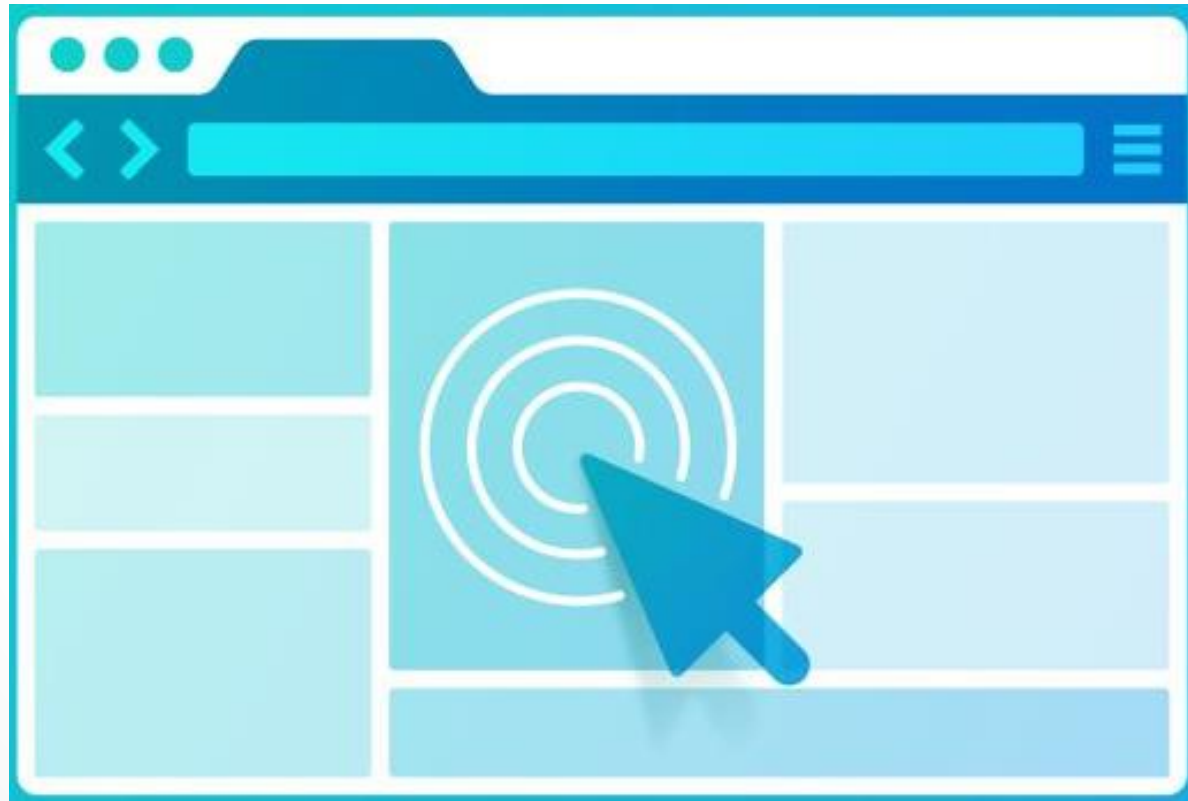


Web Design

HTML

Amal Elnuri

Websites



What is the Website?

- A **website** or **web site** is a collection of related network web resources , such as web pages, multimedia content.
- which are typically identified with a common **domain name**, and published on at least one **web server**.
- Websites can be accessed via a public (**IP**).
- Uniform Resource Locator (**URL**) that identifies the site or web address.

What is the Website?

- Web pages, which are the building blocks of websites, are documents, typically composed in plain text interspersed with formatting instructions of Hypertext Markup Language (**HTML**).
- Web pages are accessed and transported with the Hypertext Transfer Protocol (**HTTP**).
- End users can access websites on a range of devices, including desktop and laptop computers, tablet computers, smartphones and smart TVs.

STATIC WEBSITE

Quick to develop

Cheaper to develop

Easier to host

Ideal for small websites

DYNAMIC WEBSITE

Restructuring is simple

Multiple user access

Ensure longevity

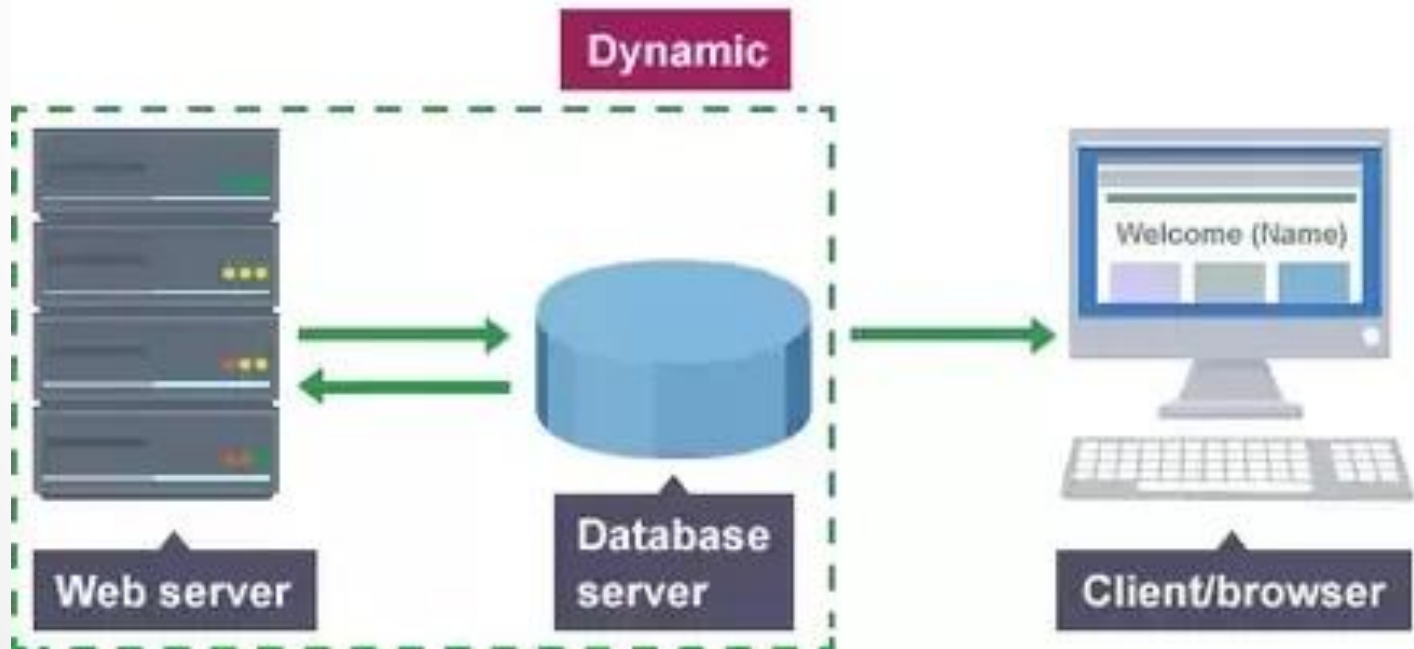
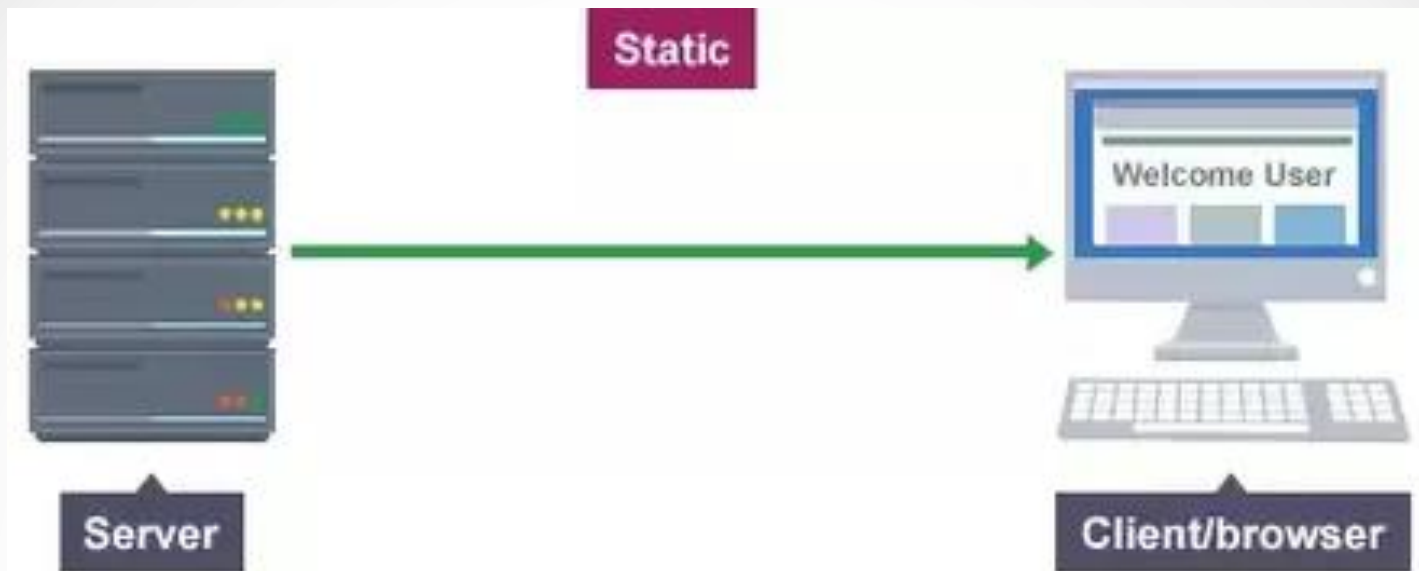


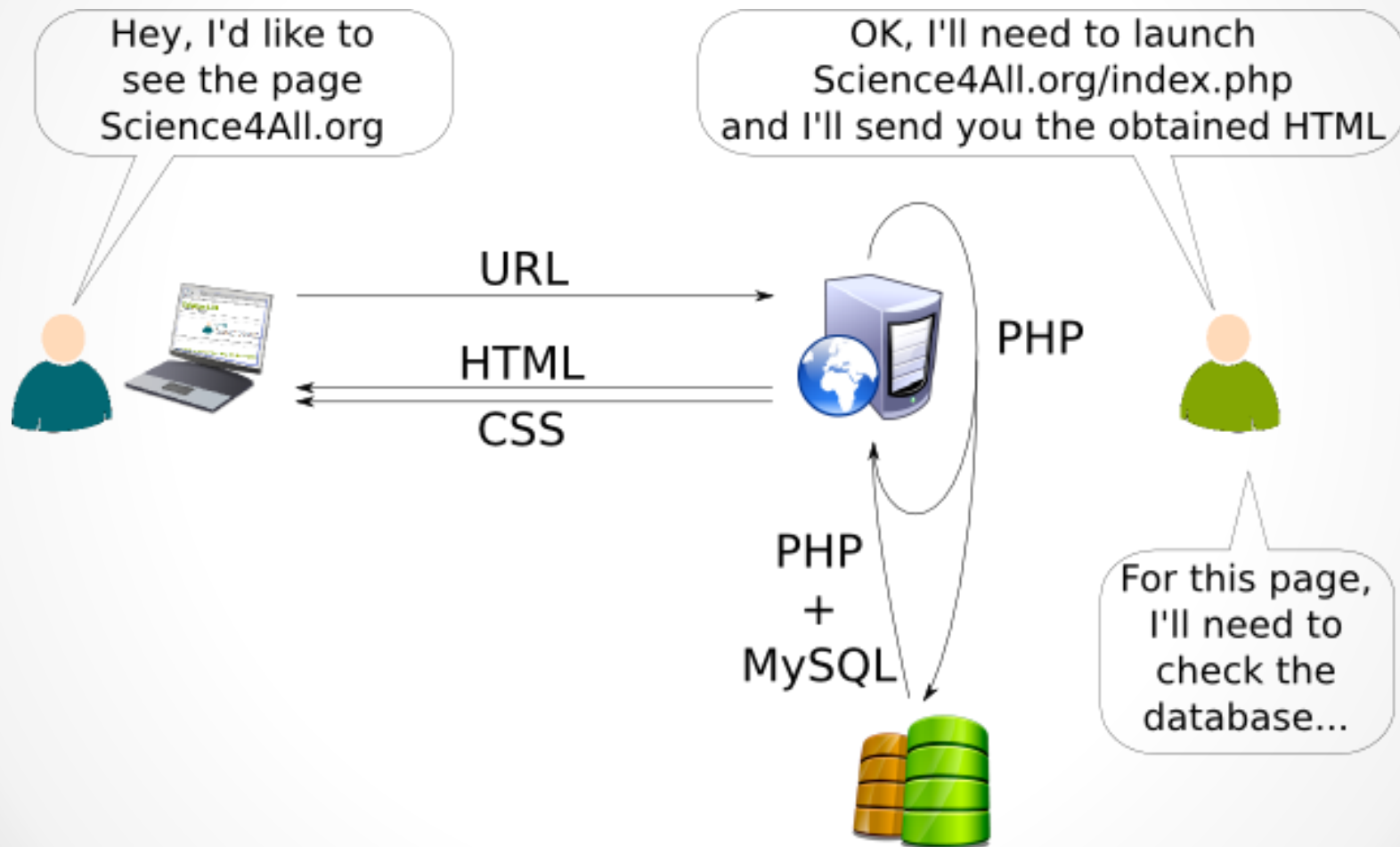
Static website






- A static site or a set of related static pages, create a page as a file and save it to your computer or server.
- Every time there is a need to change the HTML file needs to be reopened, edited and saved. In this case do not use any PHP development language.

