



# *Client-side Technologies*

*Dr. Niveen Nasr El-Den*  
*iTi*



Knowing doesn't build Skills..  
Practicing do!!

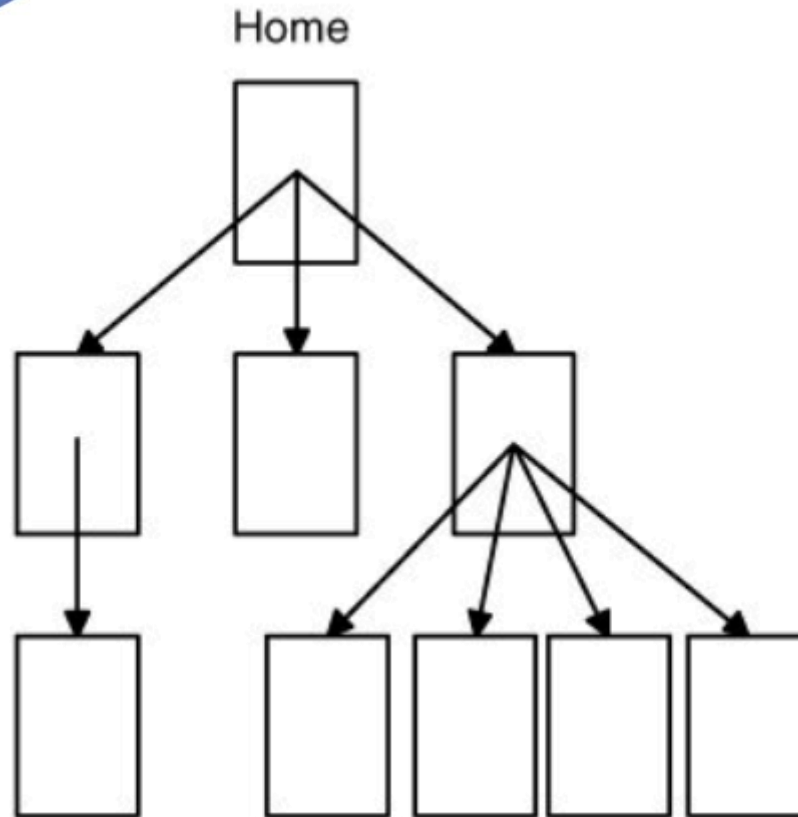
The background features abstract, curved shapes in various shades of blue and purple. On the left, there are overlapping shapes in dark blue, medium blue, and light blue. On the right, there are shapes in light purple and dark purple. The central area is white, providing a clean space for the text.

