**Lecture 4**

**HTML Elements and HTML page structure**


## *1- The Standard Character set*

In digital word there are different devices and computer system, therefor using a standard character set enable the global exchange of information and make the text readable and interpreted accurately on various platforms or browses. Without a standardized character set, the same numerical value could be interpreted differently by different computers, resulting in **garbled** or incorrect text. The following is a list of different character sets:

**1-** The **ASCII** Character set was the first character encoding standard for the web. It defined 128 different characters that could be used on the internet: the English letters, Numbers, and special characters like ! $ + - ( ) @ < >.

**2-**The **ANSI** Character Set, ANSI (Windows-1252) was the original Windows character set: it is *dientical to ASCII for the first 127 characters, Special characters from 128 to 159, Identical to UTF-8 from 160 to 255*

**3-Unicode** character set: is a standardized collection of letters, numbers, symbols and emoji, it assigns a unique numerical value, called a "code point," to every character, example (A= U+004A in hex). Before Unicode, there were different character sets, causing a problem, if a file created on a computer might display garbled on another if they used **different** character set. Unicode aim s create a universal standard to solve that problem.

4-**Unicode Transformation Format** (**UTF**) It define how the Unicode code points are translated into binary numbers to be stored in the computer. UTF version 8 covers almost all of the characters and symbols in the world. Almost every webpage is stored in UTF-8.

The Standard is implemented in HTML, XML, JavaScript, E-mail, PHP, Databases and in all modern operating systems and browsers. The most commonly used encodings are UTF-8, UTF-16 and UTF-32. The Unicode code space ranges from U+0000 to over than million symbol.

Currently UTF-8 became the common standard set, use a variable-length character encoding (1 to 4 bytes long). UTF-8 is compatible with ASCII *for the values from 0 to 127 and Identical to ANSI and 8859-1 from 160 to 255.* UTF-8 is the preferred encoding for e-mail and web pages.

**5-The ISO-8859-1** Character Set: is the default character set for HTML 4. This character set supported 256 different character codes. HTML 4 also supported UTF-8.

## 2-HTML document Elements

HTML document divided into two main parts: the head and the body, each part contains elements, attributes and embedded tags, which are explained in the following:

### 1-Tag:

Tags represent the structural components of a document, such as  . Tages are the starting and ending parts of an HTML element, used to specify special regions to the web browser.

### 2-Element:

Elements are the building blocks of this language. They are in every part of the page, including the header, footer, body, and text. Each element is an individual component on a web page or document. Elements are formed by tags. A complete HTML element consists of both the opening <tag> and closing </tag> tags and what's inside those tags. ☞ However, not all the HTML elements have closing words. They go by the names void, singleton, or empty elements.

*Example:* ****This is the content.****

*Example:* This text contains<br>a line break.

### 3-Attribute:

HTML Attributes are special words used within the opening tag of an HTML element. They provide additional information about HTML elements. HTML attributes are used to adjust the element's behavior, appearance, or functionality in a variety of ways. Attributes tell the browser how to render the element. Every tag has two sections, the attribute name and its value, separated by an equals sign (=) such as (**name="value"**).

⇒ *NOTE* : Attributes always placed in the opening tag of an element. It generally provides additional styling (attribute) to the element.

*Example:* **<tagname attribute_name="attribute_value">** ***content*** **</tagname>**

*Example:* **<p** align="center">This is paragraph.** **

- In this example, the element consists of three basic parts: an opening tag, the element's content, and finally, a closing tag.
- ** ** - opening **paragraph** tag
- **Attribute**: align="center"
- **Element Content** - paragraph content
- ** ** - closing tag

## 3-HTML Basic Document Structure

### 3-1-HTML document deceleration

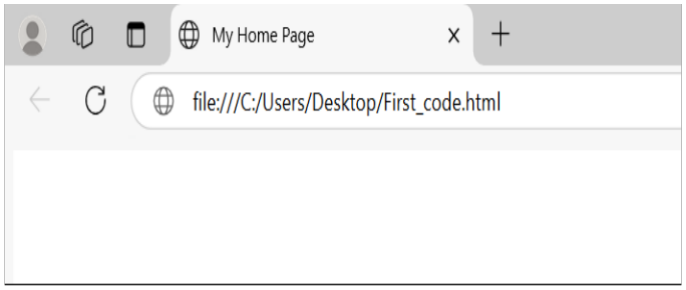
Currently there are different document types on the web, thus there must be to declare the document type. **DOCTYPE** declaration, is the HTML document type declaration, it is the first line of code required in every HTML or XHTML document, Th doctype's purpose is to tell the browser what version of HTML it should use to render the document. (it means which version is used to design the page). <!DOCTYPE html> is **not** an HTML **tag** or **element**, it an instruction to help the browser to display a web page correctly,

therefor **without** <! DOCTYPE html> the HTML code can still **run**, but it may face some drawbacks.

