

INFX 1606

INTRODUCTION TO
WEBSITE CREATION

HTML & CSS - 6

Raghav V. Sampangi

Instructor, Faculty of Computer Science, Dalhousie University

raghav@cs.dal.ca

Three things for today

- Including media (images, video, audio) – cont'd
- HTML links
 - No, not anchor links!
- CSS Selectors - 1

Including “id” for elements: A discussion

- It may be tempting to include *id*'s for all elements in your markup.
 - You don't have to!
- Include *id*'s when it matters:
 - When you expect the element to change its appearance or functionality during user interaction.
 - When you expect the content of the element to change with update to a database or other data storage.
 - When you expect the appearance or functionality of an element to change with changes to the medium of access (i.e. device, browser resizing, etc.).

HTML Media (Cont'd)

Captioning figures

Often, when we include figures in articles, we might consider including captions.



Captioning figures

By default, **** tags don't allow including any text other than the alternate text (using the **alt**) attribute.

To include captions, you will need to ***encapsulate*** the image inside a ***figure*** element, and include a ***figure caption***.

For this, you would use the **<figure>** and **<figcaption>** tags.

Captioning figures

<figure> tag specifies self-contained content, such as pictures, graphs, illustrations, etc.

New in HTML5.

Acts like an image and caption container, and can be manipulated using JavaScript, and styled using CSS.

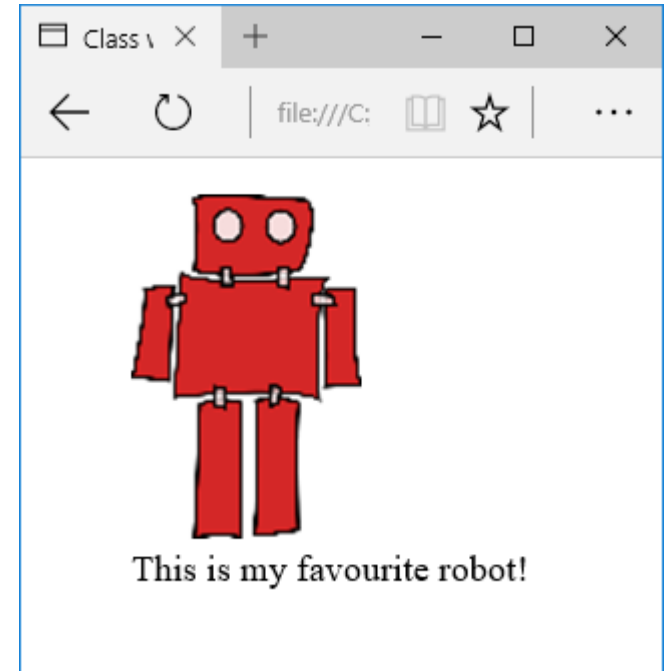
Captioning figures

Example:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Class work</title>
    <meta charset="UTF-8">
  </head>