*Day 1*



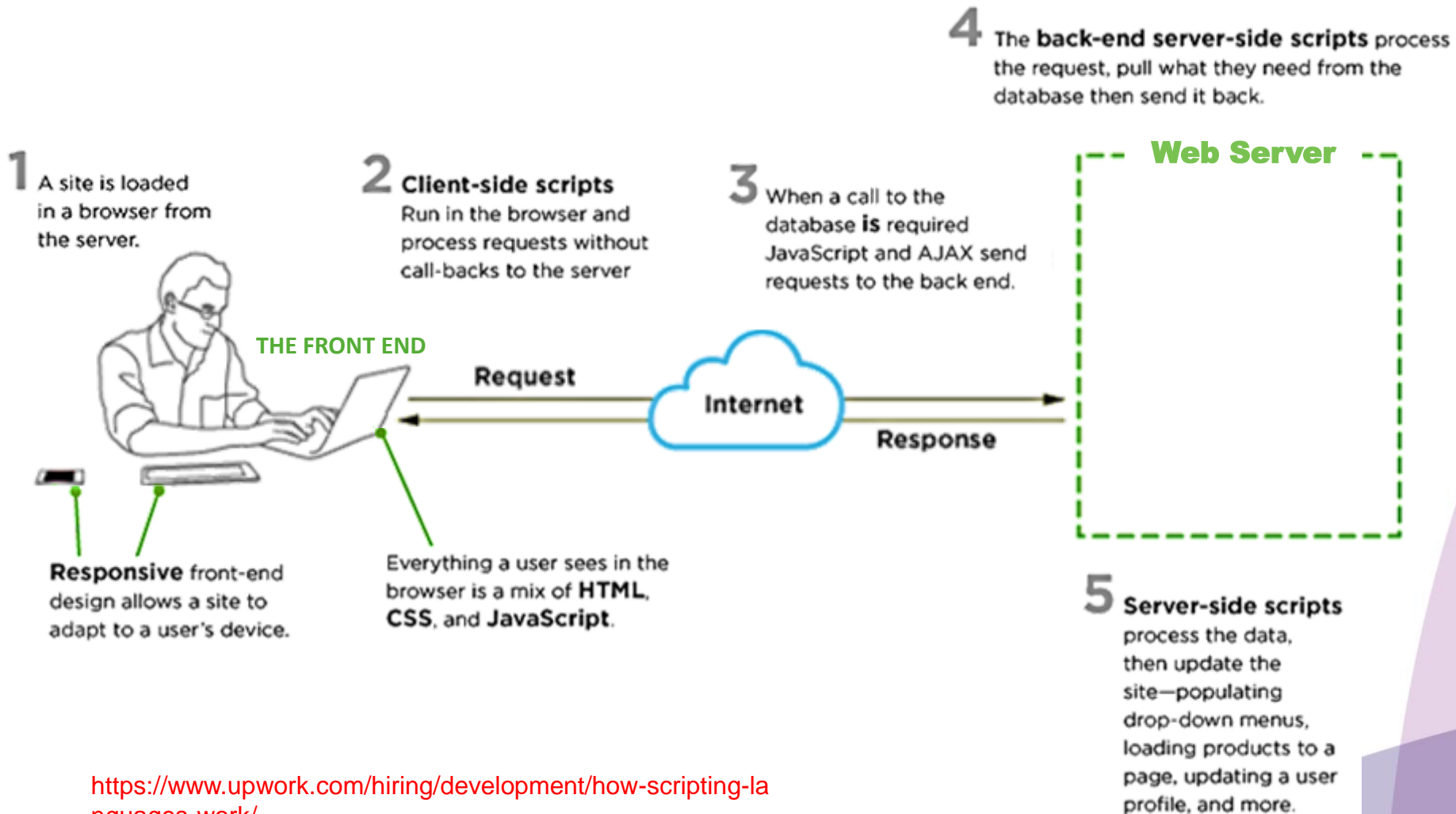
# World Wide Web

- The *World Wide Web* (*Web*) is a network of information resources.
- The **Web** relies on three **mechanisms** to make these resources readily available to the widest possible audience:
  - A uniform naming scheme for locating resources on the Web (e.g., **URLs**).
  - Protocols, for access to named resources over the Web (e.g., **HTTP**).
  - Hypertext, for easy navigation among resources (e.g., **HTML**).



Client-side technologies used to create web sites.

# Client-Server Model

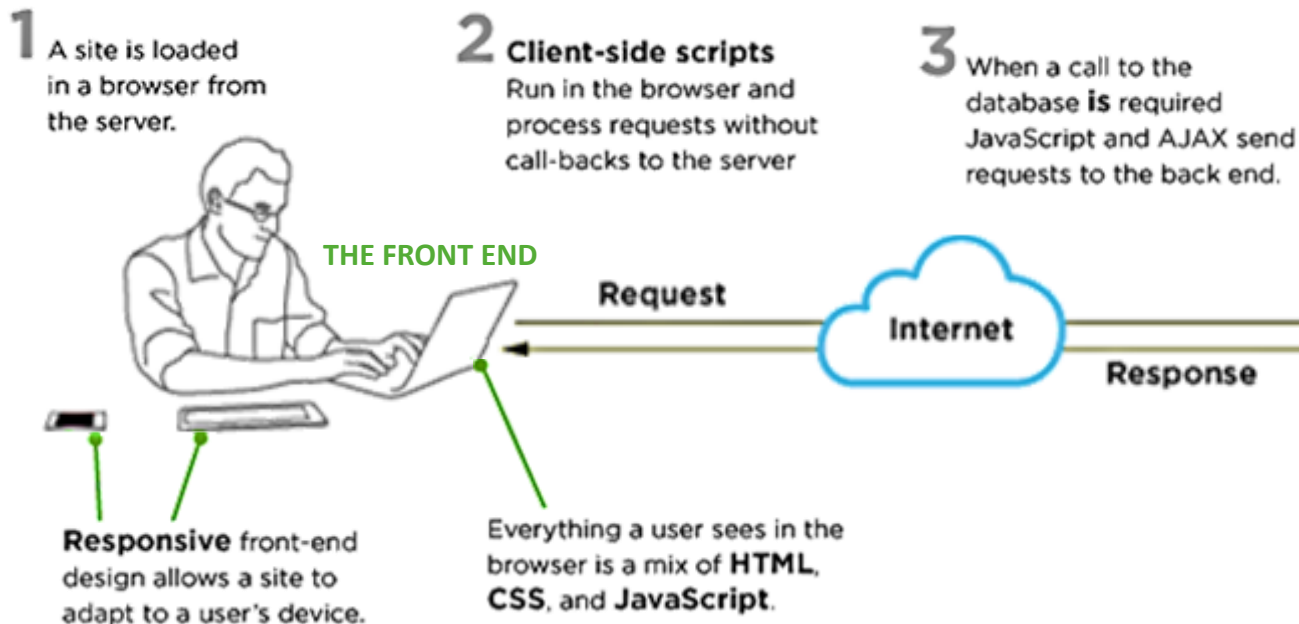


# Client-Server Model

## Full Stack

### Client-side

### Server-side



**4** The **back-end server-side scripts** process the request, pull what they need from the database then send it back.



**5** **Server-side scripts** process the data, then update the site—populating drop-down menus, loading products to a page, updating a user profile, and more.



# Essential Technologies of WWW



HTML

Content &  
Structure.



CSS

Presentation



JavaScript

Behavior

# *HTML*

*The Mother Tongue of The  
Browser*

# HTML Background

---

- HTML stands for “Hyper Text Mark-up Language”.
- The language used to design Web Page.
- HTML was invented in 1990 by a scientist called Tim Berners-Lee. The purpose was to make it easier for scientists at different universities to gain access to each other's research documents.
- HTML standards are organized by W3C :  
<http://www.w3.org/MarkUp/>

# Hyper-Text-Markup-Language

---

- **Hyper** is the method by which you move around on the web.
- **Text** is self-explanatory.
- **Mark-up** is what **HTML tags** do to the text inside them.
- **Language** is what HTML is. It uses many English words.