Dynamic website

- Ability to display the content of the page in a different way each time it is displayed.
- Possibility to include a site on a script by Client or a program by Server until a variable content is obtained. Or a combination of both.
- Content that is created on a user's computer instead of in a server, uses Client-based scripting, such as .JavaScript
- Content that is created on Server and sent to a Client computer, uses server-side scripting, such as:
 - PHP
 - ASP





Language	Role	Where it runs
HTML	Content and structure (text, images , tables , forms , etc)	 Browser
CSS	Style and Presentation (color , fonts , background , margins , padding , etc)	 Browser
JavaScript	Client side scripting (dynamic user interaction , popups , forms)	 Browser
PHP	Server Side Scripting (server side logic and data processing)	 Server
MySQL	Data Management (stores all sorts of data needed by a website)	 Server

Types of Websites

- E-commerce. ...
- Blogs and Personal. ...
- Informational. ...
- Online Community. ...
- Photo Sharing. ...
- Business Directory.....
- News....
- Educational.....



HTML

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content

HTML

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content

HTML

- 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 Is Not Case Sensitive
- HTML tags are not case sensitive: <P> means the same as <p>.

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

Write and Read

Write by:

- Notepad or TextEdit
- Dreamweaver
- JetBrains WebStorm
- Visual Studio
- **Visual Studio Code**
- NetBeans

Reading by:

- Web Browser.
- Internet Explorer.
- Firefox.
- **Google Chrome.**

HTML Structure

`<html>`

`<head>`

(title, meta, Links, scripts)

`</head>`

`<body>`

(paragraphs, tables, images, videos, etc.).

`</body>`

`</html>`

HTML Example

```
<!DOCTYPE html>
```

```
<!--
```

To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.

```
-->
```

```
2    <html>
```

```
3    <head>
```

```
4    |    <title>Page Title</title>
```

```
5    </head>
```

```
6    <body>
```

```
7    |
```

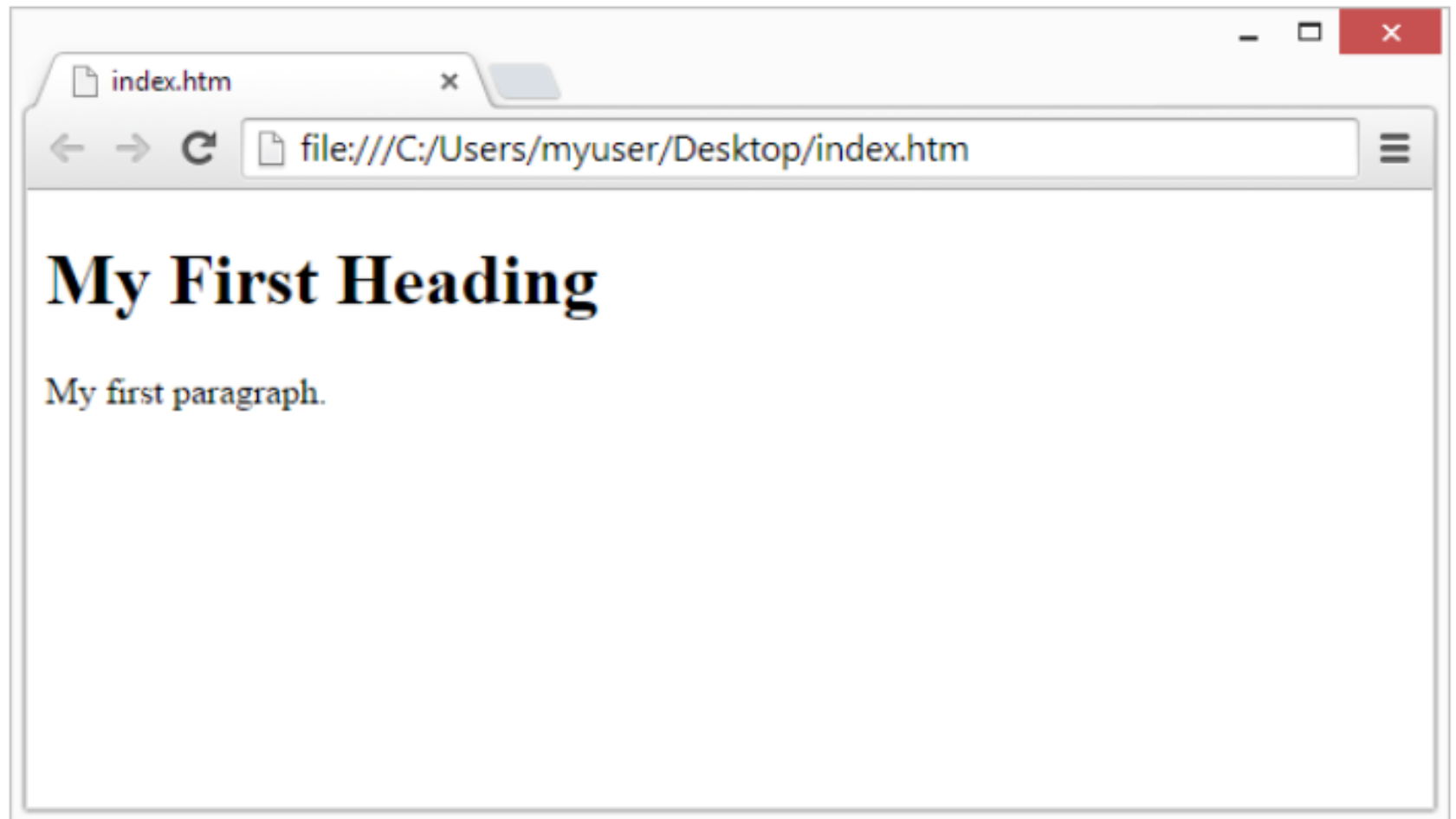
```
8    |    <h1>My First Heading</h1>
```

```
9    <p>My first paragraph.</p>
```

```
10
```

```
11    </body>
```

```
12    </html>
```



HTML Tags

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

- The **<!DOCTYPE html>** or ***Document Type*** 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

Comments

- Insert comments in your HTML5 markup to improve readability and describe the content of a document.
- The browser ignores comments when your document is rendered.
- Comments start with `<!--` and end with `-->`.

HTML Elements

- An HTML element usually consists of a **start** tag and an **end** tag, with the content inserted in between:

<tagname>Content goes here...</tagname>

Empty HTML Elements

- HTML elements with no content are called empty elements.
- `
` is an empty element without a closing tag (the `
` tag defines a line break):

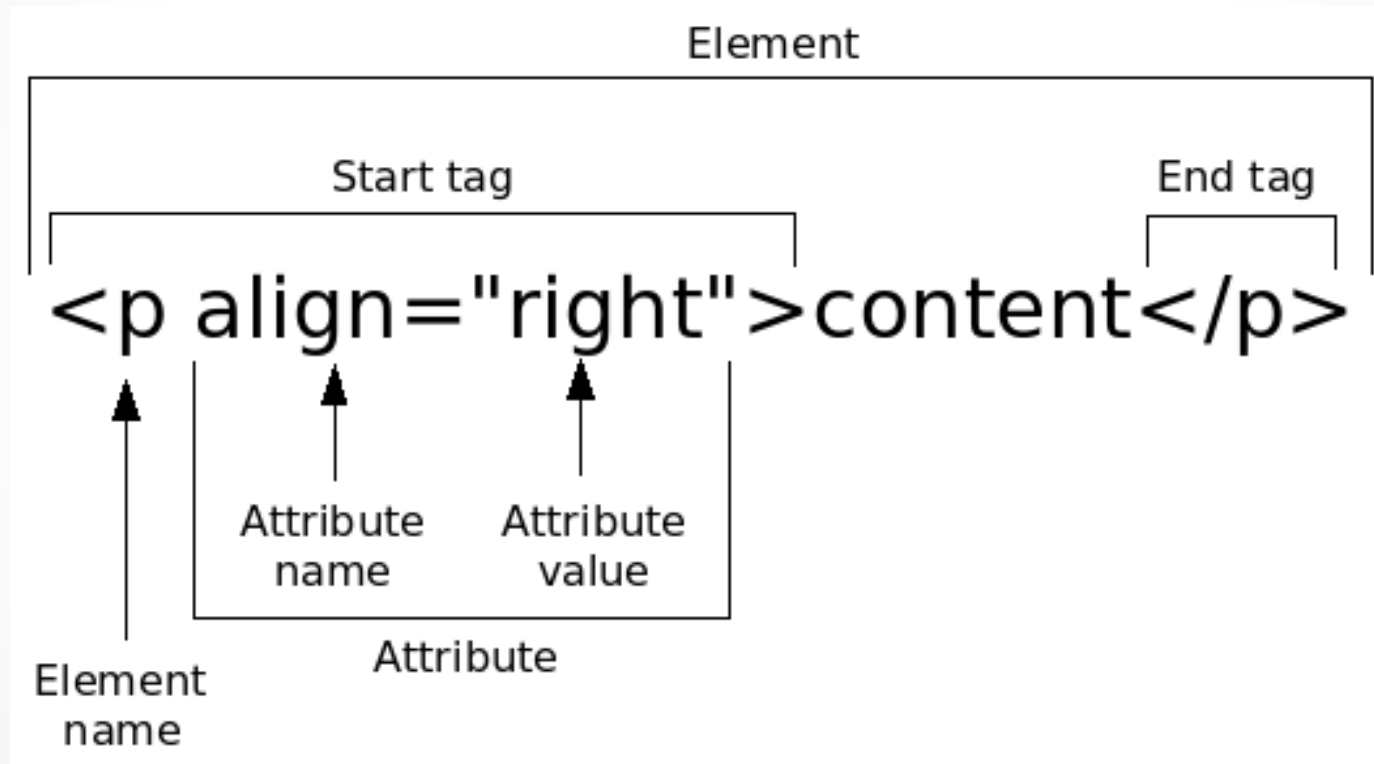
HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

Example

- ``

Tag/Elements/Attributes



To change this template file, choose Tools | Templates and open the template in the editor.

-->

```
<html>
```

```
  <head>
```

```
    <title>TODO supply a title</title>
```

```
    <meta charset="UTF-8">
```

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

```
  </head>
```

```
  <body>
```

```
    <h1>Heading 1</h1>
```

```
    <h2>Heading 2</h2>
```

```
    <h3>Heading 3</h3>
```

```
    <h4>Heading 4</h4>
```

```
    <h5>Heading 5</h5>
```

```
    <h6>Heading 6</h6>
```

```
  </body>
```

```
</html>
```

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Meta

- Metadata is data (information) about data.
- The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parable.
- Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.
- The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

viewport

- **Example 1 - Define keywords for search engines:**

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

- **Example 2 - Define a description of your web page:**

`<meta name="description" content="Free Web tutorials on HTML and CSS">`

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

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

- **Example 4 - Refresh document every 30 seconds:**

`<meta http-equiv="refresh" content="30">`

- **Example 5 - Setting the viewport to make your website look good on all devices:**

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

- **Example 6 - Specify the character encoding for the HTML document:**

`<meta charset="UTF-8">`

●●●● N Telenor 12.29 77 %



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duiis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet domine

●●●● N Telenor 12.29 77 %



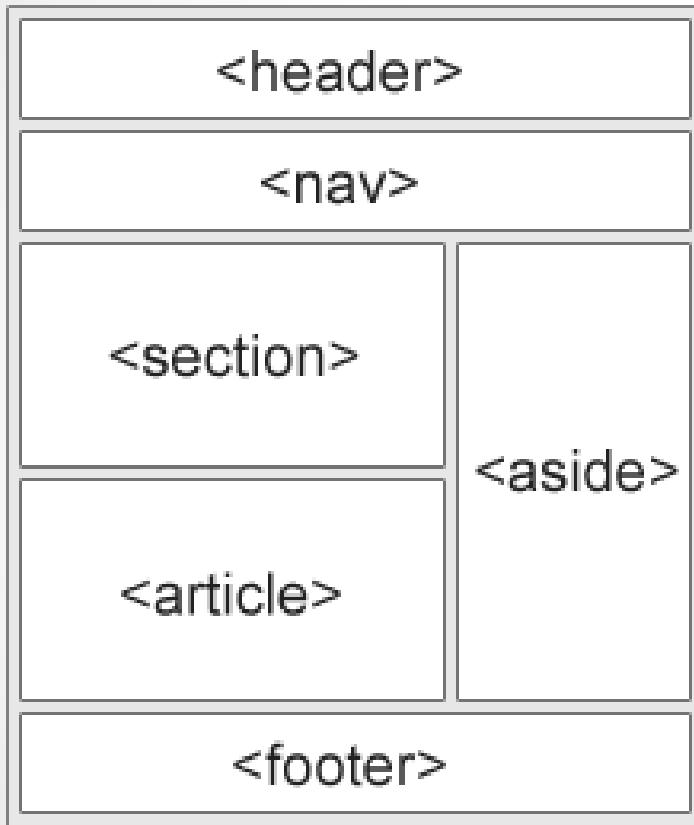
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duiis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet domine

Style

Use the style attribute for styling HTML elements:

- Use background-color for background color.
- Use color for text colors.
- Use font-family for text fonts
- Use font-size for text sizes.
- Use text-align for text alignment.