  <body>
    <figure>
      

      <figcaption>
        This is my favourite robot!
      </figcaption>
    </figure>
  </body>
</html>
```



HTML Elements: Embedding Videos

Videos in HTML

Prior to HTML5, there was no standard mechanism to play videos on web pages.

The preferred way was to include a plug-in (such as flash) and use it to play videos.

HTML5 includes a **<video>** element that lets you embed and play videos.

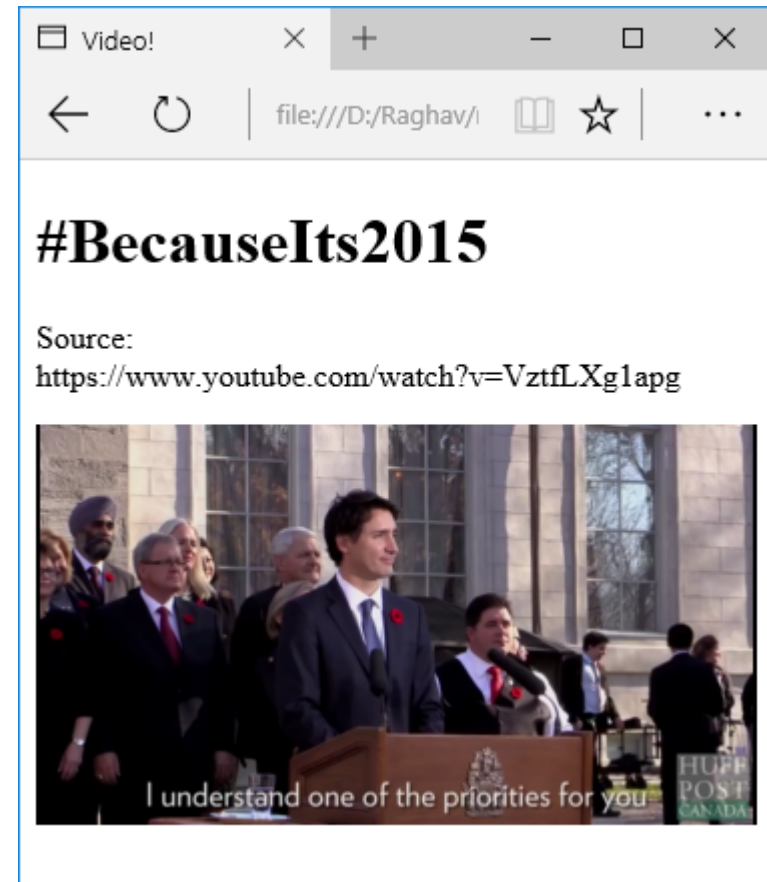
As with images, it is always a good idea to specify the dimensions (width and height) for videos.

You may include text between **<video>** and **</video>**, which will only be displayed if the browser does not support HTML5 videos.

Videos in HTML

Example:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Class work</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>#BecauseIts2015</h1>
    <p>Source:<br />
      https://www.youtube.com/
      watch?v=VztfLXg1apg</p>
    <video src="files/2015.mp4"
      style="width:360px; height:200px;">
      <p>Video not supported!</p>
    </video>
  </body>
</html>
```



There are no video controls here – difficult to play / pause / control volume of the video!

Videos in HTML

Introducing the ***controls*** attribute → an easy way to enable video controls.

You can also set it up so that the video starts playing as soon as the page loads:

Not using the ***controls*** attribute, but using the ***autoplay*** attribute.

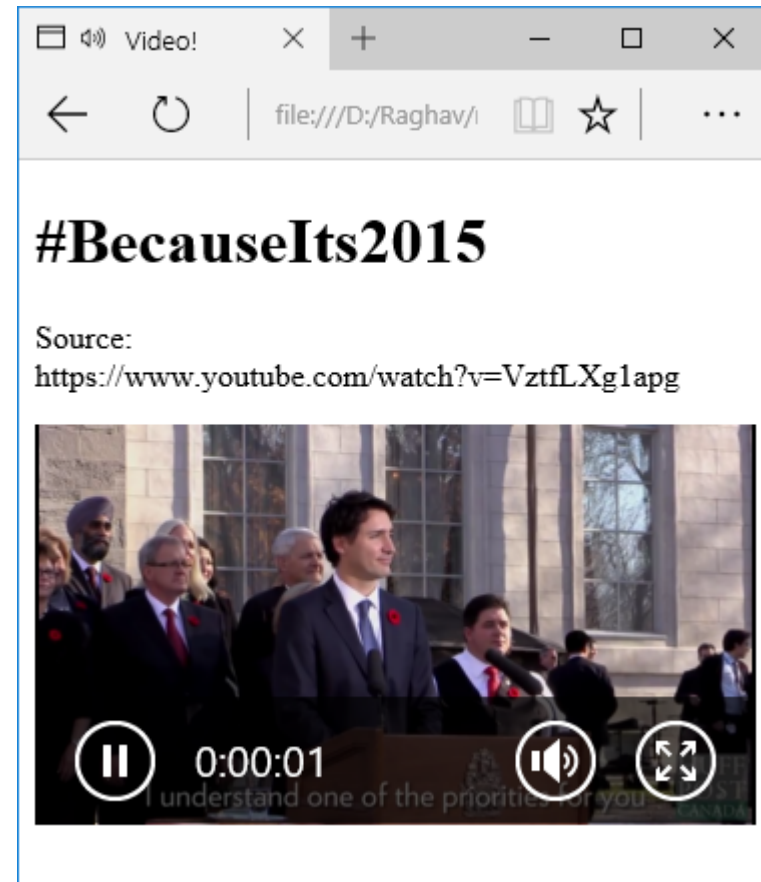
- *Some browsers (e.g. Safari in macOS High Sierra, 2017) will **disable autoplay** by default.*

You can use both ***controls*** and ***autoplay*** in a video element. This will help you in making sure that the videos start playing automatically, but users can also stop it if they don't want to view the video.