# HTML

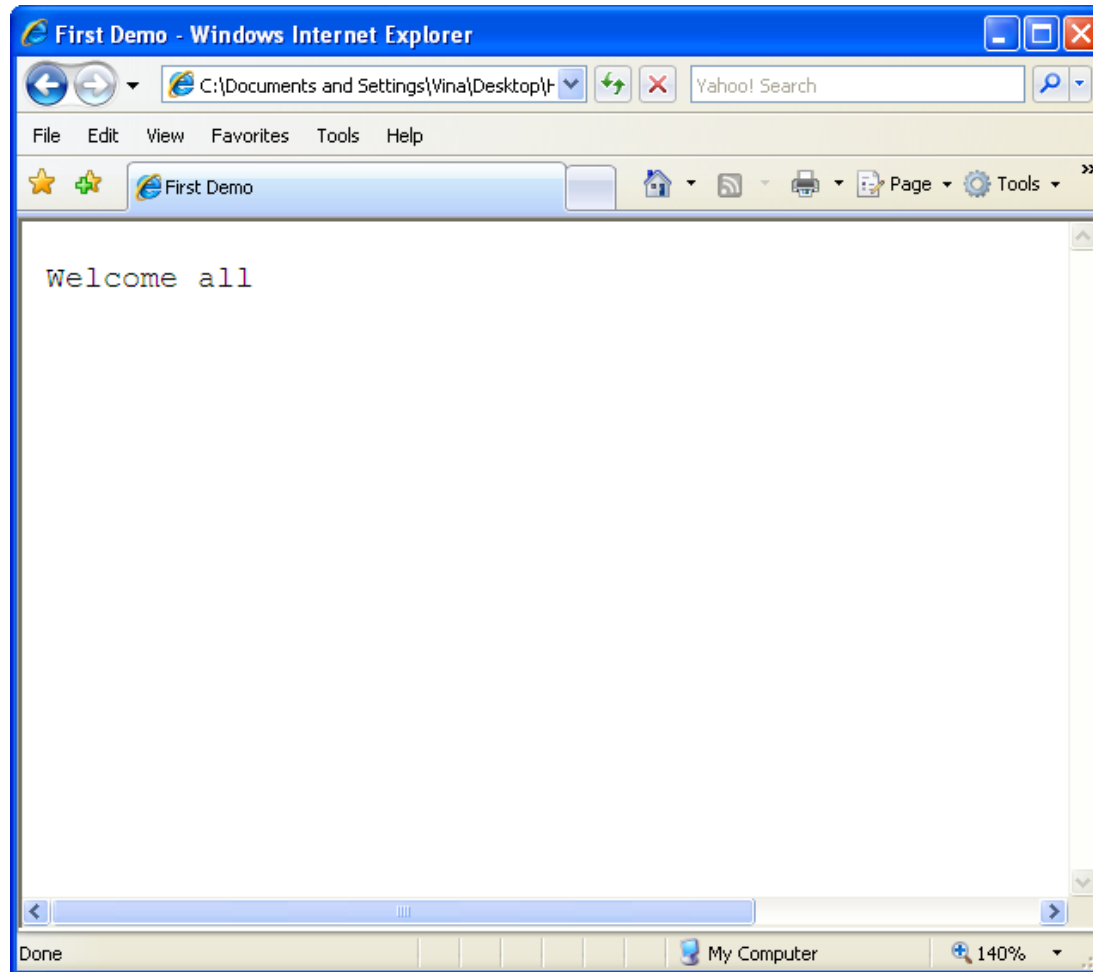
---

- HTML is used for creating **static** web pages.
- It is designed to **display** data & focus on how data looks.
- HTML's role on the web is to tell the browser **how** a document should **appear**.

# HTML Functionalities

- HTML gives authors the means to:
  - Publish online **documents** with headings, text, tables, lists, photos, etc.
    - Include video clips, sound clips, and other applications directly in their documents.
  - Link information via hypertext **links**, at the click of a button.
  - Design **forms** for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.

# Sample Webpage



# Sample Webpage HTML Structure

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title>First Demo</title>
```

```
</head>
```

```
<body>
```

```
<p>Welcome all</p>
```

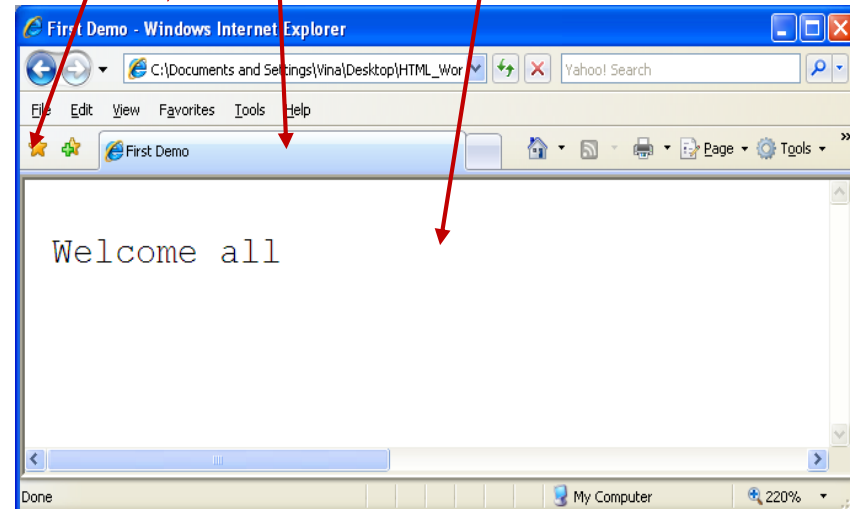
```
<!-- The content of the document -->
```

