraligned:

```
.center {  
    text-align: center;  
    color: red;  
}
```

- **CLASS SELECTORS**

You can also specify that only specific HTML elements should be affected by a class.

In the example below, only elements with class="center" will be center-aligned:

```
p.center {  
    text-align: center;  
    color: red;  
}
```

• GROUPING SELECTORS

If you have elements with the same style definitions, like this:

```
h1 {  
    text-align: center;  
    color: red;  
}  
  
h2 {  
    text-align: center;  
    color: red;  
}  
  
p {  
    text-align: center;  
    color: red;  
}
```

- **GROUPING SELECTORS**

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

In the example below we have grouped the selectors from the code above:

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```


CSS & JAVASCRIPT

CSS & JAVASCRIPT

Purpose:

- **CSS**: CSS is primarily used for styling and layout. It's responsible for defining how the content on a web page should be presented visually. This includes setting colors, fonts, positioning elements, and creating responsive designs.
- **JavaScript**: JavaScript is a programming language used to add interactivity and dynamic behavior to web pages. It enables you to create things like pop-up windows, interactive forms, animations, and perform various actions in response to user input.

CSS & JAVASCRIPT

Styling vs. Functionality:

- **CSS**: CSS is all about styling and appearance. It focuses on the presentation of the content, such as fonts, colors, spacing, and layout.
- **JavaScript**: JavaScript is about functionality and behavior. It allows you to change or manipulate content dynamically, handle user events, and perform complex calculations.

WIREFRAME & MOCKUP

WIREFRAME

- Wireframes are low-fidelity, basic, and simplified representations of a user interface (UI) or webpage. They focus on the structural layout and functionality rather than visual design.
- Wireframes typically consist of simple shapes, boxes, and lines to outline the placement of elements such as text, images, buttons, and navigation components.
- Wireframes are used to define the layout, structure, and flow of a design. They help in making decisions about the information hierarchy and user interactions.
- Wireframes are created early in the design process and are useful for sketching out ideas and concepts without getting into the details of colors, typography, or images.

WIREFRAME EXAMPLE

The wireframe illustrates a website layout for a company specializing in weather monitoring equipment. The header features a navigation bar with links to Home, About Us, Products, Blog, News, Support, and Contact, along with a search bar and a phone number (800-888-8888). A sidebar on the left lists various product categories. The main content area includes a large central placeholder for a featured product or image, and a grid of six smaller product cards below it.

View Cart Order History

Products, Parts, Keywords Search

Home | About Us | Products | Blog | News | Support | Contact 800-888-8888

Rainfall Sensors
Wind Anemometer
Weather Stations
Controllers
Indicators/Displays
Temp/Humidity Sensors
Barometric Pressure
Solar Radiation
Replacement Parts

Featured Products

Product Title SKU Lorem ipsum dolor sit amet, consectetur adipiscing elit. More Info »	Product Title SKU Lorem ipsum dolor sit amet, consectetur adipiscing elit. More Info »	
Product Title SKU Lorem ipsum dolor sit amet, consectetur adipiscing elit. More Info »	Product Title SKU Lorem ipsum dolor sit amet, consectetur adipiscing elit. More Info »	

MOCKUP

- Mockups are higher-fidelity, more detailed visual representations of a design. They provide a clearer picture of how the final product will look in terms of colors, typography, and style.
- Mockups often include real content and imagery, making them look much closer to the final product than wireframes.
- Mockups are used to visualize the aesthetics and design elements of a project, which is crucial for client or stakeholder presentations and for getting design approval.
- Mockups are typically created after wireframes and are a more polished representation of the final product.

MOCKUP EXAMPLE

