

# HYPertext Markup Language



**Navigation, Lists, Images, Iframes, AND Inputs**

COS216  
AVINASH SINGH  
DEPARTMENT OF COMPUTER SCIENCE  
UNIVERSITY OF PRETORIA

# HTML - ATTRIBUTES

- Some attributes exists for all HTML objects
- Often used by Document Object Model (DOM) manipulation to access objects through JavaScript
- Identifier (`id`)
  - A unique ID to disguise one element from the other
- Class (`class`)
  - Multiple elements can belong to the same class, one element can belong to multiple classes
  - Can be used by DOM or to assign the same CSS to all elements belonging to a class

```
<div id="box1" class="redbox thickborder"></div>
<div id="box2" class="redbox thinborder"></div>
```

# HTML - NAVIGATION

- Various objects exist to structure your page and handle navigation:
  - `nav`: A navigation bar with links to other parts of your site
  - `header`: A header at the top of your page which may contain a logo and navigation
  - `footer`: A footer at the bottom of the page which may contain the copyright and year
- Typical structure of a website:



# HTML - LINKS

- Add a link to another page
- Links are the basis on which HTML was built
- If a link is clicked, navigates to that page
- Most browsers display links in blue color and underlined

```
<a href="https://coinmarketcap.com">View Crypto Prices</a>
```

```
<a href="subdirectory/file.html">View Subpage</a>
```

- The href attribute specifies an absolute URL or relative path
- The inner text is displayed on the site and is clickable

# HTML - LISTS

- Add lists with list items (li)
- Ordered list (numbered)
  - ol element
  - type attribute: specifies the number/letters to use for the list (1, A, a, I, i)
  - start attribute: specifies at which number to start
- Unordered list (bullet points)
  - ul element
  - type attribute: specifies the bullet type (disc, square, circle) not supported in HTML5 anymore

# HTML - LISTS

```
<ol type="A">
    <li>Bitcoin</li>
    <li>Ethereum</li>
    <li>Cardano</li>
</ol>
```

- A. Bitcoin
- B. Ethereum
- C. Cardano

```
<ul>
    <li>Bitcoin</li>
    <li>Ethereum</li>
    <li>Cardano</li>
</ul>
```

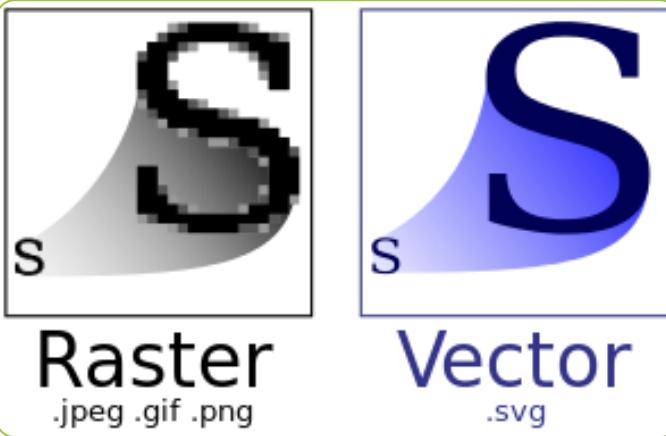
- Bitcoin
- Ethereum
- Cardano

# HTML - IMAGES

- Add images to websites
- Image formats depends on browser support
- Choose format according to support and your requirements

Wide Support	Limited Support
JPG	SVG
PNG	APNG
GIF	
BMP	
TIFF	

# HTML - SVG



- Scalable Vector Graphics (SVG)
- XML based vector image format
- Images are not stored as raster pixels, but as vectors
- Vectors are calculated mathematically, zoom in/out without loosing quality
- File sizes are a lot smaller than rasters
- Can manipulate SVG through CSS/JS (eg: color, shadows, point width, etc)
- Can only store vectors, not pixels, therefore not suited for photos

- Always try to use JPG or PNG

# HTML - IMAGES

Format	Compressed	Transparency	Animation	Use Case
<b>JPG</b>	Yes	No	No	Use for larger images, artwork, and photos.
<b>PNG</b>	Yes	Yes	Yes (APNG)	Use for discrete graphics, icons and comics.
<b>GIF</b>	Yes	Yes (limited)	Yes	Try to avoid. De facto standard for animations. However, the image quality is poor. Try to use APNG for animations.
<b>BMP</b>	No	No	No	Avoid at all cost
<b>TIFF</b>	No	No	No	Avoid at all cost
<b>SVG</b>	Yes (vector)	Yes (vector)	Yes (limited)	For high quality resolution with small file size. Some browsers have limited SVG support (eg: animation). SVGs can also be manipulated from CSS (eg: color, borders, shadows).