```
</body>
```

```
</html>
```

**Title tags**

**Body tags**





# HTML Document Basic Structure

- HTML documents contain text and various tags that define elements .
- HTML document contains **<html>** element that wraps
  - **head** section
    - The **title** of the document appears in the head along with other information about the document related to browser & search engine.
  - **body** section
    - The content of the document appears in the body.

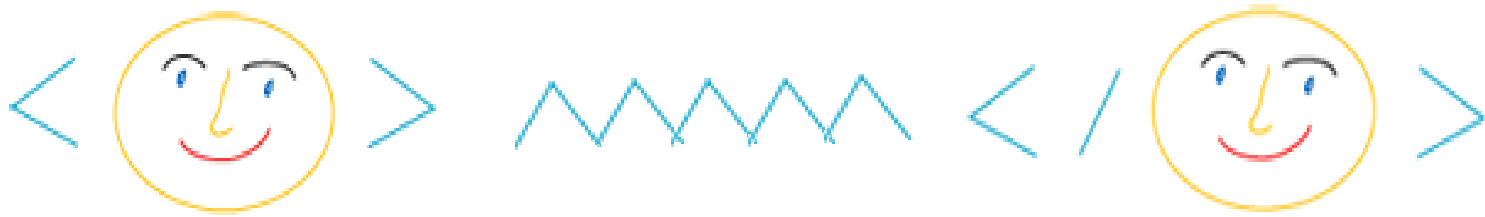


**Example!**

# Sample HTML Tags Example

- `<p>This is a paragraph.</p>`
- `<b>Hello</b> world`
- Hi `<i>Ahmed</i>` Ali
- `< a href="http://www.gamingegypt.com">`  
click here  
`</a>`
- `<b><i>Hello</i></b> world`
- `<hr/>`
- etc...

# HTML Element Syntax



# HTML Element Syntax

```
<start_of_tag attribute_name="attribute value">>  
    Content  
</end_of_tag>
```

- An HTML element consists of an opening tag, a closing tag and the content inside.
- Tags tell the browser how it should display content on screen.
- Tags can have attributes, some tags have obligatory attributes.

# HTML Element Syntax

```
<start_of_tag attribute_name="attribute value">  
    Content  
</end_of_tag>
```

- Attributes provide **additional information** about the element to configure and adjust the behavior of tag.
- Attributes are always specified in **the start tag**.
- Attributes come in name/value pairs like: **name="value"**.

# HTML Element Syntax

- Each element has a number of properties associated with it:
  - starts with a **start tag / opening tag**, begins with a (<) and ends with a (>).
  - ends with an **end tag / closing tag**, begins with a (</) and ends with a (>).
  - The **element content** is everything between the start and the end tag.
  - Some HTML elements have **empty content**.
  - Empty elements are **closed in the start tag**.
  - Most HTML elements can have **attributes**.
  - HTML documents consist of nested HTML elements.
    - Most elements can contain other HTML elements.

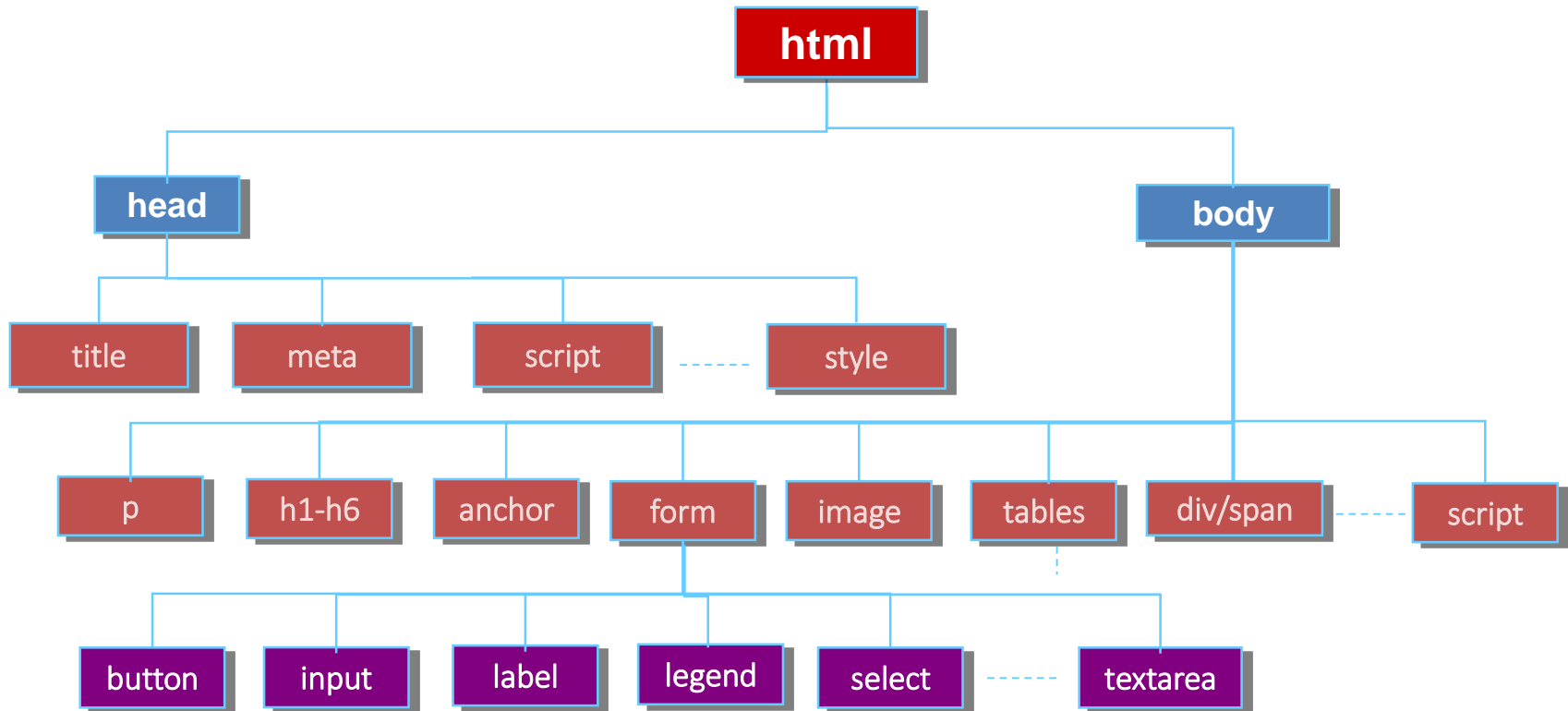
# General Element Attributes

- Core Attributes
  - Used on the majority of HTML elements (although not all)
  - Not valid in base, head, html, meta, script, style, and title elements.

Attribute	Value	Description
class	<i>classname</i>	Specifies a classname for an element
id	<i>id</i>	Specifies a unique id for an element
style	<i>style_definition</i>	Specifies an inline style for an element
title	<i>text</i>	Specifies extra information about an element. It is often displayed as a tooltip or while the element is loading.

# HTML Document Elements Hierarchy

HTML page is like a tree, series of nested elements (tags)





# <head> Element

- 1<sup>st</sup> Child of <html> element
- Many elements may be nested as a child for <head>

Child Tag	Description
<title>	defines the title of the document, its required.
<style>	Defines style information for a document
<script>	Used to define a Client-Side script. Either contains scripting statements or points to an external file
<link>	Defines the relationship between a document and an external resource
<meta>	provides metadata about the HTML document, like page description , keywords, author of the document, last modified
<base>	specifies a default address or a default target for all links on a page

# <title> Tag

- Defines the title of the document
- Shown in Tab
- Used in adding the page to favorite or bookmark list