Videos in HTML

Example:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Class work</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>#BecauseIts2015</h1>
    <p>Source:<br />
      https://www.youtube.com/
      watch?v=VztfLXg1apg</p>
    <video src="files/2015.mp4"
      style="width:360px; height:200px;"
      controls autoplay>
      <p>Video not supported!</p>
    </video>
  </body>
</html>
```



You can include controls using JavaScript – we will discuss more about this later.

HTML Elements: Embedding Audio

Audio in HTML

Similar to videos, there was no standard mechanism to play audio files on web pages prior to HTML5. The preferred way was to include a plug-in (such as flash).

HTML5 includes an **<audio>** element that lets you embed and play audio files.

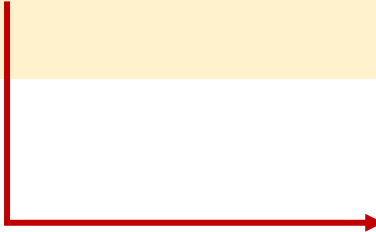
You may include text between **<audio>** and **</audio>**, which will only be displayed if the browser does not support the HTML5 audio element.

Similar to `<video>`, you can also use the ***controls*** and ***autoplay*** attributes in `<audio>` as well.

Audio in HTML

The suggested syntax to include audio files in HTML is as follows:

```
<audio id="preferred_id">  
  <source src="filename.mp3" type="audio/mpeg">  
  <source src="filename.ogg" type="audio/ogg">  
  <source src="filename.wav" type="audio/wav">  
</audio>
```



*If you have multiple sources
(with different formats),
the browser will choose
whatever format it supports.*

Media types:

MP3: type="audio/mpeg"

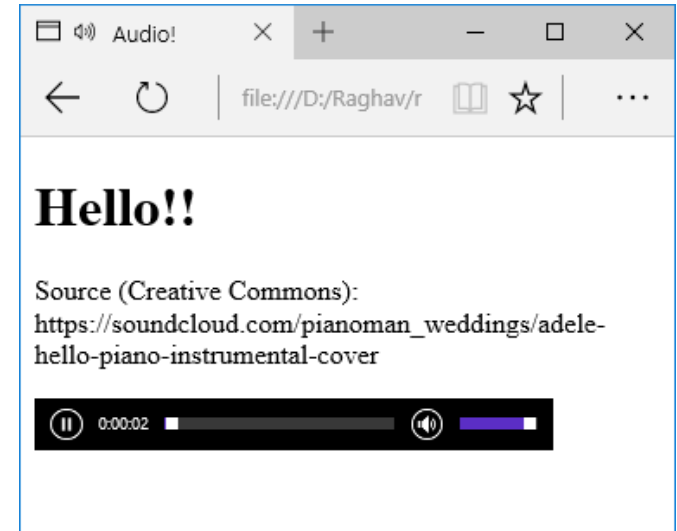
OGG: type="audio/ogg" → open container format

WAV: type="audio/wav"

Audio in HTML

Example (with controls):

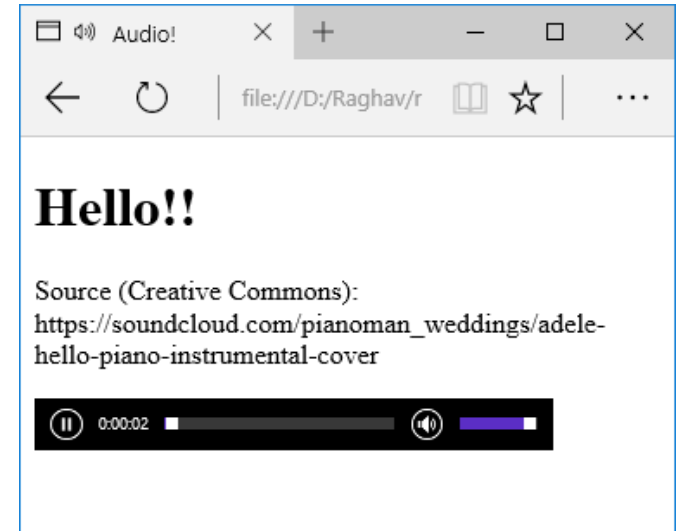
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Class work</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Hello!!</h1>
    <p>Source (Creative Commons):<br />
    https://soundcloud.com/pianoman_weddings/adele-
    /adele-hello-piano-instrumental-cover</p>
    <audio id="myAudio" controls>
      <source src="files/hello.mp3"
        type="audio/mpeg">
      <p>Audio not supported!</p>
    </audio>
  </body>
</html>
```