# HTML - IMAGES

- Does not need a closing tag

```

```

- Attributes:
  - Source (src) the source URL of the image (relative or absolute)
  - Alternative Text (alt): Text to display if the image cannot be loaded
  - Size (width & height): The size of the image. Images are resized by browser to this size.  
If no size is provided, uses the resolution of the image
  - Other less used attributes available

# HTML - IMAGES

- Image source can be an absolute URL

```

```

- Image source can be a relative path
  - Image is in the same directory as the current HTML file  

```

```
  - Image is in a parent directory of the current HTML file  

```

```
  - Image is in a subdirectory of the current HTML file  

```

```

# HTML - IMAGES

- Image source can also be Base64
- Directly embedded in the HTML page
  - Does not require an additional file to be retrieved
- Base64 is about 30% larger than the binary data
  - Only use for small images, such as icons or logos

```

```

# HTML - IFRAMES

- Displays an HTML document inside another HTML document
- Can be a document from the same or an external site
- Basic asynchronous call
  - Frame is loaded in the background
  - Separate from main site
  - First version of AJAX (discussed later in the semester)

```
<iframe src="demo.html" height="200" width="300"></iframe>
```

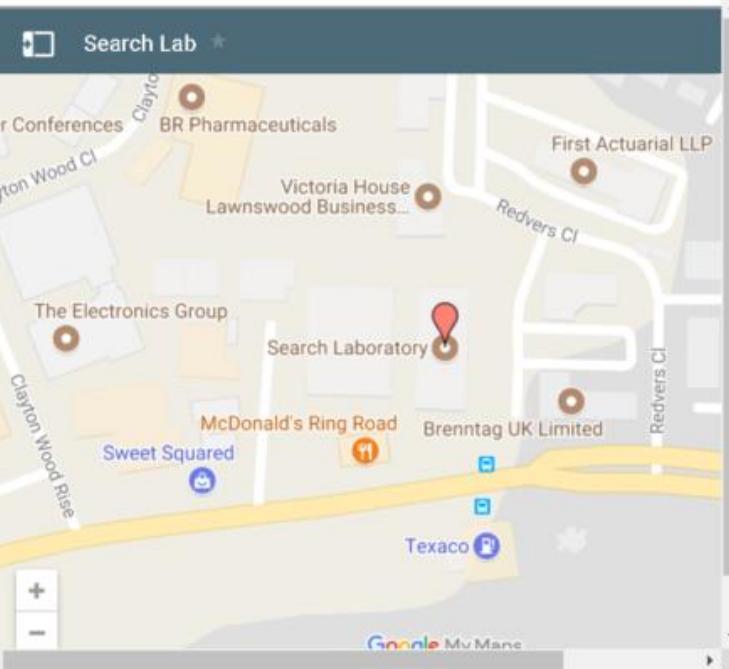
# HTML - IFRAMES

- Attributes are similar to images
  - Source (`src`) the source URL of the iframe
  - Size (`width & height`): The size of the iframe
- The source can be:
  - Absolute URL to own or external site
  - Relative path to other HTML document on same site

# HTML - IFRAMES

Master page

INSERT BIO HERE



Elements Console Sources Network Performance Memory 9

**INSERT BIO HERE**

**Iframed page 1**

```
<p>
<strong>INSERT BIO HERE</strong>
</p>
<div>
<iframe src="http://.../maps/gmap1.html" style="background: url('.../wp-content/uploads/2017/05/loading.gif') center no-repeat; border:0px;" width="100%" height="480px">
</div>
```

**Iframed page 2**

```
<div>
<iframe src="https://www.google.com/maps/d/u/0/embed?mid=16yn1JFh8LjG8blytvBX02dhs3d0" width="640" height="480">
</div>
```

# HTML - IFRAMES

- Google website

**COS216 Frames**

# HTML - CORS

- Google iframe does not load
- Error message in browser console:
  - *Refused to display 'https://www.google.com' in a frame because it set 'X-Frame-Options' to 'sameorigin'*
- Cross-Origin Resource Sharing (CORS)
  - Additional HTTP headers are set in requests
  - Based on the HTTP headers, the server can reject access to resources if the request has a different origin (domain) than the destination (server domain)
  - Used to eliminate abuse by other websites using your resources (HTML, JS, images, etc)
  - Apache can be configured (or .htaccess file) to allow any/some/none requests from external domains

# HTML - INPUTS

- HTML supports a range of different input elements
- Used to get input from the user
- HTML only provides the input elements
  - HTML cannot do anything with the input
  - JavaScript is needed to read the input and do something with it
  - JavaScript has events (text changed, mouse click, etc) to help with this
  - JavaScript events can be specified as a HTML attribute

# **HTML - INPUTS**

- Traditional HTML inputs:
  - Buttons
  - Text Fields
  - Text Areas
  - Password
  - File
  - Image
  - Checkboxes
  - Radio Buttons
  - Hidden

# HTML - INPUTS

- New HTML inputs:
  - Color
  - Date
  - Date Time
  - Time
  - Month
  - Week
  - Email
  - Range
  - Search
  - Number
  - Url