```
<html>  
  <head>  
    <title>Trial Demo</title>  
  </head>  
</html>
```

# <meta> Tag

- Meta tags are used to store information usually relevant to browsers and search engines.
  - Provides additional information about the page; for example, which character encoding the page uses, a summary of the page's content, instructions to search engines about whether or not to index content, and so on.
  - Define the author of the document as well as the content of the webpage.

```
<meta name="description" content="an html tutorial" />
```

```
<meta name="keywords" content="html, webdesign, javascript" />
```

```
<meta name="author" content="bill gates" />
```

```
<meta http-equiv="refresh" content="5; url=http://www.abc.com" />
```

# <!-- --> Tag

- <!-- --> is the comment tag of html.
- Its used to insert comments in the source code, either as head child or body child.
- Comments are not displayed in the browsers.

- Example:

```
<html>  
  <head>  
    <!--This is a comment in head section-->  
  </head>  
  <body>  
    <!--This is a comment in body section-->  
    <p>This is a paragraph.</p>  
  </body>  
</html>
```

# <body> Element

- Last Child of <html> element
- The <body> element defines the document's body.
- Many elements may be nested as a child for <body>
- Inside <body> Section:
  - Text
    - ▷ Formatting
    - ▷ Resizing
    - ▷ Layout
    - ▷ Listing
  - Images
    - ▷ Inserting images (GIF & jpg)
    - ▷ Adding a link to an image
  - Links
    - ▷ To local pages
    - ▷ To pages at other sites
    - ▷ To bookmarks
  - Forms
  - Tables

# Text Format Appearance

Tag	Description
<code>&lt;b&gt;text&lt;/b&gt;</code>	writes text as bold
<code>&lt;i&gt;text&lt;/i&gt;</code>	writes text in italics
<code>&lt;u&gt;text&lt;/u&gt;</code>	writes underlined text
<code>&lt;em&gt;text&lt;/em&gt;</code>	defines emphasized text
<code>&lt;sub&gt;text&lt;/sub&gt;</code>	lowers text and makes it smaller
<code>&lt;sup&gt;text&lt;/sup&gt;</code>	lifts text and makes it smaller
<code>&lt;del&gt;text&lt;/del&gt;</code>	defines text that has been deleted from a document.
<code>&lt;ins&gt;text&lt;/ins&gt;</code>	defines text that has been inserted into a document.
<code>&lt;strike&gt;text&lt;/strike&gt;</code>	strikes a line through the text
<code>&lt;strong&gt;text&lt;/strong&gt;</code>	usually makes text bold

# Text Size Appearance

Tag	Description
<code>&lt;big&gt;text&lt;/big&gt;</code>	increase the size by one
<code>&lt;small&gt;text&lt;/small&gt;</code>	decrease the size by one
<code>&lt;h1&gt;text&lt;/h1&gt;</code>	writes text in biggest heading
<code>&lt;h6&gt;text&lt;/h6&gt;</code>	writes text in smallest heading

# Text Layout

Tag	Description
<code>&lt;p&gt;text&lt;/p&gt;</code>	Adds a paragraph break after the text.
<code>&lt;p align="<del>left center right</del>"&gt; text &lt;/p&gt;</code>	Directs the alignment of text in paragraph.
<code>&lt;pre&gt;text&lt;/pre&gt;</code>	writes text exactly as it is, including spaces.
<code>&lt;div&gt; text&lt;/div&gt;</code>	Defines a section in a document
<code>&lt;span&gt;text&lt;/span&gt;</code>	

Example!



# Block vs. Inline Elements

- Block elements
  - Container elements for grouping other elements.
  - May contain other block elements & inline elements.
  - Normally start (and end) with a new line when displayed in a browser.  
e.g. <div>, <p>, <h1>..

######
- Inline elements
  - Container for text and other inline elements.
  - Normally displayed without starting a new line.  
e.g. <span>, <b>, <td>, <a>, <i> etc.

# <div> vs. <span>

- **<div>** Defines a section in a document (block-level)
  - creates logical divisions within a page
- **<span>** Defines a section in a document (inline)
  - Useful for modifying a specific portion of text
- HTML elements can be grouped together with **<div>** and **<span>**.
- Useful with CSS

**Example!**

# Text breaking and white space

- Whitespace generally ignored in block and inline
- `<pre>`
  - whitespace is respected
- `<br />`
  - Explicit line break
- `<hr />`
  - Horizontal rule
- Use Character entities
  - entities for HTML markup characters.

# Special Character Entities

---

- Entities are used to implement reserved characters or to express characters that cannot easily be entered with the keyboard.
- Syntax:  
    &entity\_name or &#entity\_num

# Special Character Entities

Name	Symbol	HTML Equivalent
Ampersand	&	&amp; or &#38;
cent sign	¢	&cent; or &#162;
copyright symbol	©	&copy; or &#169;
degree sign	°	&deg; or &#176;
greater than	>	&gt; or &#62;
less than	<	&lt; or &#60;
non-breaking space		&nbsp; or &#160;
registered trademark	®	&reg; or &#174;

<https://dev.w3.org/html5/html-author/charref>

# Special Character Entities

Name	Symbol	HTML Equivalent
trademark	™	&trade; or &#153;
quotation mark	“	&quot; or &#34;
apostrophe	‘	&apos; or &#39;
Euro	€	&euro; or &#8364;
British Pound	£	&pound; or &#163;
Japanese Yen	¥	&yen; or &#165;
Cent sign	¢	&cent; or &#162

# HTML Lists

---

- HTML supports
  - ordered “Numbered” lists,
  - unordered “Bulleted” lists, &
  - description “Definition” lists.

# Numbered List

Attribute	Value	Description
Start	Number (default) Capital letter	Use styles instead. Specifies the start point in a list
Type	Small letter Capital Roman # Small Roman #	Use styles instead. Specifies which kind of bullet points will be used

- An ordered list starts with the `<ol>` tag
- `<li>` tag to define list items.