Layout



- **<header>** - Defines a header for a document or a section
- **<nav>** - Defines a container for navigation links
- **<section>** - Defines a section in a document
- **<article>** - Defines an independent self-contained article
- **<aside>** - Defines content aside from the content (like a sidebar)
- **<footer>** - Defines a footer for a document or a section

HTML Layout Techniques

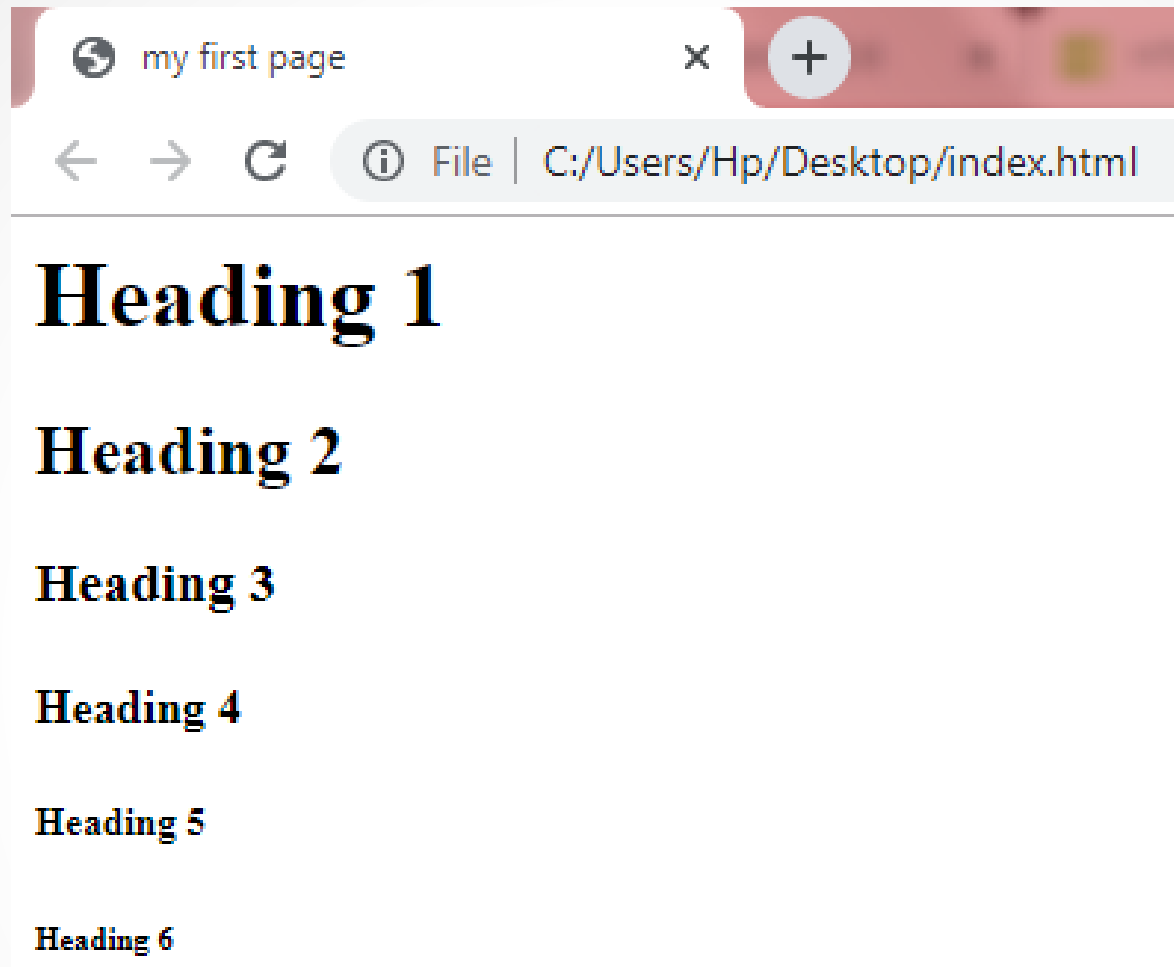
There are five different ways to create multicolumn layouts. Each way has its pros and cons:

- HTML tables (not recommended)
- CSS framework
- CSS float property
- CSS flexbox
- CSS grid

HTML Headings

- Search engines use the headings to index the structure and content of your web pages.
- **<h1>** headings should be used for main headings, followed by **<h2>** headings, then the less important **<h3>**, and so on.
- Each HTML heading has a default size. However, you can specify the size for any heading with the **style attribute**, using the CSS **font-size** property:
- **Example**
`<h1 style="font-size:60px;">Heading 1</h1`

```
1  <html>
2  |   <head>
3  <title>
4  |   my first page
5  </title>
6  |   </head>
7  |   <body>
8  |       <h1>Heading 1</h1>
9  |       <h2>Heading 2</h2>
10 |       <h3>Heading 3</h3>
11 |       <h4>Heading 4</h4>
12 |       <h5>Heading 5</h5>
13 |       <h6>Heading 6</h6>
14 |   </body>
15 </html>
16
```

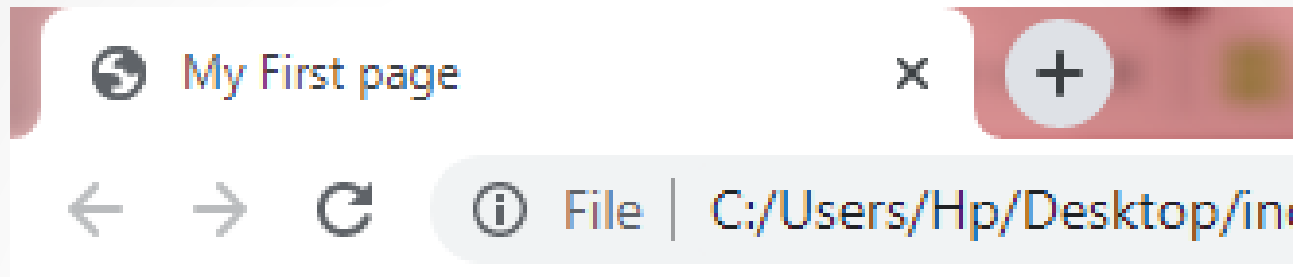


- **Note:** Browsers automatically add some white space (a margin) before and after a heading.

HTML Paragraphs

- The HTML `<p>` element defines a **paragraph**.
- The browser will remove any extra spaces and extra lines when the page is displayed.
- Don't Forget the End Tag

```
1  <html>
2  |   <head>
3  <title>
4  |   my first page
5  </title>
6  |   </head>
7  |   <body>
8  |       <p>This is a paragraph 1.</p>
9  |       <p>This is a paragraph 2.</p>
10 |       <p>This is a paragraph 3.</p>
11 |
12 |   </body>
13 </html>
14
```



This is a paragraph.

This is a paragraph.

This is a paragraph.

HTML Line Breaks

- Use **
** if you want a **line break** (a new line) without starting a new paragraph, **example:**
`<p>This is
a paragraph
with line breaks.</p>`

The HTML **<pre>** Element

- The HTML **<pre>** element defines preformatted text.
- The text inside a **<pre>** element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks.

HTML Horizontal Rules

The **<hr>** tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

```
8      <body>
9
10     <p>This is a paragraph 1.</p>
11     <p>This is a paragraph 2.</p>
12     <p>This is a paragraph 3.</p>
13     <hr>
14     <p> This is a paragraph 1.
15     |   This is a paragraph 2.
16     |   This is a paragraph 3.
17     </p>
18     <hr>
19     <p>This is<br>a paragraph<br>with line breaks</p>
20     <hr>
21     <pre>
22     |   This is a paragraph 1.
23     |   This is a paragraph 2.
24     |   This is a paragraph 3.
25     </pre>
26     </body>
```




This is a paragraph 1.

This is a paragraph 2.

This is a paragraph 3.

This is a paragraph 1. This is a paragraph 2. This is a paragraph 3.

This is
a paragraph
with line breaks

```
This is a paragraph 1.  
This is a paragraph 2.  
This is a paragraph 3.
```

HTML Styles

- Setting the style of an HTML element, can be done with the style attribute.

- **syntax:**

<tagname style="property:value;">

Some of properties:

- Use **background-color** for background color
- Use **color** for text colors
- Use **font-family** for text fonts
- Use **font-size** for text sizes
- Use **text-align** for text alignment

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>
5       HTML
6     </title>
7   </head>
8
9   <body style="background-color: beige;">
10    <h1 style="color: brown;">HTML Style</h1>
11    <br>
12    <p style="font-family:sans-serif;">We can use some of attributes in HTML page not just in CSS page</p>
13    <br>
14    <p style="font-size:160%; color: crimson;"> but by the same syntax of the CSS</p>
15    <hr>
16    <p style="text-align: center; font-style: italic;"> You will learn more about CSS later </p>
17    |
18  </body>
19 </html>
20
```

HTML Style

We can use some of attributes in HTML page not just in CSS page

but by the same syntax of the CSS

You will learn more about CSS later

HTML Text Formatting

- HTML defines special **elements** for defining text.
- **** - Bold text
- **** - Important text
- **<i>** - Italic text
- **** - Emphasized text
- **<mark>** - Marked text
- **<small>** - Small text
- **** - Deleted text
- **<ins>** - Inserted text
- **<sub>** - Subscript text
- **<sup>** - Superscript text

```
<body style="background-color: beige;">
<h1 style="color: brown;">HTML <small> Text</small> Formatting</h1>
<br>
<p>We can use some of<b> attributes</b> in HTML page not just in <em> CSS page</em></p>
<br>
<p style="font-size:160%; color: crimson;"> but by the same <mark> syntax</mark> of the CSS</p>
<hr>
<p style="text-align: center; font-style: italic;"> You will learn more about<del> CSS</del> later </p>
<br>
<p>My favorite <ins>color</ins> is red.</p>
<br>
<p>This is <sub>subscripted</sub> text.</p>
<br>
<p>This is <sup>superscripted</sup> text.</p>
</body>
```

HTML Text Formatting

We can use some of **attributes** in HTML page not just in *CSS page*

but by the same **syntax** of the CSS

You will learn more about ~~CSS~~ later

My favorite color is red.

This is _{subscripted} text.

This is ^{superscripted} text.

HTML Links -Hyperlinks

- When you click on a link it helps you jump to another document or web page.
- When you move the mouse over a link, the mouse arrow will turn into a little hand.
- A link does not have to be text. It can be an image or any other HTML element.

- **Syntax**

`link text`

The target Attribute

- A local link (link to the same web site) is specified with a relative URL (without <https://www....>).
- **_blank** - Opens the linked document in a new window or tab
- **_self** - Opens the linked document in the same window/tab as it was clicked (this is default)
- **_parent** - Opens the linked document in the parent frame
- **_top** - Opens the linked document in the full body of the window
- **framename** - Opens the linked document in a named frame

HTML bookmarks

- **HTML bookmarks** are used to allow readers to jump to specific parts of a Web page.
- when the link is clicked, the page will scroll to the location with the bookmark.
- Use the **id attribute** (`id="value"`) to define bookmarks in a page
- Use the **href attribute** (`href="#value"`) to link to the bookmark

```
2 <html>
3   <head>
4     <title>
5       |       |       |       HTML Links
6     </title>
7   </head>
8
9   <body style="background-color: □ beige;">
10     <h2>HTML Links</h2>
11     <p><a href="https://www.google.com">Google website</a></p>
12   <hr>
13
14   <h2>Image Links</h2>
15   <a href="https://www.google.com">
16   | 
17   </a>
18
19   <hr>
20   <h2>bookmark</h2>
21   <p><a href="#C4">Jump to Chapter 4</a></p>
22   <p><a href="#C7">Jump to Chapter 7</a></p>
23
24   <h2>Chapter 1</h2>
25   <p>This chapter explains ba bla bla</p>
26
27   <h2>Chapter 2</h2>
28   <p>This chapter explains ba bla bla</p>
29
30   <h2>Chapter 3</h2>
31   <p>This chapter explains ba bla bla</p>
32
```

