MODULE V: BASIC HTML & CSS

COURSE OUTCOMES

Develop a simple web page using HTML and CSS basics.

LEARNING OUTCOMES

- Identify the basic concept of HTML and CSS
- Create a simple web page using HTML and CSS basics.

Throughout this course, you'll learn about the underlying structure of the web – HTML or Hypertext Markup Language. You'll learn how to use this tree-like structure to create websites. You'll also learn how to apply styling to a website through CSS or Cascading Style Sheet. You'll learn about CSS syntax, selectors, and units. Along the way, you'll also learn how to code editors and a browser's developers tools.

HTML

HTML is 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

A Simple HTML Document

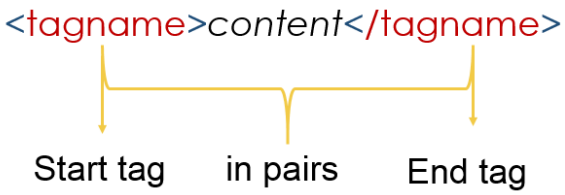
```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<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 element defines a paragraph

HTML TAGS

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



Example:

My First Paragraph.

- HTML tags normally come in pairs like and
- 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 slashinserted before the tag name

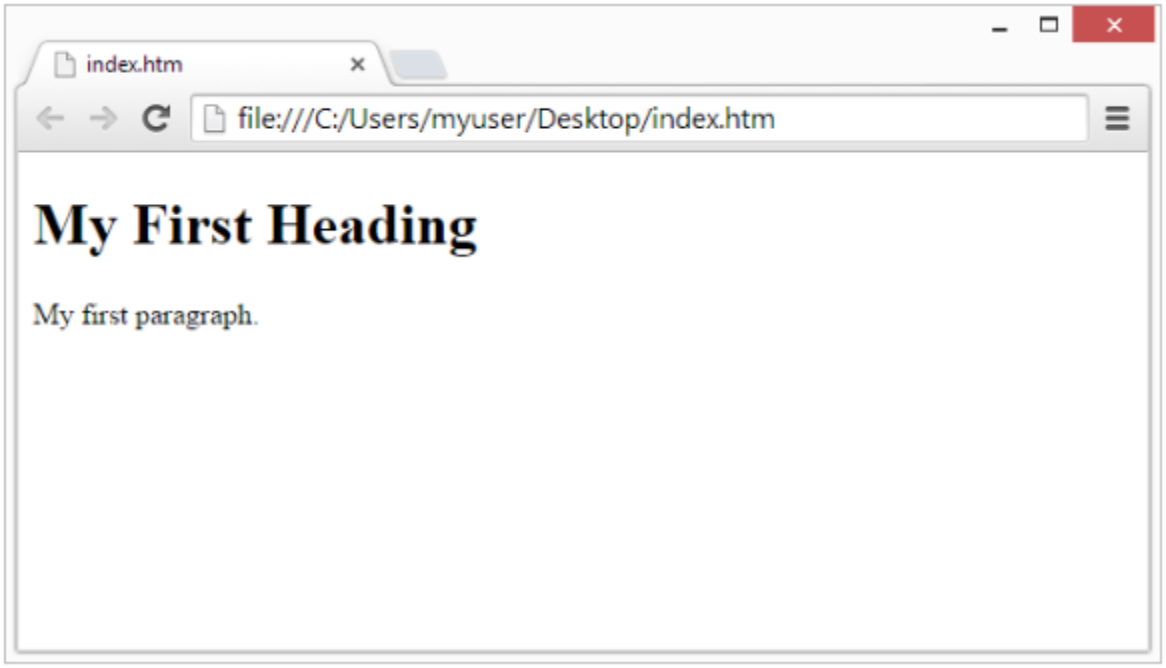
Four basic HTML tags:

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

Web Browsers

The purpose of a web browser (Chrome, IE, Firefox, and Safari) is to read HTML documents and display them.

The browser does not display the HTML tags, but uses them to determine how to display the document:



HTML PAGE STRUCTURE

Below is a visualization of an HTML page structure:

```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

Only the <body> area (the white area) is displayed by the browser.

The <!DOCTYPE> Declaration

The `<!DOCTYPE>` declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The `<!DOCTYPE>` declaration is not case sensitive.

The `<!DOCTYPE>` declaration for HTML5 is:

```
<!DOCTYPE html>
```

HTML Versions

Since the early days of the web, there have been many versions of HTML:

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

ASSESSMENT

Instructions: Create your first HTML file follow the instructions and screenshot your first HTML file in a web browser.

Step 1: Open Notepad/TextEdit or any text editor available in your computer.