```
<ol start="5">  
  <li>text</li>  
  <li>text</li>  
</ol>
```

```
5. text  
6. text  
7. text  
8. text  
9. text
```

```
<ol type="A" reversed>  
  <li>text</li>  
  <li>text</li>  
</ol>
```

```
E. text  
D. text  
C. text  
B. text  
A. text
```



# Bulleted Lists

```
<ul type="circle">  
  <li>text</li>  
  <li>text</li>  
</ul>
```

```
<ul type="disc">  
  <ul type="circle">  
    <ul type="square">
```

- An unordered list starts with the **<ul>** tag.
- **<li>** tag to define list items.
- You have the following bullet options as a value for *type* attribute:
  - **disc** (default)
  - **circle**
  - **square**

Example!

# Definition List

- An definition list starts with the `<dl>` tag.
- `<dt>` definition term tag present the item in the list to be defined.
- `<dd>` definition description tag is used to describe an item in a definition list.

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>

  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

**Example!**

# HTML Links

```
<a href="url" target="">Link text</a>
```

Click [here](http://www.yahoo.com) to go to yahoo.

- Image link content

```
<a href="myfile.htm"></a>
```

- Link Within a Page

- To link to an anchor you need to:

- Create a link pointing to the anchor
    - Create the anchor itself.

```
< tag id|name="top"></tag>
```

```
<a href="#top">Top</a>
```

- Link to email

```
<a href="mailto:email@hotmail.com?subject=SweetWords  
&body=Please send me a copy of your new program!">  
Email Me  
</a>
```

# HTML Images

```
<img src="" width="" height="" alt="" />
```

- Images commonly types used in browsers are : GIFs, JPEGs, & PNGs

```

```

```

```

- Alternative Text

```

```

- Resizing

```

```

# Image Map

---

- Image maps are images, that have been divided into regions.
- Clicking in a region of the image cause the web surfer to be connected to a new URL.
- Image maps are graphical form of creating links between pages

# Image Map

- Possible shapes for areas inside image are

```
<img src="" usemap= "#example" />
```

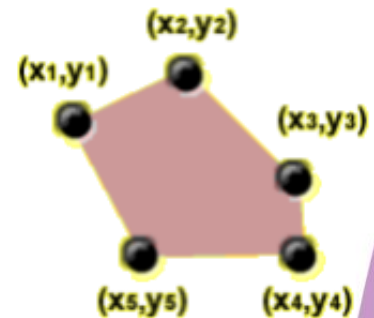
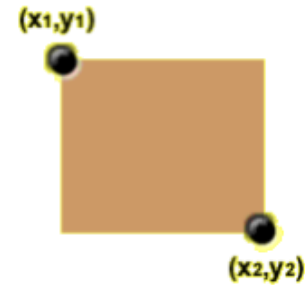
```
<map name= "example">  
  <area shape=rect coords= "x1,y1,x2,y2"  
    href="http://www.abc.com" />
```

```
  <area shape=circle coords= "x1,y1,x2,y2"  
    href="http://www.abc.com" />
```

```
  <area shape=polygon coords= "x1,y1,x2,y2,... xn,yn"  
    href="http://www.abc.com" />
```

.....