```
19 </hr>
20 <h2>bookmark</h2>
21 <p><a href="#C4">Jump to Chapter 4</a></p>
22 <p><a href="#C7">Jump to Chapter 7</a></p>
23
24 <h2>Chapter 1</h2>
25 <p>This chapter explains ba bla bla</p>
26
27 <h2>Chapter 2</h2>
28 <p>This chapter explains ba bla bla</p>
29
30 <h2>Chapter 3</h2>
31 <p>This chapter explains ba bla bla</p>
32
33 <h2 id="C4">Chapter 4</h2>
34 <p>This chapter explains ba bla bla</p>
35
36 <h2>Chapter 5</h2>
37 <p>This chapter explains ba bla bla</p>
38
39 <h2>Chapter 6</h2>
40 <p>This chapter explains ba bla bla</p>
41
42 <h2 id="C7">Chapter 7</h2>
43 <p>This chapter explains ba bla bla</p>
44
45 <h2>Chapter 8</h2>
46 <p>This chapter explains ba bla bla</p>
47
48 </body>
49 </html>
```

HTML Links

[Google website](#)

Image Links



bookmark

[Jump to Chapter 4](#)

[Jump to Chapter 7](#)

Chapter 1

This chapter explains ba bla bla

Chapter 2

This chapter explains ba bla bla

Chapter 3

This chapter explains ba bla bla

Absolute URLs **VS.** Relative URLs

Absolute URL (a full web address) in the href attribute.

A relative URL (without the "https://www" part):

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

HTML Image

- Images can improve the design and the appearance of a web page.
- In HTML, images are defined with the **** tag.
- The **** tag is empty, it contains attributes only, and does not have a closing tag.
- The **src** attribute specifies the URL (web address) of the image:
- **Syntax**

- **The alt Attribute:** provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the **src** attribute, or if the user uses a screen reader).
- **Width and Height :** You can use the style attribute to specify the **width** and **height** of an image.
- **Images in Another Folder:** it is common to store images in a sub-folder. You must then include the **folder name** in the **src attribute**:

```

```


Image as Link

- It is common to use images as links:
- **Syntax**

```
<a href="https://www.google.com">  
  
</a>
```

alt attribute: specifies an alternate text for an image, if the image cannot be displayed.

- **Image Floating:** Use the CSS **float property** to let the image float to the **right** or to the **left** of a text:

```

```

- **Background Images:** A background image can be specified on almost any HTML element.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>
5  |       HTML Image
6  |   </title>
7  </head>
8
9  <body background="pic.jpg" >
10
11
12  <h2>Background Image</h2>
13  <p ">
14  |   
15  A paragraph with a floating image on the right
16  |   </p>
17
18  </body>
19  </html>
```

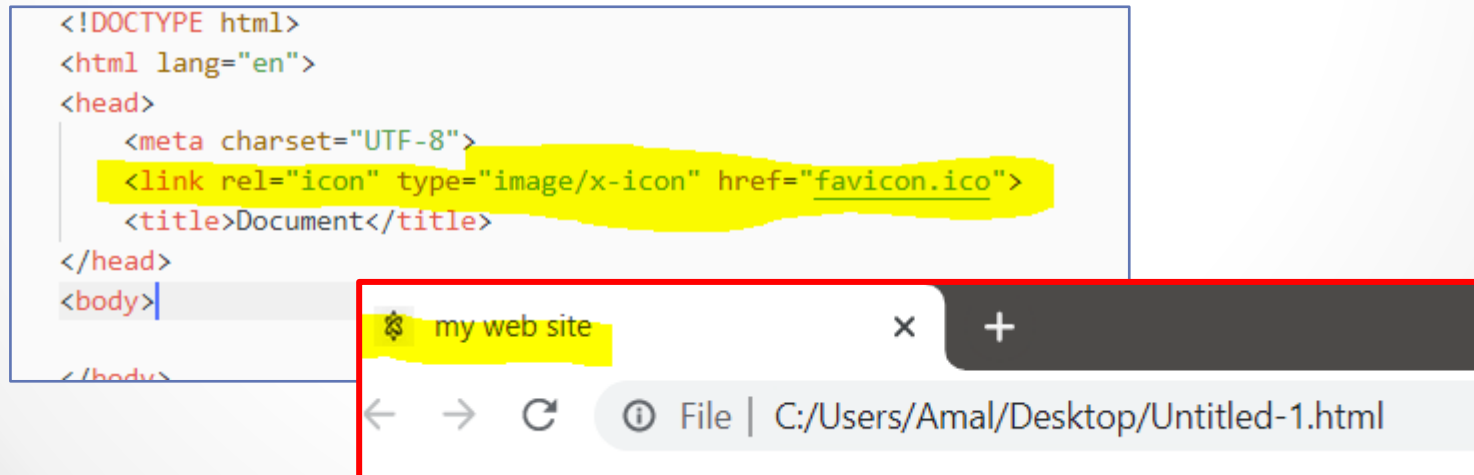
Background Image



A paragraph with a floating image on the right

HTML Favicon

- A favicon is a small image displayed next to the page title in the browser tab.
- You can use any image you like as your favicon. You can also create your own favicon on sites like <https://www.favicon.cc>.
- add a <link> element to your "index.html" file, after the <title> element.



HTML Image Maps

- The idea behind an image map is that you should be able to perform different actions depending on where in the image you click.
- You can use any image as an image map.
- You can use the **onclick** attribute to execute a **JavaScript function** when the area is clicked
- To create an image map you need an **image**, and a **map containing** some rules that describe the clickable areas.

1. The Image

- The image is inserted using the **** tag. The only difference from other images is that you must add a **usemap attribute**:

```

```

- The **usemap** value starts with a hash tag **#** followed by the name of the image map, and is used to **create** a relationship between the **image** and the **image map**.

2. The Map

- Then add a **<map>** element is used to create an image map, and is linked to the image by using the **name attribute**:

```
<map name="workmap">
```

- The **name attribute** must have the **same value** as the **usemap attribute**.

3. The Areas

- A clickable area is defined using an <area> element.

4. Shape

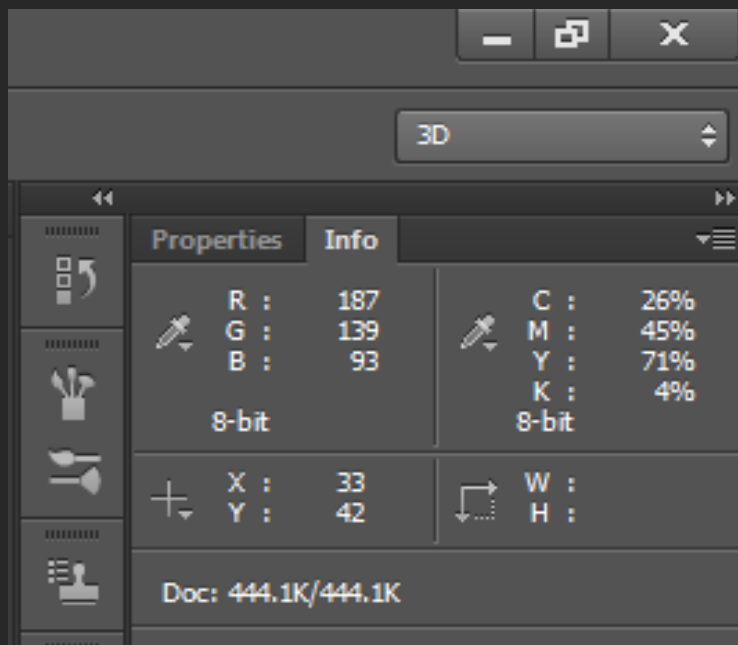
You must define the shape of the area, and you can choose one of these values:

- **rect** - defines a rectangular region
- **circle** - defines a circular region
- **poly** - defines a polygonal region
- **default** - defines the entire region

5. Coordinates

- You must define some coordinates to be able to place the clickable area onto the image.
- The coordinates come in pairs, one for the x-axis and one for the y-axis.

- The coordinates 33, 42 is located 33 pixels from the left margin and 42 pixels from the top:



- The coordinates 271, 349 is located 271 pixels from the left margin and 349 pixels from the top:



Properties		Info	
R :	95	C :	51%
G :	80	M :	54%
B :	37	Y :	95%
		K :	40%
8-bit		8-bit	
X :	271	W :	
Y :	349	H :	
Doc: 444.1K/444.1K			

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>
5              HTML image map
6          </title>
7      </head>
8
9  <body>
10
11  <h2>Image Maps</h2>
12  <p>Click on the computer, to go to a new page and read more about the topic:</p>
13
14  
15
16  <map name="workmap">
17      <area shape="rect" coords="34,44,270,350" alt="Computer" href="html link.html">
18      <area shape="rect" coords="290,170,335,250" alt="phone" href="html image.html">
19      <area shape="circle" coords="341,293,40" alt="cup" href="html style.html">
20  </map>
21
22  </body>
23  </html>
```

Image Maps

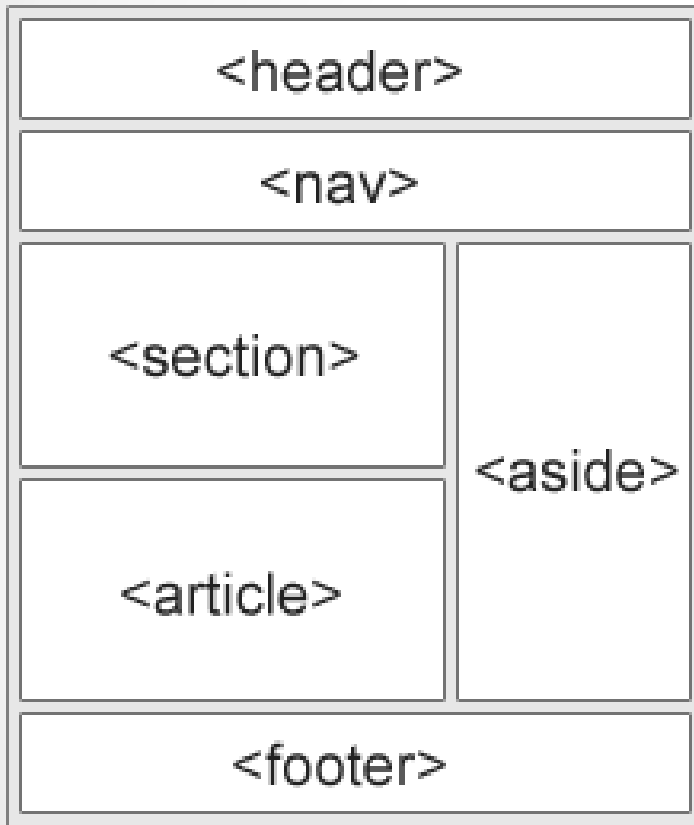
Click on the computer, phone or cup:



Practical exercise

- Create **three** web pages with a logo and image background
- Create **links** between all pages
- Create simple content that includes **paragraphs**, **order list** and **headings** of different formats
- Add social media icons and create actual links.

Layout



- **<header>** - Defines a header for a document or a section
- **<nav>** - Defines a container for navigation links
- **<section>** - Defines a section in a document
- **<article>** - Defines an independent self-contained article
- **<aside>** - Defines content aside from the content (like a sidebar)
- **<footer>** - Defines a footer for a document or a section

HTML List

- An **unordered list** starts with the `` tag. Each list item starts with the `` tag.
- The list items will be marked with bullets (small black circles) by default.
- An **ordered list** starts with the `` tag. Each list item starts with the `` tag.
- The list items will be marked with numbers by default.
- The **definition list** start with `<dl>` tag. Each term start with the `<dt>` tag and `<dd>` contain the definition

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>  HTML List </title>
5  </head>
6  <body >
7  |   <h2>An Unordered HTML List</h2>
8  |   <ul>
9  |   |   <li>Item</li>
10 |   |   <li>Item</li>
11 |   |   <li>Item</li>
12 |   </ul>
13
14 |   <h2>An Ordered HTML List</h2>
15 |   <ol>
16 |   |   <li> First item</li>
17 |   |   <li>Second  item</li>
18 |   |   <li>Third item</li>
19 |   </ol>
20 </body>
21 </html>
22 |
```



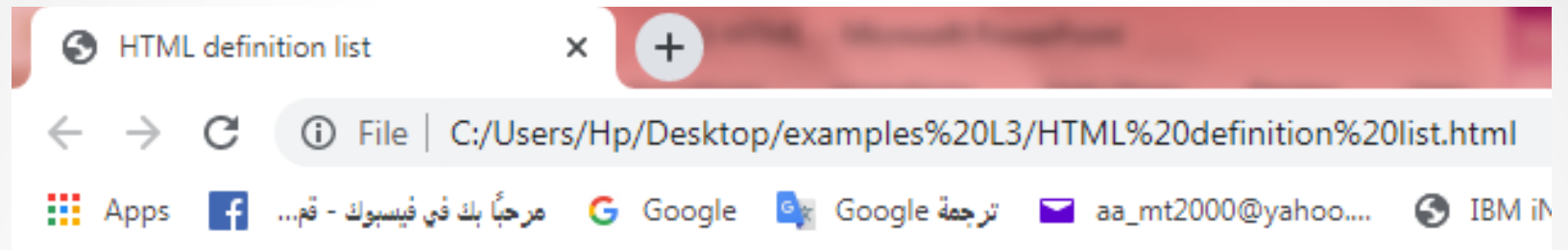

An Unordered HTML List

- Item
- Item
- Item

An Ordered HTML List

1. First item
2. Second item
3. Third item

```
2  <html>
3  <head>
4  |   <title> HTML definition list </title>
5  </head>
6  <body >
7  |   <h2>The definition  HTML List</h2>
8  |   <dl>
9  |   |
10 |   |   <dt>HTML</dt>
11 |   |   <dd>
12 |   |   |   HTML (HyperText Markup Language) is the most basic building block of the Web.<br>
13 |   |   |   It defines the meaning and structure of web content.
14 |   |   </dd>
15 |   |   <dt>CSS</dt>
16 |   |   <dd>
17 |   |   |   Cascading Style Sheets is a style sheet language used <br>
18 |   |   |   for describing the presentation of a document written in a markup language like HTML.
19 |   |   </dd>
20 |   |   <dt>Javascript</dt>
21 |   |   <dd>
22 |   |   |   JavaScript, often abbreviated as JS, is a high-level, just-in-time compiled, <br>
23 |   |   |   object-oriented programming language that conforms to the ECMAScript specification.
24 |   |   </dd>
25 |   </dl>
26 </body>
27 </html>
```



The definition HTML List

HTML

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content.

CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

Javascript

JavaScript, often abbreviated as JS, is a high-level, just-in-time compiled, object-oriented programming language that conforms to the ECMAScript specification.

Unordered HTML List - Choose List Item Marker

The CSS list **style** type property is used to define the style of the list item marker:

Syntax:

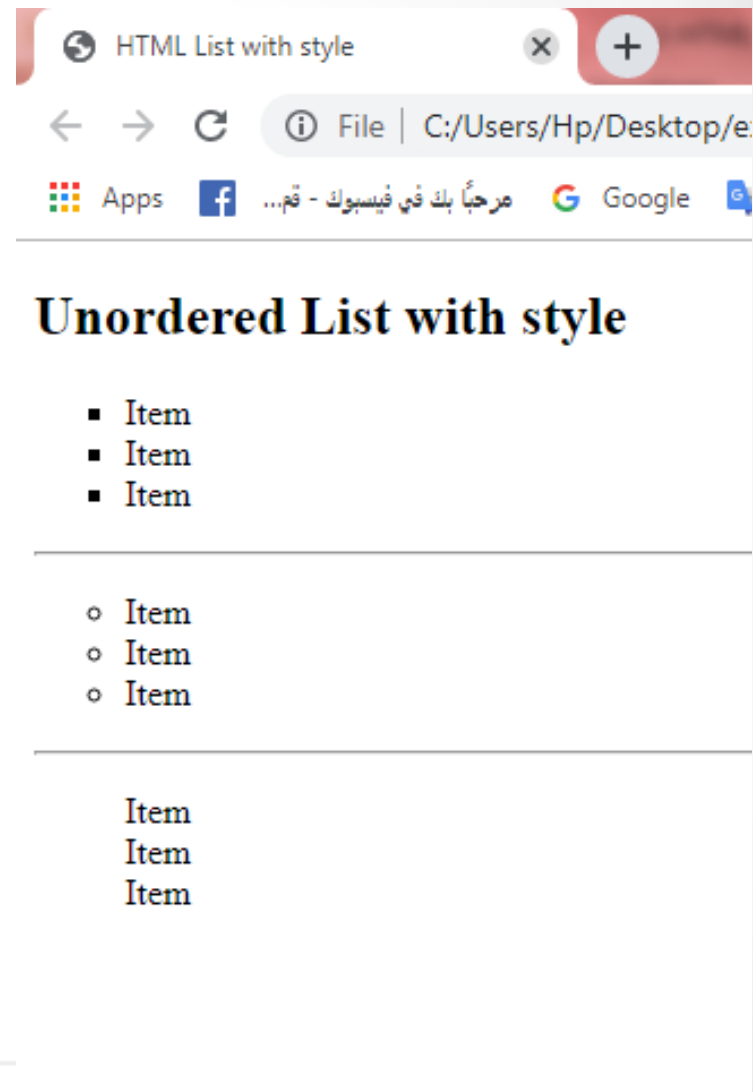
```
<ul style="list-style-type:value;">
```

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

```

4      HTML List with style
5      </title>
6      <body>
7
8      <h2>Unordered List with style</h2>
9
10     <ul style="list-style-type:square;">
11         <li>Item</li>
12         <li>Item</li>
13         <li>Item</li>
14     </ul>
15     <hr>
16     <ul style="list-style-type:circle;">
17         <li>Item</li>
18         <li>Item</li>
19         <li>Item</li>
20     </ul>
21     <hr>
22     <ul style="list-style-type:none;">
23         <li>Item</li>
24         <li>Item</li>
25         <li>Item</li>
26     </ul>
27 </body>
28 </html>
29

```



Ordered HTML List - The Type Attribute

- The **type** attribute of the `` tag, defines the type of the list item marker:
- **Syntax:**

`<ol type="value">`

Type	Description
<code>type="1"</code>	The list items will be numbered with numbers (default)
<code>type="A"</code>	The list items will be numbered with uppercase letters
<code>type="a"</code>	The list items will be numbered with lowercase letters
<code>type="I"</code>	The list items will be numbered with uppercase roman numbers
<code>type="i"</code>	The list items will be numbered with lowercase roman numbers

```
8 <h2>An Ordered HTML List with type attribute </h2>
```

```
9 <ol type="A">
```

```
10 <li> First item</li>
```

```
11 <li>Second item</li>
```

```
12 <li>Third item</li>
```

```
13 </ol>
```

```
14 <hr>
```

```
15 <ol type="a">
```

```
16 <li> First item</li>
```

```
17 <li>Second item</li>
```

```
18 <li>Third item</li>
```

```
19 </ol>
```

```
20 <hr>
```

```
21 <ol type="I">
```

```
22 <li> First item</li>
```

```
23 <li>Second item</li>
```

```
24 <li>Third item</li>
```

```
25 </ol>
```

```
26 <hr>
```

```
27 <ol type="i">
```

```
28 <li> First item</li>
```

```
29 <li>Second item</li>
```

```
30 <li>Third item</li>
```

```
31 </ol>
```

```
32 </body>
```

```
33 </html>
```

HTML List with type attribute

File | C:/Users/Hp/Desktop/examples%20L3/HTML%2

Apps f مرحباً بك في فيسبوك - قم... Google Google ترجمة aa_mt

An Ordered HTML List with type attribute

1. First item
2. Second item
3. Third item

-
- A. First item
 - B. Second item
 - C. Third item

-
- a. First item
 - b. Second item
 - c. Third item

-
- I. First item
 - II. Second item
 - III. Third item

-
- i. First item
 - ii. Second item
 - iii. Third item

HTML Table

- Tables are used in HTML documents (web pages) to present **tabular data**.
- Tables can be broken into sections using the following elements:
 - `<thead>` for Table header
 - `<tbody>` for Table body
 - `<tfoot>` for Table footer
- A table is defined using the `<table>` element.
- contains a number of table cells `<td>` for **table data** which are organized into table rows `<tr>`.
- The markup (HTML code) for a table is always based on rows, never columns.
- The `<caption>` tag defines a table caption.

Table layout ← <caption>

Diagram illustrating the structure of an HTML table layout with labels for various components:

- <table>**: The entire table structure.
- <thead>**: The top section of the table, containing header rows.
- <tbody>**: The main body of the table, containing data rows.
- <tfoot>**: The bottom section of the table, containing footer rows.
- <tr>**: A single row within the table.
- <th>**: A header cell within a row.
- <td>**: A data cell within a row.

A diagram of an HTML table layout. The table is a 7x4 grid. The first row is a header row with a light blue background, containing four dark blue rectangular cells. The last row is a footer row with a light blue background, containing four dark blue rectangular cells. The middle five rows (rows 2-6) are data rows with a white background, each containing four light gray rectangular cells. Brackets on the left side group the rows into three sections: the first row is labeled <thead>, the middle five rows are labeled <tbody>, and the last row is labeled <tfoot>. A large bracket on the far left spans all seven rows and is labeled <table>. On the right side, a bracket groups the first three rows (header, first data row, second data row) and is labeled <tr>. An arrow points from the label <th> to the first cell of the header row. Another arrow points from the label <td> to the last cell of the third data row.

```

7      <table >
8          <caption> savings</caption>
9          <thead>
10             <tr>
11                 <th>Month</th>
12                 <th>Savings</th>
13             </tr>
14         </thead>
15         <tbody>
16             <tr>
17                 <td>January</td>
18                 <td>$100</td>
19             </tr>
20             <tr>
21                 <td>February</td>
22                 <td>$80</td>
23             </tr>
24         </tbody>
25         <tfoot>
26             <tr>
27                 <td>Sum</td>
28                 <td>$180</td>
29             </tr>
30         </tfoot>
31     </table>
32 </body>

```

HTML Tables

File | C:/Users/Hp/Desktop

Apps مرحباً بك في فيسبوك - قم... Google

savings	
Month	Savings
January	\$100
February	\$80
Sum	\$180

- A border is set using the CSS border property:

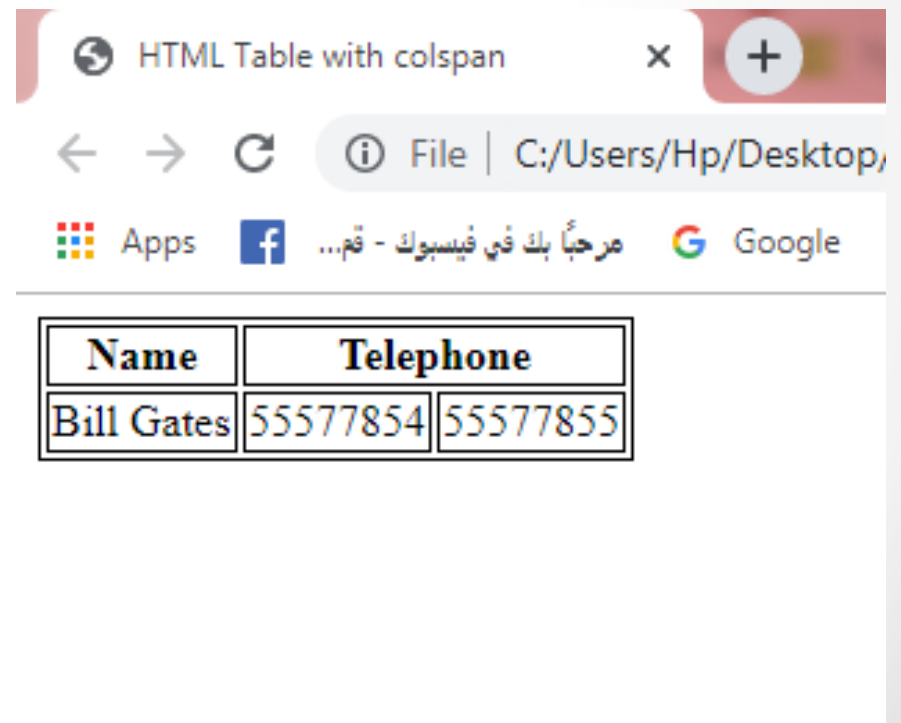
```
table, th, td {  
border: 1px solid black;  
}
```

- Use the **colspan attribute** to make a cell span many columns.
- Use the **rowspan attribute** to make a cell span many rows.

```

1  <html>
2      <head>
3          <title>
4              HTML Table with colspan
5          </title>
6          <style>
7              table, th, td {
8                  border: 1px solid black;
9              }
10         </style>
11     </head>
12     <body>
13         <table>
14             <tr>
15                 <th>Name</th>
16                 <th colspan="2">Telephone</th>
17             </tr>
18             <tr>
19                 <td>Bill Gates</td>
20                 <td>55577854</td>
21                 <td>55577855</td>
22             </tr>
23         </table>
24     </body>
25 </html>

```



```

1  <html>
2      <head>
3          <title>
4              HTML Table with rowspan
5          </title>
6          <style>
7              table, th, td {
8                  border: 1px solid black;
9              }
10         </style>
11     </head>
12     <body>
13         <table>
14             <tr>
15                 <th>Name:</th>
16                 <td>Bill Gates</td>
17             </tr>
18             <tr>
19                 <th rowspan="2">Telephone:</th>
20                 <td>55577854</td>
21             </tr>
22             <tr>
23                 <td>55577855</td>
24             </tr>
25         </table>
26     </body>

```

HTML Table with rowspan

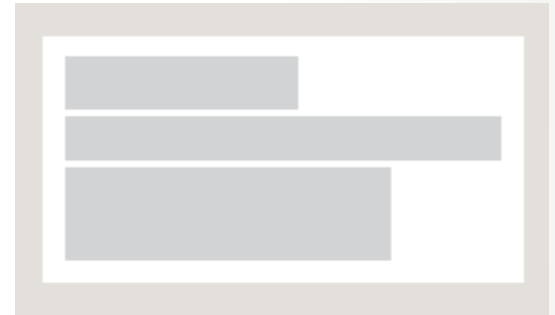
File | C:/Users/Hp/Des

Apps f مرحباً بك في فيسبوك - قم... Goo

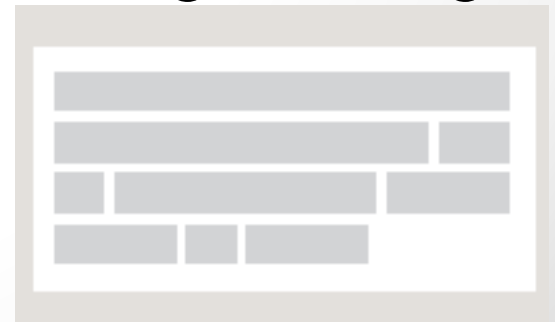
Name:	Bill Gates
Telephone:	55577854
	55577855

HTML Block and Inline Elements

- **Block elements:** Some elements will always appear to start on new line in the browser window.
- Examples of block elements are:
<h1>,<p>,<address>, <div>...etc.