# HTML - INPUTS

- There are two categories of inputs
  - Using separate elements: `textarea`, `button`
  - Using the `input` element, typically part of a form: the remainder
- Many of these inputs are very similar, but have some extra functionality
  - Eg: `password` is a text field obfuscated with dots or asterisks
  - Eg: `url` is a text field that verifies that the input has a valid URL format
  - Eg: `submit` is a button for submitting a form
  - Eg: `image` is a button that is styled with an image

# HTML - FORMS

- A form is a grouping of input elements
- Data from a form is typically send to the server
  - PHP on the server then reads/uses these values
  - The form specifies the destination PHP script (action)
  - The form specifies the HTTP method, such as GET or POST (method)

# HTML - FORMS

- Example: login form has 3 inputs
  - Username
  - Password
  - Submit button

```
<form action="page.php" method="post">
    Username: <input type="text" name="user"><br>
    Password: <input type="password" name="pass"><br>
    <input type="submit" value="Login">
</form>
```

Username:

Password:

# HTML - BUTTON

- Standard button

```
<button type="button">Click Me!</button>
```

Click Me!

- Submit button for a form

```
<input type="submit" value="Click Me!">
```

Click Me!

- Button with an image

```
<input type="image" src="submit.png" alt="Submit" width="48" height="48">
```



# HTML - TEXT

- Text input field

```
<input type="text" name="username">
```

username

- Obfuscated password field

```
<input type="password" name="pass">
```

.....

- Text area for paragraphs

```
<textarea rows="4" cols="50">Some long paragraph</textarea>
```

Some long paragraph

# HTML - HIDDEN

- Hidden text input field
- Hidden to the user
- Can be used to add control data that is submitted to the server

```
<input type="hidden" name="hiddenvalues">
```

# HTML – CHECK BOXES

- A group of check boxes
- Multiple boxes can be selected at the same time
- The name attribute specifies the group the button belongs to

```
<input type="checkbox" name="btc" value="btc"> Bitcoin Price<br/>
<input type="checkbox" name="eth" value="eth" checked> Ethereum Price<br/>
<input type="checkbox" name="ada" value="ada" checked> Cardano Price<br/>
```

- Bitcoin Price
- Ethereum Price
- Cardano Price

# HTML – RADIO BUTTONS

- A group of radio buttons
- Only a single button can be selected at the same time
- The name attribute specifies the group the button belongs to

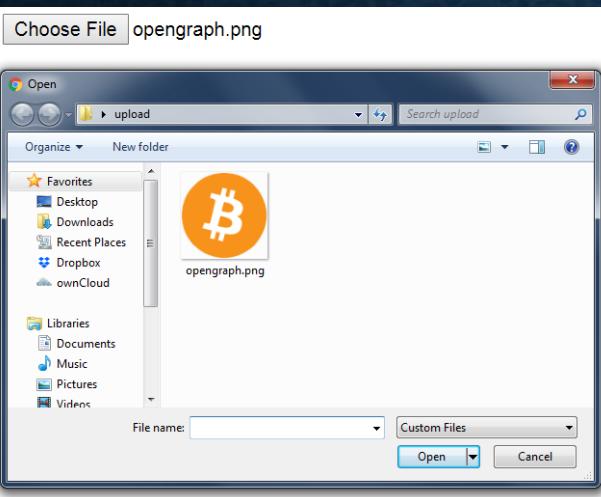
```
<input type="radio" name="prices" value="btc"> Bitcoin Price<br/>
<input type="radio" name="prices" value="eth" checked> Ethereum Price<br/>
<input type="radio" name="prices" value="ada"> Cardano Price<br/>
```

- Bitcoin Price
- Ethereum Price
- Cardano Price

# HTML – FILES

- Upload files to the server
- Always requires a button press and selection dialog
  - For security reasons, otherwise websites can access your personal files without permission
- Can restrict the selected file type (eg: only images)

```
<input type="file" name="profilepic" accept=".jpg, .jpeg, .png">
```



```
<stop stop-color="#06101F" offset="0%>
<stop stop-color="#1D304B" offset="100%>
</linearGradient>
</defs>
<rect width="800" height="450" rx="8" fill="url(#gradient)" y="500"/>
</svg>
<div class="media-control">
<svg width="96" height="96" viewBox="0 0 96 96" fill="none" stroke="white" stroke-width="2px">
<defs>
<linearGradient x1="87.565%" y1="15.875%" x2="15.875%" y2="87.565%">
<stop stop-color="#FFF" stop-opacity="0" offset="0%>
<stop stop-color="#FFF" offset="100%>
</linearGradient>
<rect width="500%" y="-500%" fill="SourceLight" stroke="white" stroke-width="2px" style="filter: drop-shadow(0 0 10px black);"/>
</defs>
<path d="M 48 48 L 48 52 L 52 52 Z" fill="white" stroke="white" stroke-width="2px" style="filter: drop-shadow(0 0 10px black);"/>
<path d="M 48 48 L 48 52 L 52 52 Z" fill="white" stroke="white" stroke-width="2px" style="filter: drop-shadow(0 0 10px black);"/>
</svg>
</div>
```

# QUESTIONS