```
</map>
```



# *Cascading Style Sheets*

*the sister technology to HTML  
that is used to style your web pages*



Designed to separate presentation from content



# CSS

- CSS stands for Cascading Style Sheets.
- CSS was developed by the W3C.
- CSS is a style sheet language used to describe the **presentation** of a document written in a markup language.
- Its most common application is to style web pages written in HTML, XHTML and **any** kind of XML document.
- Styles define **how to display** HTML elements (font face, size, color, alignment, ...etc)
- Styles are normally stored in *Style Sheets*
- The term **cascading** derives from the fact that multiple style sheets can be applied to the same Web page.
- Due to CSS, all HTML presentation **tags** and attributes are **deprecated**, e.g. **font**, **center**, etc

# CSS Benefits

---

- With CSS we have the following benefits:
  1. The Separation of Structure and Presentation
  2. Managing Style at Large Sites
    - Easy maintenance.
  3. Improved performance
    - Page load faster.
  4. Decreased production work
    - Saves time.
  5. Rich design and layout

# CSS Features

---

- Provides precise control over margins, line spacing, element placement, colors, font faces, and font sizes.
- Removes the need to re-type HTML style tags each time a new style is needed.
- Ensures every user sees the same view regardless of the ways in which the browser's size and colors are configured.
- Provides the ability to change the overall look of a Web page or even an entire site by changing a single style sheet.

# CSS Versions

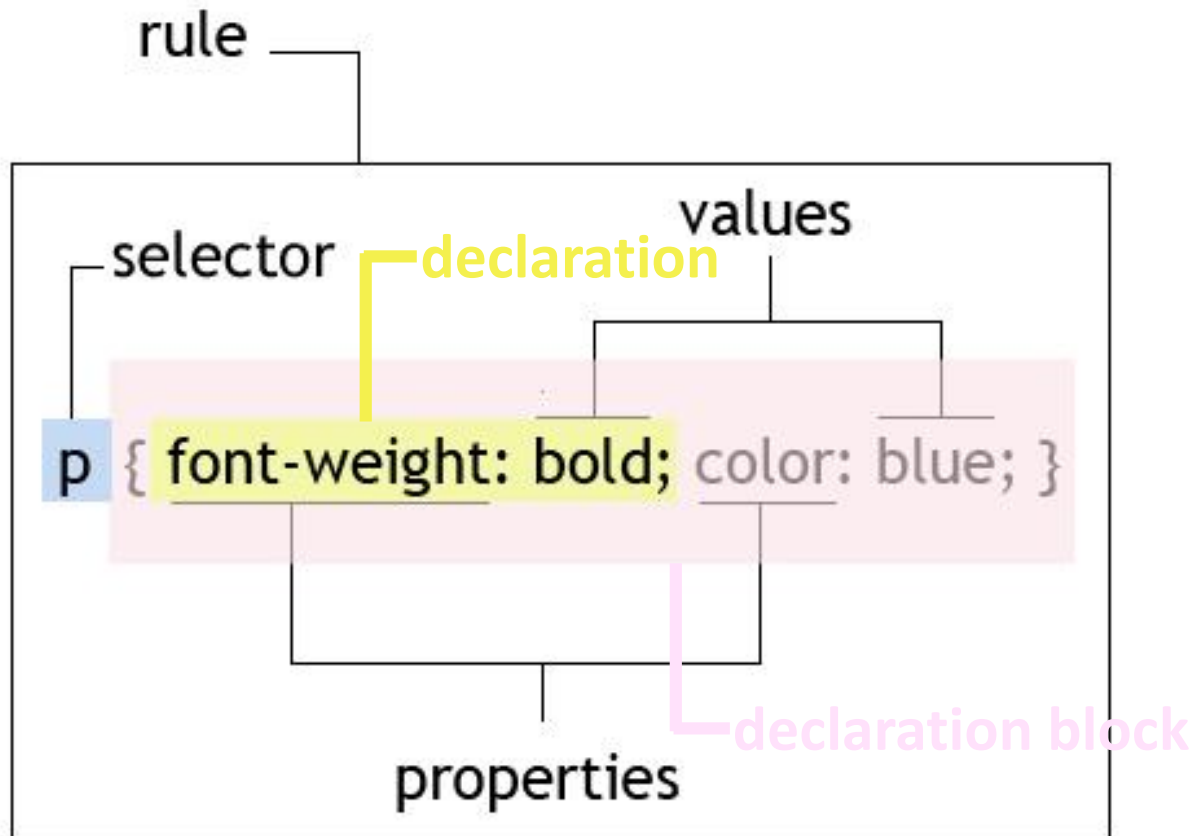
---

- Cascading Style Sheets 1 (CSS1)
  - Features: Fonts, Colors, Alignment, Spacing
- Cascading Style Sheets 2 (CSS2-CSS2.1)
  - Features: Layout, Positioning... (CSS-P)
- Cascading Style Sheets 3 (CSS 3)
  - Features: Effect, Sizing...

# CSS Syntax

- A style sheet consists of the style rules that tell your browser **how** to present a document.
- The CSS syntax rule is made up of 5 parts:
  - 1.selector
  - 2.property
  - 3.value
  - 4. declaration block
  - 5. declaration
- ***selector*** is a pattern to be affected; separated by commas.
- ***property*** and ***value*** describe the appearance of that pattern; separated by colons; building a ***declaration***.
- ***declarations*** are property-value pair; separated by semicolons; building a ***declaration block***.
- Style rules are formed as follows:  
***selector {property: value}***

# CSS Rule



# Implementing CSS

- CSS can be linked to an HTML document as:
  1. Embedding in <head> section using `<style>`
  2. Linking to an external style sheet file using
    - `<link>` element within head section
    - `@import` rule within style tag in the head section
  3. Inline style using `style` attribute
- Using external files is highly recommended

# 1. Embedding in a Style Tag

- Embedded, or **internal** styles are used for the whole page.
- You define internal styles in the **head** section by using the `<style>` tag
- An embedded (internal) style sheet should be used when a single document has a unique style.

**<head>**

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

**h1 { color: blue; }**

**h2 { color: red; }**

**</ style >**

**</head>**

**H1 header with blue color**

**H2 header with red color**

**Example!**



## 2. Linking to an External Style Sheet File

SoC

- An external style sheet is ideal when the style is applied to many pages.
- With an external style sheet, you can change the look of an entire Web site by changing one file.
- Using `<link>` tag.
  - Basically links an external style sheet to the document.
  - The `<link>` tag goes inside the `head` section.

`<head>`

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

`</head>`

Example!

## 2. Linking to an External Style Sheet File

- Using **@import** rule
  - Another way to link external CSS files
  - Basically imports one style sheet into another.
  - Placed at the top of the <style> or in external style sheets.
  - Must come before any other declaration