Audio in HTML

Example (with controls and autoplay):

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Class work</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Hello!!</h1>
    <p>Source (Creative Commons):<br />
    https://soundcloud.com/pianoman_weddings/adele-
    /adele-hello-piano-instrumental-cover</p>
    <audio id="myAudio" controls autoplay>
      <source src="files/hello.mp3"
        type="audio/mpeg">
    <p>Audio not supported!</p>
  </audio>
</body>
</html>
```



HTML Links (Cont'd)

A note on links & empty links

When you create Anchor links, you should always specify a value to HREF, as discussed previously.

However, there might be situations where you start developing a website, but don't know the pages that exist, or, you have not yet programmed some of the pages.

In such situations, you can create what are known as *empty links*.

An empty link contains all other attributes of the anchor element, but it does not have any assigned value for the HREF. It is there as a *placeholder link*, i.e. it is placed there for future use.

A note on links & empty links

Using empty links:

```
<a href="">This is an empty link</a>
```

```
<a href="" type="_blank">This is an empty link</a>
```



Some people also like to use the “hashtag” within the quotes, when creating empty links. ***This is not recommended.***

Example:

```
<a href="#">This is an empty link</a>
```



Why? Because it creates an extra entry in the browser’s history. Therefore, it is not recommended.

The Link Element

The **<link>** element (empty element):

```
<link href="resource_url" rel="resource_type" type="content_type">
```

*URL of the resource;
can be absolute or
relative path*

*Type of resource being linked;
e.g. "stylesheet", "icon"*

*MIME type, indicating
content type.
e.g. "text/css"*

The Link Element

The **<link>** element (empty element): Examples

Including stylesheets:

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

Including favicons:

```
<link href="favicon.png" rel="icon" type="image/png">
```

Or

```
<link href="favicon.png" rel="icon">
```

The Link Element

The **<link>** element (empty element): Examples

But, when using favicons, you've got to be mindful of Internet Explorer version 9 and earlier. To account for this, use a **.ico** file instead of **.png**, and use the following syntax:

```
<!--[If IE]>  
<link href="favicon.ico" rel="shortcut icon" >  
<![endif]-->
```

`<!-- [If IE]> ... <![endif]-->` is recognized as a conditional statement by Internet Explorer (version 9 and earlier).

CSS Selectors - 1

Syntax of CSS Rules: Recap

Selector.

Indicates the element to which the style rule applies.

selector

Property.

Indicates the aspects of the element that you intend to style or change.

property1: **value1;**

property2: **value2;**

/ this is a comment */*

}

Declaration.

Indicates the way in which the elements in the selector should be styled. Each declaration has a **style property** and a **value**, separated by a colon.

Value.

Indicates the settings for the chosen property.

CSS Selectors: Recap

Selectors help identify the HTML element in the DOM to which the style rule applies.

Types of selectors.

- *Tag names*

You can directly use the names of the HTML tags.

E.g. **body**, **p**, **a**, **ol**, etc.

- *Tag attributes*

You can also use tag attributes to apply styles.

Two attributes are commonly used – **class** and **id**.