- **Inline elements:** Some elements will always appear to continue on the same line as their neighbouring elements.
- Examples of inline elements are:
,<i>,,.....etc.



HTML Grouping Tags

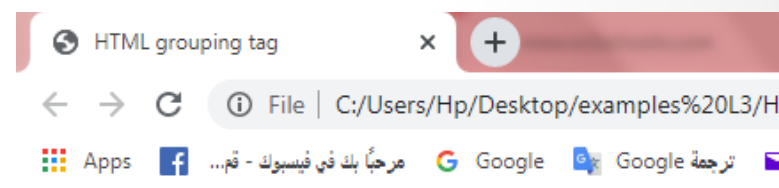
- The `<div>` tag allows you to group a set of element together in one block-level box.
- For example, you might create a `<div>` element to contain all of the elements for header (the logo and the navigation).
- It can also make it easier to follow your code if you have used `<div>` elements to hold each section of the page.

```

<!DOCTYPE html>
<html>
<head>
  <title> HTML grouping tag </title>
</head>
<body >
  <h2>the Div element</h2>
  <div style="background-color: #bisque;width: 300;height: 400;">

  
  <ul>
    <li><a href="aaaaaaa">Home</a></li>
    <li><a href="aaaaaaa">News</a></li>
    <li><a href="aaaaaaa">Contect</a></li>
  </ul>
</div>
</body>
</html>

```



the Div element

HTML and CSS

- [Home](#)
- [News](#)
- [Contect](#)

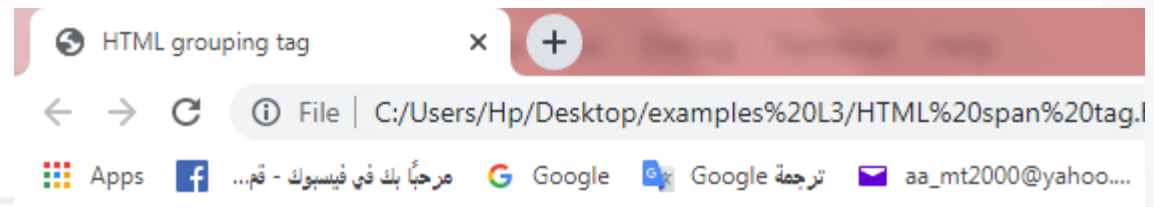
HTML Grouping Tags

- The **** element acts like an inline equivalent of the **<div>** element. it is used to either:
 1. Contain a section of text where there is no other suitable element to differentiate it from its surrounding text.
 2. Contain a number of inline elements.
- You can control the appearance of the content of these elements using **CSS**.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title> HTML grouping tag </title>
5  </head>
6  <body >
7  |   <h2>the span element</h2>
8  |   <p>   HTML <span style="background-color: ■ chocolate;font-style: italic;">(HyperText Markup
9  |   |   Language)</span> is the most basic building block of the Web.<br>
10 |   |   It defines the meaning and structure of web content.
11 |   </p>
12 </body>
13 </html>

```



the span element

HTML (*HyperText Markup Language*) is the most basic building block of the Web. It defines the meaning and structure of web content.

HTML Form

- Traditionally, the term 'form' has referred to a printed document that contains spaces for you to fill in information.
- HTML borrows the concept of form to refer to different elements that allow you to collect information from visitors to your site.
- Whether you are adding a simple search box to your website or you need to create more complicated insurance applications, HTML forms give you a set of elements to collect data from your users.

Why forms?

- The best known form on the web is probably the search box that sits right in the middle of Google's homepage!
- In addition to enabling users to search, forms also allow users to perform other functions online.
- You will see forms when registering as a member of a website , when shopping online, and when signing up for newsletters or mailing lists.

Form Controls

- There are several types of form controls that you can use to collect information from visitors to your site.
- **Adding Text:**
 - Text input(single-line)
 - Password input.
 - Text area(multi-line)
- **Making Choices:**
 - Radio buttons
 - Checkboxes
 - Drop-down boxes
 - Multiple select box

Form Controls

- **Submitting forms:**
 - Submit buttons
 - Image buttons
- **Uploading files:**
 - File upload.

How forms work

- A user fills in a form and then presses a button to submit the information to the server.
- Usually, the form data is sent to a file on the server when the user clicks on the submit button.

1

VOTE FOR YOUR FAVORITE JAZZ MUSICIAN OF ALL TIME

Username:

I vote for:

- ☐ Ella Fitzgerald
- ☒ Herbie Hancock
- ☐ John Coltrane
- ☐ Miles Davis
- ☐ Thelonius Monk

SUBMIT

The name of each form control is sent to the server along with the value the user enters or selects.

2

3

The server processes the information using a programming language such as PHP, C#, VB.net, or Java. It may also store the information in a database.

4

The server creates a new page to send back to the browser based on the information received.

Thank you, lvy!

You voted for Herbie Hancock.

Form Structure

- **<form>** element should always carry the action attribute and will usually have a method and id attribute too.
- **Action** : every <form> element requires an action attribute. Its value is the URL for the page on the sever that will receive the information in the form when it submitted.
- **Method**: forms can be sent using one of two methods: **get** or **post**.

HTML Form Attributes

In the example below, the form data is sent to a file called "[action_page.php](#)". This file contains a server-side script that handles the form data

C: > Users > Amal > Desktop > <> Untitled-2.html > html > body > form > br

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h2>HTML Forms</h2>
6
7  <form action="/action_page.php">
8      <label for="fname">First name:</label><br>
9      <input type="text" id="fname" name="fname" value="John"><br>
10     <label for="lname">Last name:</label><br>
11     <input type="text" id="lname" name="lname" value="Doe"><br><br>
12     <input type="submit" value="Submit">
13 </form>
14
15 <p>If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".</p>
16
17 </body>
18 </html>
```

HTML Form Attributes

← → ↻ ⓘ File | C:/Users/Amal/Desktop/Untitled-2.html

HTML Forms

First name:

Last name:

If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".

Tip: If the **action** attribute is omitted, the action is set to the current page.

HTML Form Attributes

The Target Attribute

The target attribute specifies where to display the response that is received after submitting the form. The target attribute can have one of the following values:

Value	Description
<code>_blank</code>	The response is displayed in a new window or tab
<code>_self</code>	The response is displayed in the current window
<code>_parent</code>	The response is displayed in the parent frame
<code>_top</code>	The response is displayed in the full body of the window

```
<form action="/action_page.php" target="_blank">
```

HTML Form Attributes

The Autocomplete Attribute

The autocomplete attribute specifies whether a form should have autocomplete on or off.

When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

```
<form action="/action_page.php" autocomplete="on">
```

Get Method

With the get method the values from the form are added to the end of the URL specified in the action attribute .

The get method is ideal for:

- Short forms (such as search boxes).
- When you are just retrieving data from the web server (not sending information that should be added to or deleted from a data base).

Post Method

With the post method the values are sent in what are known as HTTP headers. As a rule of thumb you should use the post method if your form:

- Allows user to upload a file .
- Is very long form.
- Contain sensitive data (e.g. passwords).
- Adds information to, or deletes information from , a database.

Note

**If the method attribute is not used,
the form data will be sent using the
get method.**

Adding Text elements and attributes

- **Input element**: is used to create several different form control. It is an empty element.
- **Type attribute**: to choose which of the adding text you need (text or password).
- **Name attribute**: the value of this attribute identifies the form control and is sent along with the information they enter to the server..
- **Size attribute**: used to indicate the width of the text input.
- **Maxlength attribute**: used to limit the number of characters.

Adding Text elements and attributes

- **Textarea element**: is used to create a mutli-line text input. It is **not** empty element.
- **Col attribute** : indicates how wide the text area should be (measured in number of characters).
- **Row attribute**: indicates how many rows the text area should take up vertically.

```
<h2>Login Form</h2>
```

```
<br>
```

```
<form action="http://www.example.com/login.php">
```

```
<p>Username
```

```
<input type="text" name="username" size="15" maxlength="30"/>
```

```
</p>
```

```
<p>Password
```

```
<input type="password" name="password" size="15" maxlength="30"/>
```

```
</p>
```

```
<hr>
```

```
<br>
```

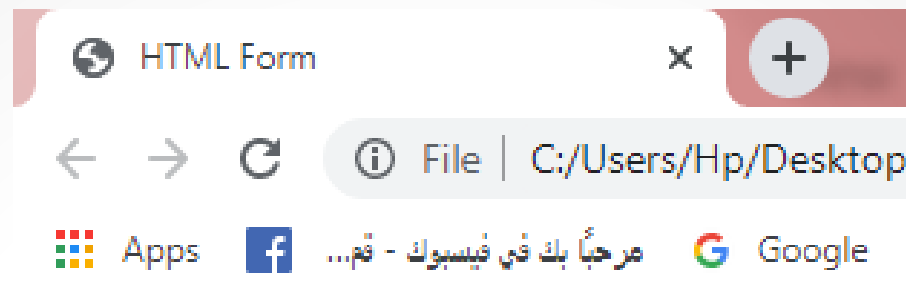
```
<p>what did you think of this Form?</p>
```

```
<textarea name="comments" cols="20" rows="4">
```

```
Enter your comments...
```

```
</textarea>
```

```
</form>
```



Login Form

Username

Password

what did you think of this Form?

Enter your comments...

Making Choices

- **Value attribute:** indicates the value that is sent to the server for the selected option.
- **Checked attribute:** can be used to indicate which value should be selected when the page loads.
- **Select element:** allows user to select one option from a drop down list.
- **Option element:** is used to specify the options that the user can select from.
- **Selected attribute:** can be used to indicate which option should be selected when the page loads.

```
<h2>Making Choices Forms</h2>
```

```
<br>
```

```
<!-- Radio button-->
```

```
<form >
```

```
<p>please select your favorite color:
```

```
<br>
```

```
<input type="radio" name="color" value="red" checked="checked"> Red
```

```
<input type="radio" name="color" value="blue" > Bule
```

```
<input type="radio" name="color" value="green"> Green
```

```
</p>
```

```
</form>
```

```
<hr>
```

```
<!-- chechbox button-->
```

```
<form >
```

```
<p>please select your favorite color:
```

```
<br>
```

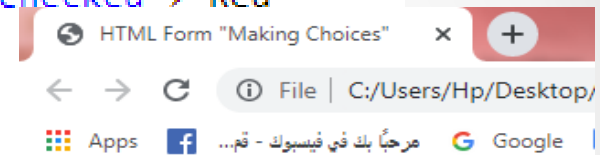
```
<input type="checkbox" name="color" value="red" checked="checked"> Red
```

```
<input type="checkbox" name="color" value="blue" > Bule
```

```
<input type="checkbox" name="color" value="green"> Green
```

```
</p>
```

```
</form>
```



Making Choices Forms

please select your favorite color:

☐ Red ☒ Bule ☐ Green

please select your favorite color:

☐ Red ☒ Bule ☒ Green

```

<!DOCTYPE html>
<html>
<head>
  <title> HTML Form "Making Choices" </title>
</head>
<body >

  <h2>Making Choices Forms</h2>
  <br>

  <!-- drop-down list box-->
  <form >
    <p>please select your favorite color:</p>
    <select name="color">
      <option value="red"> Red </option>
      <option value="bule"> Bule </option>
      <option value="green"> Green </option>
    </select>

  </form>
  <hr>
  <!--multiple select box -->
  <p>please select your favorite color:</p>
  <select name="color" size="3" multiple="multiple">
    <option value="red" selected="selected">Red</option>
    <option value="bule"> Bule </option>
    <option value="green"> Green </option>
  </select>

</body>
</html>

```

Making Choices Forms

please select your favorite color:

Bule ▼

please select your favorite color:

Red ▲
Bule
Green ▼

Uploading and submitting form

<fieldset>

You can group related form controls together inside the `<fieldset>` element. This is particularly helpful for longer forms.

<legend>

The `<legend>` element can come directly after the opening `<fieldset>` tag and contains a caption which helps identify the purpose of that group of form controls.

<label>

When introducing form controls, the code was kept simple by indicating the purpose of each one in text next to it. However, each form control should have its own `<label>` element as this makes the form accessible to vision-impaired users.

`type="file"`

This type of input creates a box that looks like a text input followed by a *browse* button. When the user clicks on the *browse* button, a window opens up that allows them to select a file from their computer to be uploaded to the website.