⇒ ***Note*:** <! DOCTYPE html> tage does not appear in the web page, when you run the code, as explained in this example below, where the HTML5 declaration **Syntax is**:

<!DOCTYPE. html >

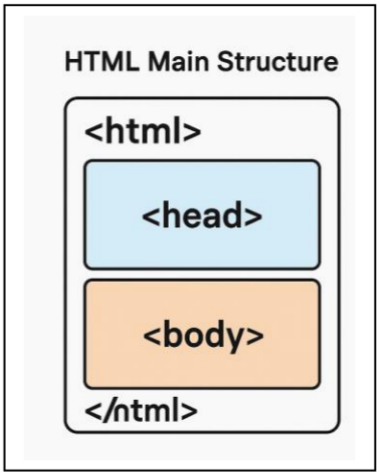
**The next example is the basic code to create a web page using <!DOCTYPE html>**

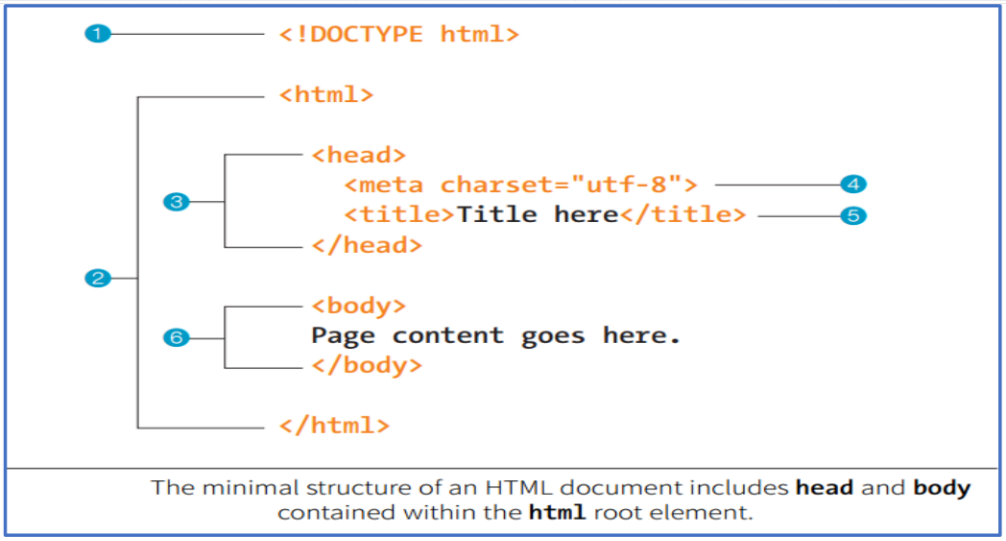
| **HTML code with doc. definition** | **The result** |
|---|---|
| <!DOCTYPE html><br><HTML><br><HEAD><br><TITLE>My Home Page </TITLE><br></HEAD><br><BODY><br></BODY><br></HTML> |  |

***Tip*:** The <!DOCTYPE> declaration is NOT case sensitive. You can write it <!DOctyPE.HTmL> or <!Doctype Html> or any other form.

**The next step after the declaration, is building the HTML document elements Now let's descript them in the following:**

**3-2** The <html> **element**: the entire document is contained within the html element. html element, it is divided into a head and a body. <**html**> element called the **root** element, because it contains all the content of the page, except the Document Type Definition (DTD).






The minimal structure of an HTML document includes **head** and **body** contained within the **html** root element.

**3-3-The <head> element:** contains elements that are not **rendered** as part of the content, it is a container for metadata (data about data) such as its title, style sheets, scripts, and metadata. Now let's explain the **head elements**:

- The element <**Meta**>: Metadata is data about the HTML document and is **not** displayed on the webpage, meta element **specifies the Character set**, Keywords, author, publishing status, viewport settings, Page description that can be used by search engines of the document and. (Metadata is not displayed in the page)

## Examples

**1-Define the character set used:**

<meta charset="UTF-8">

**2-Define keywords for search engines:**

<meta name="keywords" content="HTML, CSS, JavaScript">

**3-Define the author of a page:**

<meta name="author" content="John Doe">

**4-Setting the viewport to make the website look good on all devices.** It will be smaller on a mobile phone than on a computer screen. The viewport is the user's visible area of a web page. The following ***Example*** *on* <meta> element must be included in the designed web pages:

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

This line gives the browser instructions on how to control the page's dimensions and scaling. The **width=device-width**, sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser. The next figure is an example of a web page without and without the viewport meta tag:


**Without the viewport meta tag** **With the viewport meta tag**

**5-Define description of the web page:** gives users a quick summary about web page contents, and what the page is about, it help to improve the overall user experience.

