

HTML Classes, id, graphics and media.

HAJJAT HADIJAH KYAMPEIRE

CLASSES

The HTML **class** attribute is used to specify a class for an HTML element. Multiple HTML elements can share the same class.

Using The class Attribute

The **class** attribute is often used to point to a class name in a style sheet. It can also be used by a JavaScript to access and manipulate elements with the specific class name.

In the following example we have two `<div>` elements with a **class** attribute with the value of "city". All of the two `<div>` elements will be styled equally according to the **.city** style definition in the head section:

E.g

```
<div class="city"><h2>London</h2><p>London is the capital of England.</p></div>
```

```
<div class="city"><h2>Paris</h2><p>Paris is the capital of France.</p></div>
```

In the css file or style tags add styles for that class.

Multiple Classes

HTML elements can belong to more than one class.

To define multiple classes, separate the class names with a space, e.g. `<div class="city main">`. The element will be styled according to all the classes specified.

In the following example, the first `<h2>` element belongs to both the `city` class and also to the `main` class, and will get the CSS styles from both of the classes:

```
<h2 class="city main">London</h2>
```

```
<h2 class="city">Paris</h2>
```

You can add additional styles to the second element and that will differentiate it from the other.

Exercise: Make London city bold and paris should not be bold

Use of The class Attribute in JavaScript

The class name can also be used by JavaScript to perform certain tasks for specific elements. JavaScript can access elements with a specific class name with the `getElementsByClassName()` method:

```
<script>
```

```
function myFunction() {
```

```
    var x = document.getElementsByClassName("city");
```

```
    for (var i = 0; i < x.length; i++) {
```

```
        x[i].style.display = "none";
```

```
    }
```

```
}
```

```
</script>
```

NB: The script tag go below the body.

HTML id Attribute

The HTML **id** attribute is used to specify a unique id for an HTML element. You **cannot** have more than one element with the same id in an HTML document.

Using The id Attribute: The **id** attribute specifies a unique id for an HTML element. The value of the **id** attribute must be unique within the HTML document.

The **id** attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id. The syntax for id is: write a hash character (#), followed by an id name. Then, define the CSS properties within curly braces {}.

In the following example we have an `<h1>` element that points to the id name "myHeader". This `<h1>` element will be styled according to the `#myHeader` style definition in the head section:

```
<h1 id="myHeader">My Header</h1>
```

```
#myHeader { background-color: lightblue; color: black; padding: 40px; text-align: center; }
```

HTML Bookmarks with ID and Links

HTML bookmarks are used to allow readers to jump to specific parts of a webpage. Bookmarks can be useful if your page is very long. To use a bookmark, you must first create it, and then add a link to it. Then, when the link is clicked, the page will scroll to the location with the bookmark.

First, create a bookmark with the `id` attribute: `<h2 id="C4">Chapter 4</h2>`

Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

`Jump to Chapter 4` Add a bookmark to jump to page 10

Add multiple pages including upto page 15:

`<h2>Chapter 1</h2>`

`<p>This chapter explains ba bla bla</p>`

`<h2>Chapter 2</h2>`

`<p>This chapter explains ba bla bla</p>`

Using The id Attribute in JavaScript

The **id** attribute can also be used by JavaScript to perform some tasks for that specific element. JavaScript can access an element with a specific id with the **getElementById()** method: Use the **id** attribute to manipulate text with JavaScript:

```
<script>
```

```
function displayResult() {
```

```
    document.getElementById("myHeader").innerHTML = "Have a nice day!";
```

```
}
```

```
</script>
```

Exercise:

- a) Add the correct HTML attribute to make the H1 element red.
- b). Use Javascript to add your name to the P element

HTML Media

What is Multimedia?

Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more.

Multimedia Formats

Multimedia elements (like audio or video) are stored in media files. The most common way to discover the type of a file, is to look at the file extension.

Multimedia files have formats and different extensions like: .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

HTML Video

The HTML `<video>` element is used to show a video on a web page.

```
<video width="320" height="240" controls><source src="movie.mp4" type="video/mp4"><source  
src="movie.ogg" type="video/ogg">Your browser does not support the video tag.</video>
```


How it Works

The **controls** attribute adds video controls, like play, pause, and volume.

It is a good idea to always include **width** and **height** attributes. If height and width are not set, the page might flicker while the video loads.






The **<source>** element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the **<video>** and **</video>** tags will only be displayed in browsers that do not support the **<video>** element.

To start a video automatically, use the **autoplay** attribute:

Browser Support

The numbers in the table specify the first browser version that fully supports the **<video>** element.

Element					
<video>	4.0	9.0	3.5	4.0	10.5

HTML Video - Methods, Properties, and Events

The HTML DOM defines methods, properties, and events for the `<video>` element. This allows you to load, play, and pause videos, as well as setting duration and volume. There are also DOM events that can notify you when a video begins to play, is paused, etc.

```
<div style="text-align:center"> <button onclick="playPause()">Play/Pause</button> <button  
onclick="makeBig()">Big</button>
```

```
<button onclick="makeSmall()">Small</button><button onclick="makeNormal()">Normal</button><br><br>
```

```
<video id="video1" width="420">
```

```
<source src="mov_bbb.mp4" type="video/mp4">
```

```
<source src="mov_bbb.ogg" type="video/ogg">
```

Your browser does not support HTML video.

```
</video>
```

```
</div>
```

Scripts

<script>

```
var myVideo = document.getElementById("video1");
```

```
function playPause() {
```

```
    if (myVideo.paused) myVideo.play(); else myVideo.pause();
```

```
}
```

```
function makeBig() { myVideo.width = 560; }
```

```
function makeSmall() { myVideo.width = 320; }
```

```
function makeNormal() { myVideo.width = 420; }
```

</script>

Next week Test