`type="submit"`

The submit button is used to send a form to the server.


```
<form >
<fieldset>
<legend>Contact details</legend>

<label>First name:<br>
<input type="text" name="First name" maxlength="50">
</label>
<br>
<label>Last name:<br>
  <input type="text" name="Last name" maxlength="50">
</label>
<br>
<label>Telephone:<br>
  <input type="text" name="First name" maxlength="50">
</label>
  <br>
  <p> upload your photo
  <input type="file" name="photo">
  <input type="submit" name="upload" value="upload">
  </p>
  <br>
  <input type="submit" name="subscribe" value="subscribe">
</fieldset>
</form>
```

Contact details form

+

←

→

↻

File

|

C:/Users/Hp/Desktop/example...

☆

a

!

Apps

f

مرحباً بك في فيسبوك - قم...

G

Google

G

Google ترجمة

»

Contact details

First name:

Last name:

Telephone:

upload your photo

Choose File

No file chosen

upload

subscribe

HTML5 Form

Validation helps ensure the user enters information in a form that the server will be able to understand when the form is submitted.

An example of HTML5 form validation is the `required` attribute, which can be used on any form element that the user is expected to fill in.

`type="search"`

If you want to create a single line text box for search queries, HTML5 provides a special search input.

`type="email"`

If you ask a user for an email address, you can use the email input. Browsers that support HTML5 validation will check that the user has provided information in the correct format of an email address. Some smart phones also optimize their keyboard to display the keys you are most likely to need when entering an email address (such as the `@` symbol).

`type="date"`

If you are asking the user for a date, you can use an `<input>` element and give the `type` attribute a value of `date`.

This will create a date input in browsers that support the new HTML5 input types.

`placeholder`

On any text input, you can also use an attribute called `placeholder` whose value is text that will be shown in the text box until the user clicks in that area.

```
<form >
<fieldset>
<legend>HTML5 form</legend>
<label>Name
<input type="text" name="name" maxlength="30" required="required">
<br>
</label>
<label>Data of brith</label>
<input type="date" name="date">
<br>
<label>Email address</label>
<input type="email" name="email">
<br>
<label>search</label>
<input type="search" name="search" placeholder="Enter name">
<input type="button" value="Search">
<br>
<input type="submit" name="submit">

</fieldset>
</form>
```

HTML5 form

Name

Data

Email

search



Please fill out this field.

HTML5 form

Name

Data of birth

Email address

search



Please include an '@' in the email address. 'askjkj' is missing an '@'.

HTML5 form

Name

Data of birth

Email address

search

May, 2019

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

HTML5 form

Name

Data of birth

Email address

search

HTML Input Types

- <input type="button">
- <input type="color">
- <input type="datetime-local">
- <input type="file">
- <input type="image">
- <input type="number">
- <input type="radio">
- <input type="reset">
- <input type="submit">
- <input type="text">
- <input type="url">

- <input type="checkbox">
- <input type="date">
- <input type="email">
- <input type="hidden">
- <input type="month">
- <input type="password">
- <input type="range">
- <input type="search">
- <input type="tel">
- <input type="time">
- <input type="week">

Summary

- Whenever you want to collect information from visitors you will need a form, which lives inside a `<form>` element.
- Information from a form is sent in name / value pairs.
- Each form control is given a name, and the text the user types in or the values of the options they select are sent to the server.
- HTML5 introduces new form elements which make it easier for visitors to fill in forms.

Adding a flash movie to your web page

- The most popular way of adding flash into a web page is using **JavaScript**.
- There are several scripts that allow you to do this without an in-depth understanding of the JavaScript language.
- One **advantage** to using this technique is that it allows browsers to show alternative content for user whose browsers are not capable of showing flash.

Understanding video formats and players

- **Formats** :movies are available in many formats (Blu-ray, DVD, VHS, etc. online ,there are even more video formats (AVI, Flash video, H264, MPEG, etc.
- **Players/plugins**: browsers were initially designed to show text and images only. For this reason, browsers built prior to 2010 generally required another program called a player or plugin to be installed in order to play video content.
- **Approach**: the easiest way to add video to your site is to use a hosted service such as YouTube or Vimeo. Or by using HTML5 <video>.

HTML5, adding video

- The **<video>** element has a number of attributes which allow you to control video playback:

src

This attribute specifies the path to the video. (The example video is in H264 format so it will only work in IE and Safari.)

poster

This attribute allows you to specify an image to show while the video is downloading or until the user tells the video to play.

width, height

These attributes specify the size of the player in pixels.

controls

When used, this attribute indicates that the browser should supply its own controls for playback.

autoplay

When used, this attribute specifies that the file should play automatically.

loop

When used, this attribute indicates that the video should start playing again once it has ended.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Adding HTML5 Video</title>
```

```
  </head>
```

```
  <body>
```

```
    <video src="video/puppy.mp4"
```

```
      poster="images/puppy.jpg"
```

```
      width="400" height="300"
```

```
      preload
```

```
      controls
```

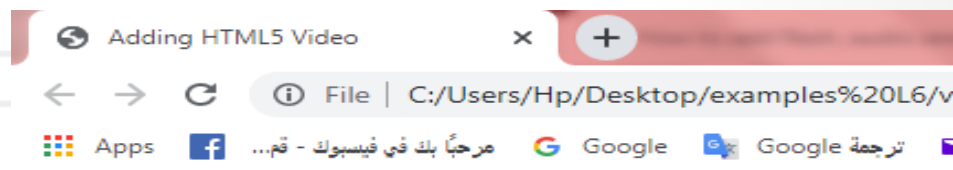
```
      loop>
```

```
    <p>A video of a puppy playing in the snow</p>
```

```
  </video>
```

```
</body>
```

```
</html>
```



Multiple video sources

```
<!DOCTYPE html>
<html>
  <head>
    <title>Multiple Video Sources</title>
  </head>
  <body>
    <video poster="images/puppy.jpg"
      width="400" height="320" preload controls loop>

      <source src="video/puppy.mp4"
        type='video/mp4;codecs="avc1.42E01E, mp4a.40.2"' />

      <source src="video/puppy.webm"
        type='video/webm;codecs="vp8, vorbis"' />

      <p>A video of a puppy playing in the snow</p>
    </video>
  </body>
</html>
```

HTML5, adding audio

- The **<audio>** element to include audio file in you pages.

src

This attribute specifies the path to the audio file.

controls

This attribute indicates whether the player should display controls. If you do not use this attribute, no controls will be shown by default. You can also specify your own controls using JavaScript.

autoplay

The presence of this attribute indicates that the audio should start playing automatically. (It is considered better practice to let visitors choose to play audio.)

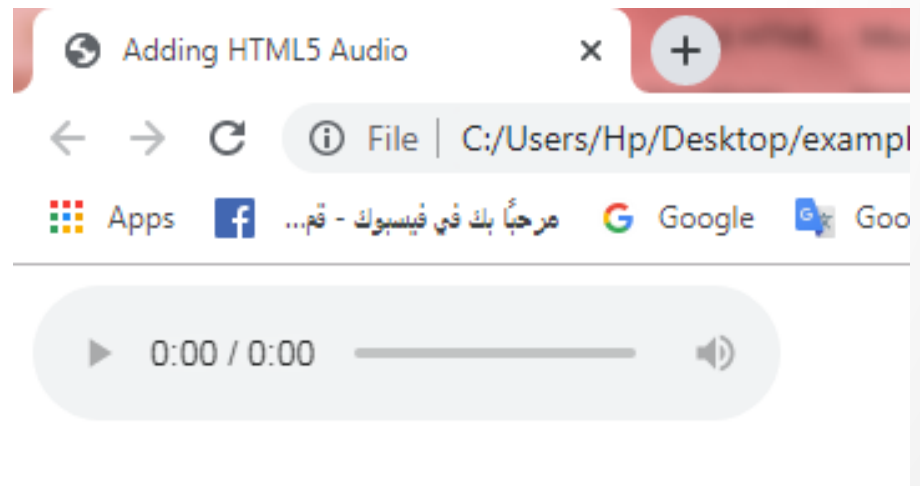
preload

This attribute indicates what the browser should do if the player is not set to autoplay.

loop

This attribute specifies that the audio track should play again once it has finished.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Adding HTML5 Audio</title>
  </head>
  <body>
    <audio src="audio/test-audio.ogg" controls autoplay>
  </audio>
</body>
</html>
```



Multiple audio sources

```
<!DOCTYPE html>
<html>
  <head>
    <title>Multiple Audio Sources</title>
  </head>
  <body>
    <audio controls autoplay>
      <source src="audio/test-audio.ogg" />
      <source src="audio/test-audio.mp3" />
    </audio>
  </body>
</html>
```

Layout

Traditional HTML layouts:

- For a long time, web page authors used **<div>** elements to group together related elements on the page(such as the elements that form header, an article, a footer or sidebar).
- Authors used class or id attributes to indicate the role of the **<div>** element in the structure of the page.


```
<body>

<div id="page">

  <div id="header">

    .....
    <div id="nav">
    .....

  <div id="content">

    .....
    <div class="article">
    .....

    .....
    <div class="article">
    .....

  <div id="sidebar">

  <div id="footer">
```

```
<div id="page">
  <div id="header">
    .....
    <div id="nav">
      .....
  </div>
  <div id="content">
    .....
    <div class="article">
      .....
    </div>
    .....
    <div class="article">
      .....
    </div>
    .....
  </div>
  <div id="sidebar">
    .....
  </div>
  <div id="footer">
    .....
  </div>
</div>
```

```
<div id="header">

+++++
<div id="nav">
+++++
```

```
<div id="nav">
```

```
<div id="content">
  <div class="article">
  <div class="article">
```

```
<div class="article">
```

```
<div class="article">
```

```
<div id-  
"sidebar">
```

```
<div id="footer">
```

```
<div id="footer">
```

HTML5 Layout

- HTML5 introduces a new set of elements that allow you to divide up the parts of a page.
- The names of these elements indicate the kind of content you will find in them.
- They are still subject to change, but that has not stopped many web page authors using them already.
- The point of creating these new elements is so that web page authors can use them to help describe the structure of the page.
- It also makes the code easier to follow.

<body>

<div id="page">

<header>

<nav>

<div id="content">

<article>

<article>

<aside>

<footer>

Headers and Footers

- The **<div>** element will remain an important way to group together related elements.
- The **<header>** and **<footer>** element can be used for:
- The main **header or footer** that appears at the top or bottom of every page on the site.
- A **header or footer** for an individual **<article>** or **<section>** within the page.

Navigation and articles

- The **<nav>** element is used to contain the major navigational blocks on the site such as the primary site navigation.
- The **<article>** element acts as a container for any section of the page that could stand alone and potentially be syndicated.
- This could be an individual article or blog entry, a comment or forum post, or any other independent piece of content.
- It can even be nested inside each other. For example , a blog post might live inside one **<article>** element and each comment on that article could live inside its own child **<article>** element

Asides

- The **<aside>** element has two purposes, depending on whether it is inside an **<article>** element or not.
- When the **<aside>** element is used inside an **<article>** element , it should contain information that is related to the article but not essential to its overall meaning.
- When the **<aside>** element is used outside an **<article>** element, it acts as a container for content that is related to the entire page.

Sections and figures

- The **<section>** element groups related content together, and typically each section would have its own heading.
- **<figure>** It can be used to contain any content that is referenced from the main flow of an article.

Examples of usage include:

- Images
- Videos
- Graphs
- Diagrams
- Code samples
- Text that supports the main body of an article.
- **<figcaption>** element which provides a text description for the content of the **<figure>**

Web Design

layout with CSS

HOME NEWS NOTIFICATIONS

LOGIN

Content section

Aside tag example

Aside tag content. . .

Footer Section © 2022


```
<!-- Header Section -->
<header>
  <div class="head1">Web Design</div>
  <div class="head2"> layout with CSS </div>
</header>

<!-- Menu Navigation Bar -->
<nav class="menu">
  <a href="#home">HOME</a>
  <a href="#news">NEWS</a>
  <a href="#notification">NOTIFICATIONS</a>
  <div class="menu-log">
    <a href="#login">LOGIN</a>
  </div>
</nav>

<!-- asid bar -->
<aside>
  <h1>Aside tag example</h1>
  <p>Aside tag content. . .</p>
</aside>

<!-- Body section -->
<main class = "body_sec">
  <section id = "Content">
```

```
92  <!-- Body section -->
93  <main class = "body_sec">
94    <section id="Content">
95      <h3>Content section</h3>
96    </section>
97
98  </main>
99
100  <!-- Footer Section -->
101  <footer>Footer Section &copy; 2022</footer>
102 </body>
103 </html>
104
```

```

5 <style>
6   .head1 {
7     font-size:40px;
8     color: #2918e0;
9     font-weight:bold;
10  }
11  .head2 {
12    font-size:17px;
13    margin-left:10px;
14    margin-bottom:15px;
15  }
16  body {
17    margin: 0 auto;
18    background-position:center;
19    background-size: contain;
20  }
21
22  .menu {
23    position: sticky;
24    top: 0;
25    background-color: #240099d3;
26    padding:10px 0px 10px 0px;
27    color: white;
28    margin: 0 auto;
29    overflow: hidden;
30  }

```

```

31  .menu a {
32    float: left;
33    color: white;
34    text-align: center;
35    padding: 14px 16px;
36    text-decoration: none;
37    font-size: 20px;
38  }
39  .menu-log {
40    right: auto;
41    float: right;
42  }
43  footer {
44    width: 100%;
45    bottom: 0px;
46    background-color: rgba(21, 6, 99, 0.965);
47    color: #fff;
48    position: absolute;
49    padding-top:20px;
50    padding-bottom:50px;
51    text-align:center;
52    font-size:30px;
53    font-weight:bold;
54  }
55  .body_sec {
56    margin-left:20px;

```

```

57  }
58  aside {
59    width: 40%;
60    float: right;
61    background-color: rgb(170, 182, 236);
62    color: white;
63    padding: 5px;
64    margin: 10px;
65    height: 100px;
66  }
67 </style>
68 /head>
69
70 body>
71
72 <!-- Header Section -->
73 <header>
74   <div class="head1">Web Design</div>
75   <div class="head2"> layout with CSS </div>
76 </header>
77
78 <!-- Menu Navigation Bar -->
79 <nav class="menu">
80   <a href="#home">HOME</a>

```



YOKO'S KITCHEN

JAPANESE COOKING CLASSES

[home](#)[classes](#)[catering](#)[about](#)[contact](#)

Bok Choi

Japanese Vegetarian

Five week course in London

A five week introduction to traditional Japanese vegetarian meals, teaching you a selection of rice and noodle dishes.