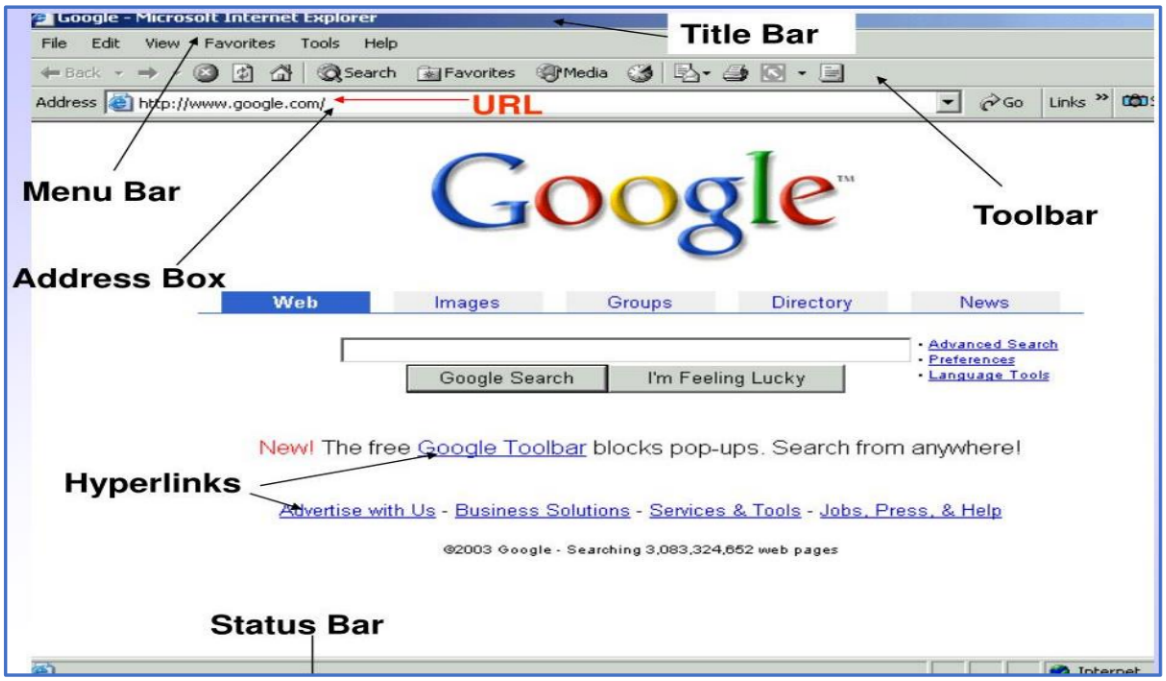
<meta name="description" content="Free Web tutorials">

The content="Free Web tutorials" attribute provides the page description.

- The **element <title>** element defines the title of the document the title is mandatory element in the head part. According to the HTML specification, every document must contain a descriptive title. The title must be text-only, and it is shown in the browser's **title bar** or in the **page's tab**. The page title is very important for the search engine, <title> displays a title of the page in search-engine results and help the engine

algorithms to decide the order when listing pages in search results. According to the HTML specification, every document must contain a descriptive title.

**Example:** <title>Web development Class</title> The HTML <style>




- The **element <style>** contains the document style information for a single HTML page such as the color, font, position. The <style> allows the CSS code to be placed within the HTML document. **Example:**

```
<style>
 h1 {color: red;}
</style>
```

***Tip***: it is better to put the styles in external stylesheets and apply them using <link>.

- The **<link> element** defines the relationship between the current document and an external resource. <link> element provide efficient code management, improves website performance, and enhances the overall user experience. The **<link>** element require **rel** and **href** attribute

- The **rel** attribute represent the "**relationship**" it define relationships between the between the current document and the linked document. The **rel** is one of the key features of the <link> element. **Syntax**: rel="value"
- The **href** attribute specify the location of the linked resource, using the URL of the page that the link goes to. If the href attribute is not present, the <a> tag will not be a hyperlink.

**Example**: use the <link> tag to link to external style sheets named "mystyle.css.

```
<link rel="stylesheet" href="mystyle.css">
```

- The **<script> element** used for embedding or referencing executable code, most commonly used to define client-side JavaScripts, it allow to directly include JavaScript code within the HTML document. This is useful for small scripts or when the web designer need to dynamically modify the page's content.

**Example**:

```
<script>
 function myFunction()
{
 document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
```

***Tip***: <script> tags can be placed in the <head> or <body> of an HTML document.

The **<base>** element It defines a base URL for all relative URLs within the HTML document. This is particularly useful when there are many relative URLs (e.g., links to images, stylesheets, or other pages) **<base>** help to ensure they all resolve correctly, regardless of the current page's URL. **<base>** can also set a default target for all hyperlinks on the page. The <base> tag must have either an **href** or a **target attribute** present, or both.

- **href**: attribute specifies the base URL.
- **target**: attribute specifies the default target for all hyperlinks on the page.