```
selector
{
    property1: value1;
    property2: value2;
    /* A comment */
}
```

CSS Selectors

Types of selectors.

- *class*

Identifies the style “class” to which an element belongs.
Values to the HTML class attribute are ***space-separated***.

E.g. `<div class="c1 c2"></div>`

Multiple HTML elements may have the same class value.

In stylesheets, include a full-stop (or, period) just before the class name, if you use it as a selector.

E.g. `<style>`

`.c1 { font-style: italic; }`

`.c2 { font-size: 125%; }`

`</style>`

CSS Selectors

Types of selectors.

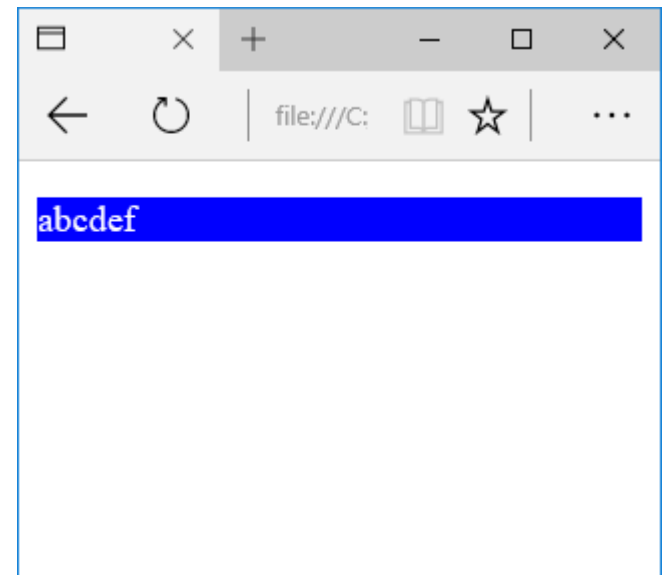
- **class**

Identifies the style “class” to which an element belongs.
Values to the HTML class attribute are ***space-separated***.

Styles are applied sequentially.
(c1 first, c2 next and so on, in example below).

```
<style>
  .c1 { background-color: red; }
  .c2 { background-color: blue; color: white; }
</style>
```

```
<p class="c1 c2">abcdef</p>
```



CSS Selectors (cont'd)

Types of selectors.

- *id*

You may also use the element's ID to apply styles individually to an element. (Remember, IDs are unique to an element)

E.g. `<div id="myDiv"></div>`

In stylesheets, include a hash sign (#) just before the ID, if you use it as a selector.

E.g. `<style>`

```
  #myDiv { font-style: italic; }  
</style>
```

CSS Selectors (cont'd)

Types of selectors.

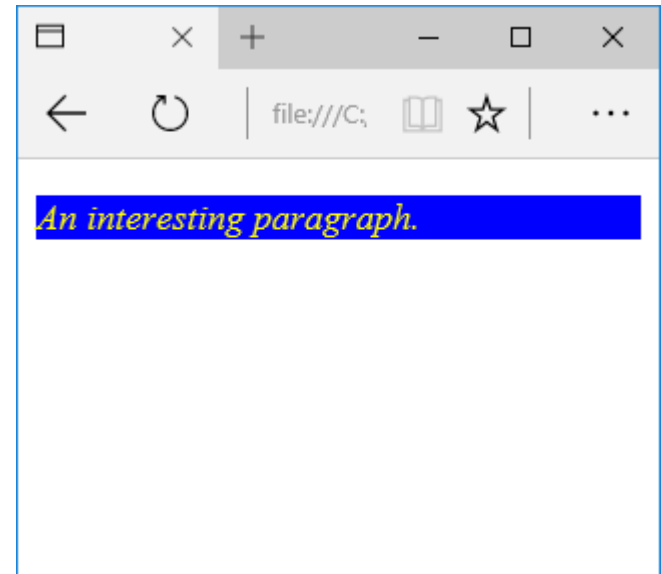
- *id*

You may also use the element's ID to apply styles individually to an element. (Remember, IDs are unique to an element)

If you choose to, you may also include style classes in addition to styling the element's ID.

```
<style>
  #myParagraph { font-style: italic; color: yellow; }
  .c1 { background-color: red; }
  .c2 { background-color: blue; color: white; }
</style>
```

```
<p id="myParagraph" class="c1 c2">An interesting
paragraph.</p>
```



CSS Selectors (cont'd)

Types of selectors.

- *Pseudo-classes*

This selector selects a ***special-state*** of the HTML element.

E.g. **hover** will apply the given style when the user “hovers” (or, moves the mouse on a given element)

In stylesheets, a colon will separate the element name and the pseudo-class.

Syntax:

```
selector:pseudo-class { property: value; }
```

More on Pseudo-classes and other selectors/element relationships on Wednesday.