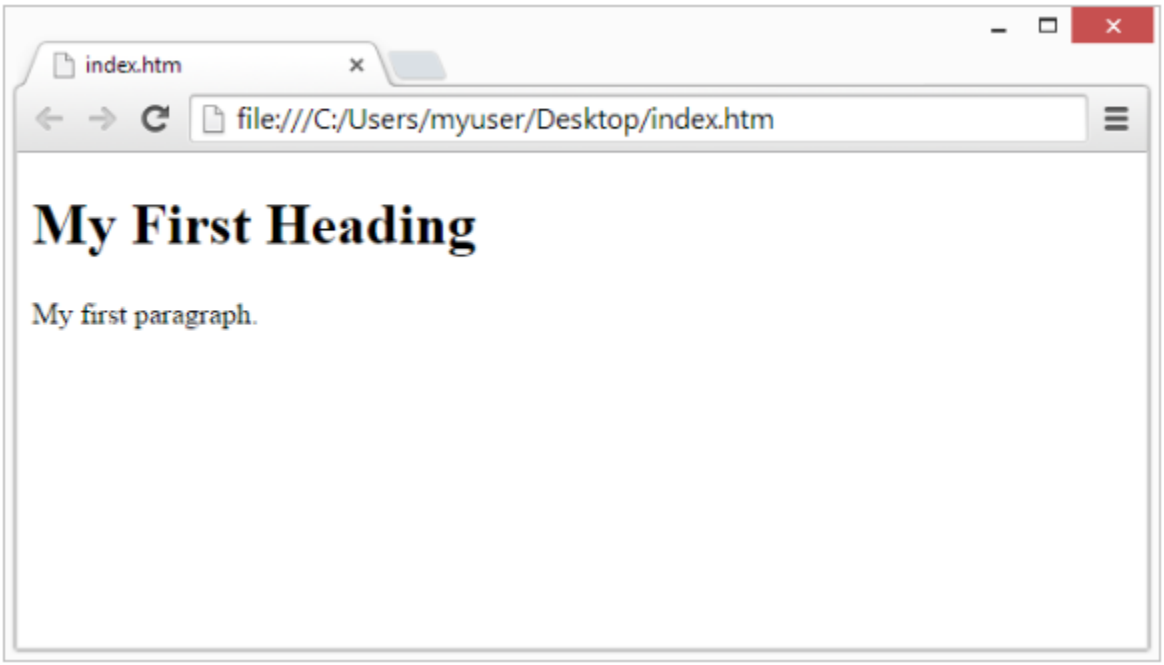
Step 2: Write/Copy some HTML into notepad

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Step 3: Save the HTML Page (.html, .htm)

Step 4: Open the saved HTML file in your favorite browser (double click on the file, or right-click - and choose "Open with").

The result will look much like this:



ASSESSMENT

Instruction: *Create a simple web page that will display the following:*

Name

Birthday

Address

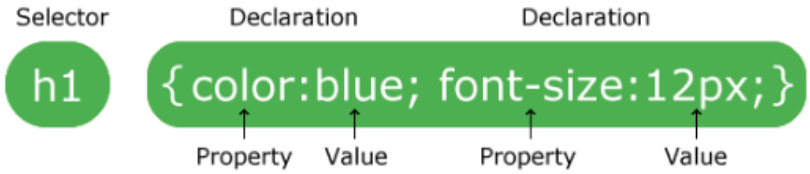
Save your file as lastname-act1.html

CSS

CSS or Cascading Style Sheet is 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:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

In the following example all <p> elements will be center-aligned, with a red text color:

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

Example:

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.

CSS Selectors

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

The element Selector

The element selector selects elements based on the element name.

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

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

The id Selector

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.

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!

The class Selector

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 center-aligned:

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

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

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

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

HTML elements can also refer to more than one class.

In the example below, the <p> element will be styled according to class="center" and to class="large":

```
<p class="center large">This paragraph refers to two classes.</p>
```

Note: A class name cannot start with a number!

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;
}
```

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 Comments

Comments are used to explain the code, and may help when you edit the source code at a later date.

Comments are ignored by browsers.

A CSS comment starts with /* and ends with */. Comments can also span multiple lines:

```css
p {
  color: red;
  /* This is a single-line comment */
  text-align: center;
}

/* This is
a multi-line
comment */
```

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

External Style Sheet

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:

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

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;
}
```

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;

Internal Style Sheet

An internal style sheet may be used if one single page has a unique style.

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

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

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

Inline Styles

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.

The example below shows how to change the color and the left margin of a <h1> element:

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

Tip: An inline style loses many of the advantages of a style sheet (by mixing content with the presentation). Use this method sparingly.

Multiple Style Sheets

If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.