Teriyaki Sauce

Sauces Masterclass

One day workshop

An intensive one-day course looking at how to create the most delicious sauces for use in a range of Japanese cookery.

Popular Recipes

[Yakitori \(grilled chicken\)](#)

[Tsukune \(minced chicken patties\)](#)

[Okonomiyaki \(savory pancakes\)](#)

[Mizutaki \(chicken stew\)](#)

Contact

Yoko's Kitchen
27 Redchurch Street
Shoreditch
London E2 7DP

```
<body>
  <div class="wrapper">
    <header>
      <h1>Yoko's Kitchen</h1>
      <nav>
        <ul>
          <li><a href="" class="current">home</a></li>
          <li><a href="">classes</a></li>
          <li><a href="">catering</a></li>
          <li><a href="">about</a></li>
          <li><a href="">contact</a></li>
        </ul>
      </nav>
    </header>
    <section class="courses">
      <article>
        <figure>
          
          <figcaption>Bok Choi</figcaption>
        </figure>
        <hgroup>
          <h2>Japanese Vegetarian</h2>
          <h3>Five week course in London</h3>
        </hgroup>
      </article>
    </section>
  </div>
</body>
```

```
</hgroup>
<p>A five week introduction to traditional Japanese vegetarian meals,
  teaching you a selection of rice and noodle dishes.</p>
</article>
<article>
  <figure>
    
    <figcaption>Teriyaki Sauce</figcaption>
  </figure>
  <hgroup>
    <h2>Sauces Masterclass</h2>
    <h3>One day workshop</h3>
  </hgroup>
  <p>An intensive one-day course looking at how to create the most delicious
    sauces for use in a range of Japanese cookery.</p>
</article>
```

```
</section>
```

```
<aside>
```

```
  <section class="popular-recipes">
    <h2>Popular Recipes</h2>
    <a href="">Yakitori (grilled chicken)</a>
    <a href="">Tsukune (minced chicken patties)</a>
    <a href="">Okonomiyaki (savory pancakes)</a>
    <a href="">Mizutaki (chicken stew)</a>
  </section>
  <section class="contact-details">
```

```
<section class="contact-details">
  <h2>Contact</h2>
  <p>Yoko's Kitchen<br />
    27 Redchurch Street<br />
    Shoreditch<br />
    London E2 7DP</p>
</section>
</aside>
<footer>
  &copy; 2011 Yoko's Kitchen
</footer>
</div><!-- .wrapper -->
</body>
```

ID attribute

- Every HTML element can carry the **id attribute**.
- The **id attribute** is known as **a global attribute** because it can be used on any element.
- It is used to uniquely identify that element from other elements on the page.
- Its value should start with a letter or an underscore (**not** number or any other character).
- It is important that no two elements on the same page have the same value for their **id attributes** (otherwise the value is no longer unique).

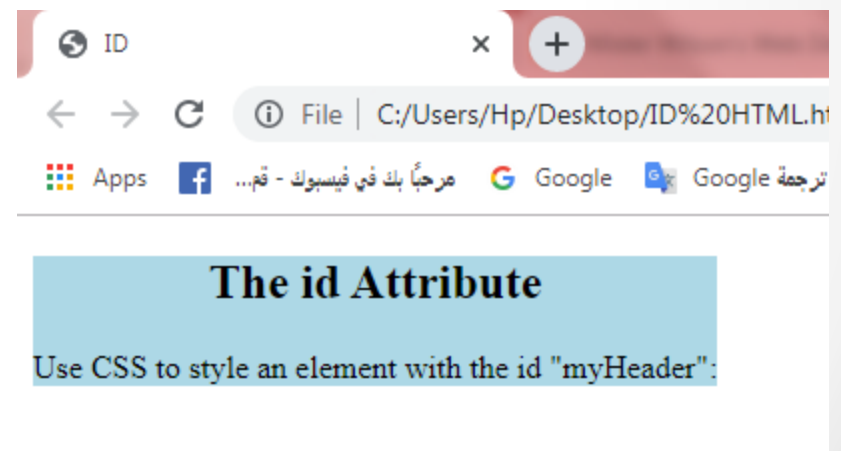
CSS and JavaScript

- Giving an element a unique identity allows you to style it differently than any other instance of the same element on the page.
- Example: you might want to assign one paragraph within the page (perhaps a paragraph containing a pull quote) a different style than all of the other paragraphs.
- Is allows you to add interactivity to your pages ,and it can be used to allow the script to work with that particular element.

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

```
<head>
  <title>ID</title>
  <style>
    #mystyle {
      background-color: lightblue;
      width: max-content;
      height: max-content;
      text-align: center;
    }
  </style>
</head>

<body>
  <div id="mystyle">
    <h2>The id Attribute</h2>
    <p>Use CSS to style an element with the id "mystyle":</p>
  </div>
```



Class attribute

- Every HTML element can also carry a **class attribute**.
- The **class attribute** selects elements with a specific **class attribute**.
- Different Tags Can Share Same Class
- HTML elements can have more than one **class name**, each **class name** must be **separated** by a space.
- To select **elements** with a specific class, write a period (.) character, followed by the name of the **class**.

```

1  <!DOCTYPE html>
2  <html>
3  <style>
4  .city {
5      background-color: ■ tomato;
6      color: ■ white;
7  }
8  }
9  .main {
10     text-align: center;
11 }
12 </style>
13 <body>
14
15 <h2>Multiple Classes</h2>
16 <p>All three headers have the class name "city".
17     In addition, London also have the class name "main",
18     which center-aligns the text.</p>
19
20 <h2 class="city main">London</h2>
21 <h2 class="city">Paris</h2>
22 <h2 class="city">Tokyo</h2>
23
24 </body>
25 </html>

```

Multiple Classes

All three headers have the class name "city". In addition, London also have the class name "main", which center-aligns the text.

London

Paris

Tokyo

IFrame

- An **iframe** is like a little window that has been cut into your page – and in that window you can see another page.
- The term **iframe** is an abbreviation of inline frame.
- One common use of **iframes** (that you may have seen on various websites) is to embed a Google Map into a page.
- The content of the **iframe** can be any html page (either located in the same server or anywhere else on the web).
- An **iframe** is created using the **<iframe>** element.

IFRAME

- There are few attributes that you will need to know to use it:

src

The `src` attribute specifies the URL of the page to show in the frame.

height

The `height` attribute specifies the height of the iframe in pixels.

width

The `width` attribute specifies the width of the iframe in pixels.

scrolling

The `scrolling` attribute will not be supported in HTML5. In HTML 4 and XHTML, it indicates whether the iframe should have scrollbars or not. This is important if the page inside the iframe is larger than the space you have allowed for it (using the `height` and `width` attributes). Scrollbars allow the user to move around the frame to see more content. It can take one of three values: `yes` (to show scrollbars), `no` (to hide scrollbars) and `auto` (to show them only if needed).

frameborder

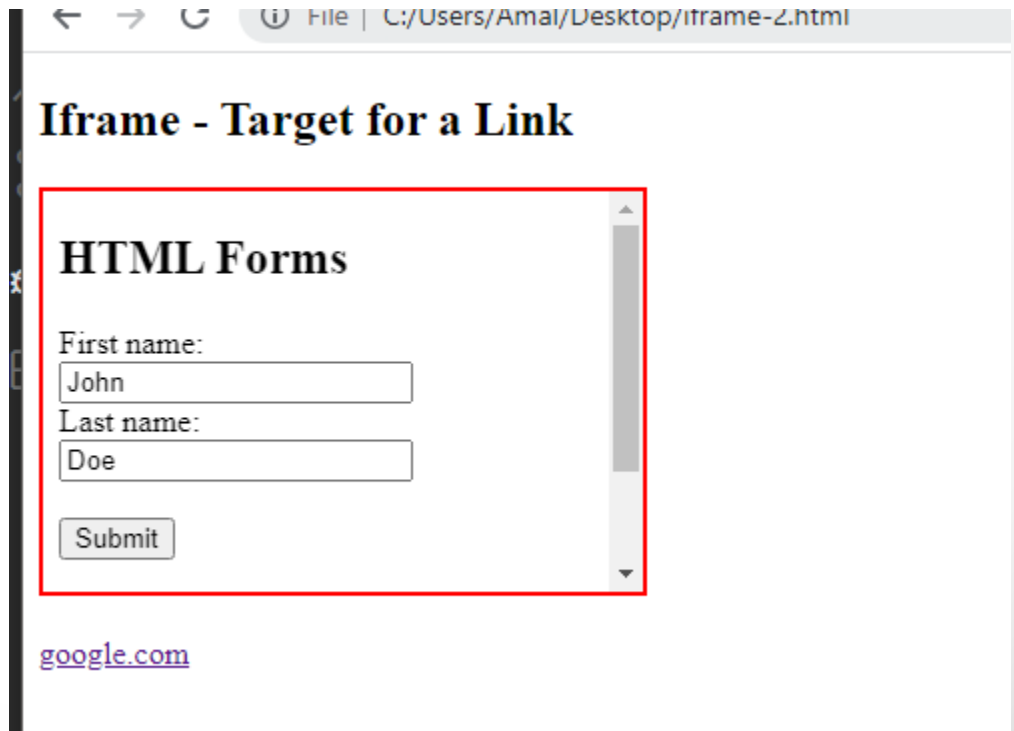
The `frameborder` attribute will not be supported in HTML5. In HTML 4 and XHTML, it indicates whether the frame should have a border or not. A value of 0 indicates that no border should be shown. A value of 1 indicates that a border should be shown.

seamless

In HTML5, a new attribute called `seamless` can be applied to an iframe where scrollbars are not desired.

C: > Users > Amal > Desktop > <> iframe-2.html > ...

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h2>Iframe - Target for a Link</h2>
6  <iframe src="page1.html"  name="iframe_1" height="200" width="300" title="Iframe Example"
7  | style="border:2px solid red;" >
8  </iframe>
9
10 <p><a href="https://www.google.com" target="iframe_1"> google.com </a>
11
12
13 </body>
14 </html>
15
```



ESCAPE Characters

- There are some **characters** that are used in and reserves by HTML code.
- If want these characters to appear on your page you need to use what are termed “**escape**” characters (also known as escape codes or entity references).
- When using **escape** characters, it is important to check the page in your browser to ensure that the correct symbol shows up.

< **Less-than sign**
<
<

¢ **Cent sign**
¢
¢

‘ **Left single quote**
‘
‘

> **Greater-than sign**
>
&

£ **Pound sign**
£
£

’ **Right single quote**
’
’

& **Ampersand**
&
&

¥ **Yen sign**
¥
¥

“ **Left double quotes**
“
“

” **Quotation mark**
"
"

€ **Euro sign**
€
€

” **Right double quotes**
”
”

© **Copyright symbol**
©
©

× **Multiplication sign**
×
×


```
<!DOCTYPE html>
```

```
<html>
```

```
<head> <title>
```

ESCAPE Characters HTML

```
</title>
```

```
</head>
```

```
<body>
```

```
<p>&copy; the art book</p>
```

```
<p>&cent;cent sign</p>
```

```
<p>&euro;euro sign</p>
```

```
<p>&ldquo;left double quotes</p>
```

```
<p>&#165;yen sign </p>
```

```
<p>&#38;Ampersand</p>
```

```
<p>&#60;less-than sign</p>
```

```
</body>
```

```
</html>
```

ESCAPE Characters HTML

← → ↻ ⓘ File | C:/U

Apps f بَأْ بَكَ فِي فَيْسِبُوكْ - قَم...

© the art book

¢cent sign

€euro sign

“left double quotes

¥yen sign

&Ampersand

<less-than sign

Resources

- HTML elements Wiki from W3C <https://www.w3.org/community/webed/wiki/HTML/Elements>.
- Website – Wikipedia <https://en.wikipedia.org/wiki/Website>
- HTML tutorial at MDN Web Docs <https://developer.mozilla.org/en-US/docs/Learn/HTML>
- HTML tutorial at w3schools <http://www.w3schools.com/html/>