```
<style type="text/css">  
    @import url("styles1.css");  
    /*same as*/  
    @import "style1.css";  
  
    p {color: yellow }  
</style>
```

### 3. In-line Style

- **In-line** styles are plunked straight into the HTML tags using the **style** attribute.
- **In-line** style loses many of the advantages of style sheets by mixing content with presentation.
- **In-line** style should be avoided wherever possible

- **Example:**

```
<p style="color: red; font-family: 'Arial' ">
```

This paragraph is styled in red with the Arial font, if available.

```
</p>
```

**Example!**

# CSS Comments

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

```
/*
```

```
h1 { color: red; font-family: "Calibri";}
```

```
*/
```

```
</style>
```

# Cascading Order

- “**Cascading**” reflects the way styles are applied to the elements in a document, because style declarations cascade down to elements from many origins.
- Styles will be applied to HTML in the following order:
  1. Browser default
  2. External style sheet
  3. Internal style sheet (in head)
  4. Inline style
- When styles conflict, the “nearest” (most recently applied) style wins.

# Grouping

DRY

- Grouping selectors is done by separating each selector with a comma to give the same properties to a number of selectors without having to repeat

**h1,h2,h3,h4,h5,h6 { color: green; font-family: "Arial" }**

## Example:

```
h1 { font-family: "sans-serif" }  
h2 { font-family: "sans-serif" }  
h3 { font-family: "sans-serif" }
```

**is equivalent to:**

```
h1, h2, h3 { font-family: "sans-serif" }
```

# Example of Cascading Order

- External Style sheet

```
h3 { color: red;
      text-align: left;
      font-size: 8pt }
```

- Internal Style sheet

```
h3 { text-align: right;
      font-size: 20pt;
      text-decoration: underline
    }
```

- Resultant attributes

```
color: red;
text-align: right;
font-size: 20pt;
text-decoration: underline
```

Example!

# CSS Selectors

- Selectors determine which element the rule applies to:
  - All elements of specific type (tag)
  - Those that match a specific attribute (id, class)
  - Elements may be matched depending on how they are nested in the document tree (HTML)
  - Examples:
    - `.header{ color: green }`
    - `#menu{ padding-top: 8px }`



# CSS Selectors

- Several types of selectors are defined for use when implementing Style Sheets:
  1. Simple Basic Selectors
  2. Attribute selectors
  3. Combinators
  4. Pseudo-Classes
  5. Pseudo-Elements
- A selector can contain a chain of one or more simple selectors separated by combinators, optionally followed by attribute selectors, ID selectors, or pseudo-classes. but it can contain only one pseudo-element, which must be appended to the last simple selector in the chain

# 1. Simple Basic Selectors

---

1. Type Selector
2. IDs
3. Classes
4. Universal Selector

# 1.1 Type Selector

- In general, **STYLE** attribute can be added to any HTML element.
- Example:  

```
<span style = ' font-family: "sans serif "; color: blue; text-align: center  
'>  
  Hello There!  
</span>
```
- Type selector selects an element of the HTML document: P, H1, BODY, etc.
- Example:  

```
h1 {color: blue;}
```

# 1.2 ID Selector

- The ID attribute is used to define a unique style for an element.

- Example:

- ✓ In the CSS

- ```
#id1 {color: red}
```

- ✓ In the HTML

- ```
<div id="id1">
```

- ```
    This is the div with the id.  
</div>
```

# 1.2 ID Selector

- Example2:

- ✓ In the CSS

- `div#id1 {color: red}`

- ✓ In the HTML

- `<div id="id1">`

- `This is the div with the id.`

- `</div>`

Example!

# 1.3 Classes Selector

- Classes allow you to define a style which can be applied to multiple elements on your page.
- Example 1:
  - To apply one class over more than one different HTML element:
    - ✓ In the CSS  
**.bold { font-weight: bold }**
    - ✓ In the HTML  
**<p class="bold">**  
**This paragraph will be Bold.</p>**  
**<span class="bold">**  
**This SPAN will be Bold too.</span>**
- Both the paragraph & the span elements will be styled by the class "bold".

# 1.3 Classes Selector

- Example 2:

- To apply more than one class per given element:

- ✓ In the CSS

- .bold { font-weight: bold }**

- .large { font-size: 20pt }**

- ✓ In the HTML

- <p class="bold large">**

- This paragraph will be Bold & very large.**

- </p>**

- The paragraph above will be styled by the class "bold" AND the class "large".

# 1.3 Classes Selector

- Example 3:
  - Say that you would like to have two types of paragraphs in your document: one right-aligned paragraph, and one center-aligned paragraph. Here is how you can do it with styles:

✓ In the CSS

**p.right {text-align: right}**

**p.center {text-align: center}**

✓ In the HTML

**<p class="right"> This paragraph will be right-aligned.</p>**

**<p class="center">**

**This paragraph will be center-aligned.**

**</p>**

**Example!**





# *Assignments*