Example

Assume that an external style sheet has the following style for the <h1> element:

```
h1 {
  color: navy;
}
```

then, assume that an internal style sheet also has the following style for the <h1> element:

```
h1 {
  color: orange;
}
```

If the internal style is defined after the link to the external style sheet, the <h1> elements will be "orange":

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
  color: orange;
}
</style>
</head>
```

However, if the internal style is defined before the link to the external style sheet, the <h1> elements will be "navy":

```
<head>
<style>
h1 {
  color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Cascading Order

What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

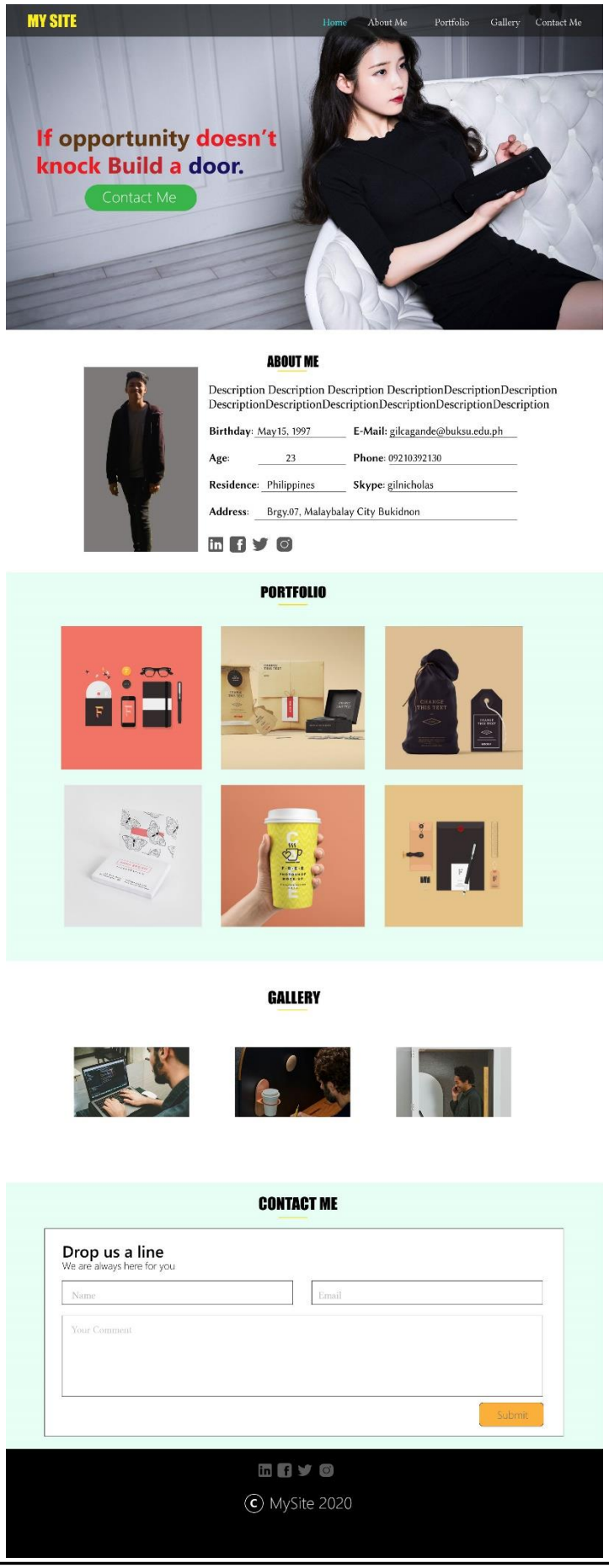
So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

ASSESSMENT

Instruction: Open your previous HMTL activity and add color, style, alignment, pictures, or anything that will aesthetically improve your webpage.

FINAL ASSESSMENT

Instruction: Create a web page with a 5 menus: Home, About Me, Portfolio, Gallery, and Contact. See picture (Not necessarily you need to copy it or come up exactly look like it, but if you have the skills then go for it). See the rubrics below for guidance.



Rubrics for Web Design

Score Level	Design & Layout	Navigation	Completion
5 (Extraordinary)	The website has an exceptionally attractive layout I is easy to locate important elements	Links are clearly labelled, consistency placed, allow reader to easily move.	100% Complete
4 Satisfactory	The website has an attractive layout It is easy to locate important elements	Links are labelled, and allow reader to easily move.	75% Complete
3 Average	The website has usable layout but may appear busy or boring	Links reader to easily move. Some links seem to be missing	50% Complete
2 Unsatisfactory	The website has a cluttered or confusing layout.	Links seem to be missing and don't allow the ready to easily navigate	25% Complete
1 Poor	The website has unusable layout	Links don't take	Less than 10% complete