

# COMP 1950

Web Development and Design 2

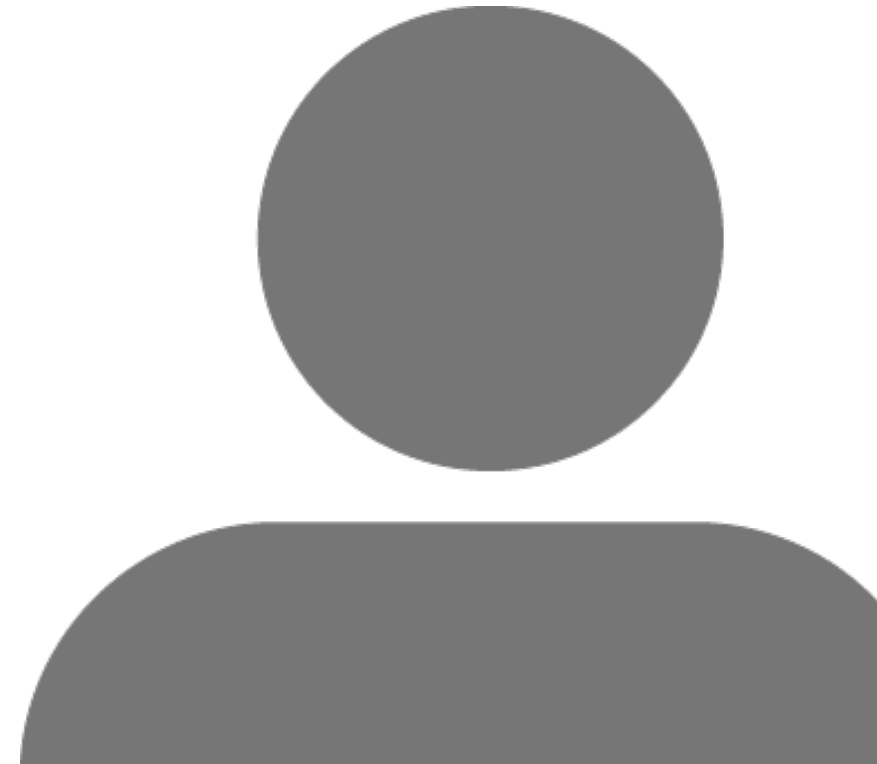
Day 01

# Agenda

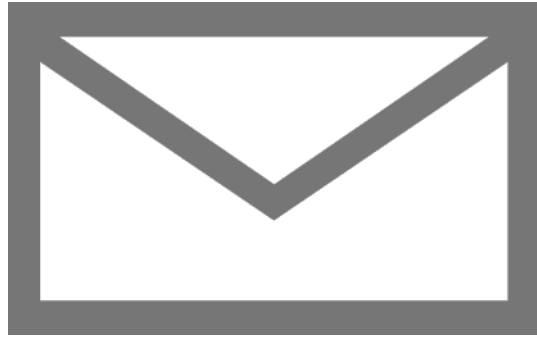
- Introductions
- Course Information & Administration
- Development setup
- HTML and CSS Review
- HTML5 Audio and Video
- CSS Filters

# About Me – Michael Whyte

- Working in the media field for 23 years
- Started in Multi-Media for live events
- Transitioned to web development in 2009
- Instructor at BCIT
  - High Tech Professional Programs
    - Technical Web Designer Program – BCIT
    - Software Systems Developer Program - BCIT
  - Part-Time Studies BCIT
- Me on the web
  - Web site: [michaelwhyte.ca](http://michaelwhyte.ca)
  - LinkedIn: [linkedin.com/in/michaelwhyte](https://www.linkedin.com/in/michaelwhyte)
  - Twitter: [twitter.com/michaelwhyte](https://twitter.com/michaelwhyte)
  - Flickr: [flickr.com/photos/mwhyte](https://www.flickr.com/photos/mwhyte)
  - Google Plus: [plus.google.com/+MichaelWhyte01](https://plus.google.com/+MichaelWhyte01)
  - GitHub: <https://github.com/michaelwhyte>
  - CodePen: <https://codepen.io/michaelwhyte>



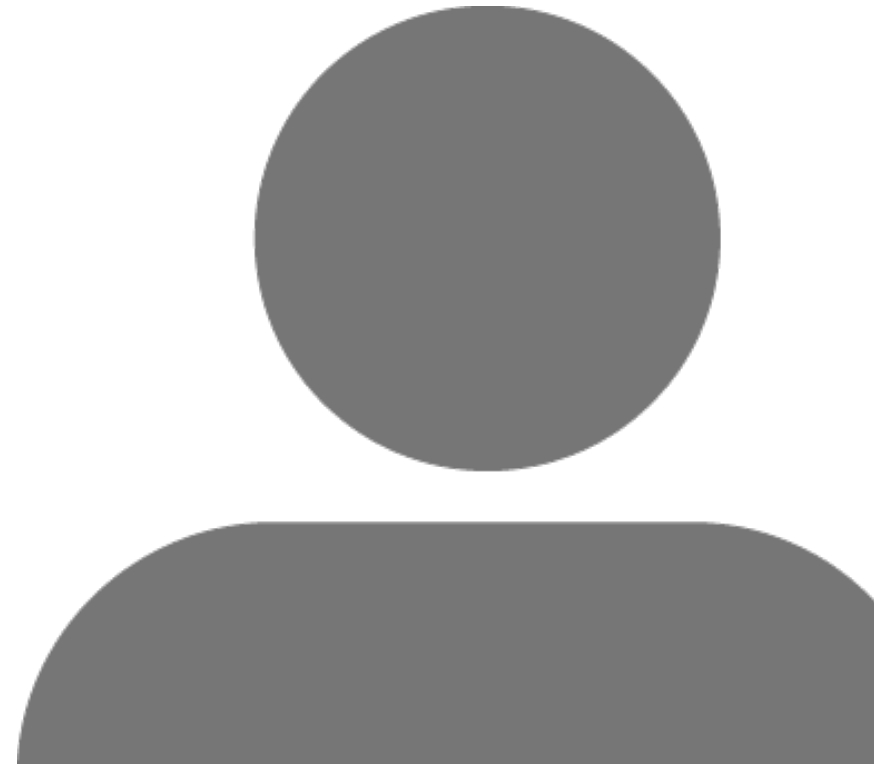
# Contact Me – Michael Whyte



[Michael\\_Whyte@bcit.ca](mailto:Michael_Whyte@bcit.ca)

# All About You

- Your name (preferred nickname if applicable)
- Your background
- What do you hope to achieve in this course? (Be specific!)
- Anything you wish to share about yourself



# About the course

- **Objectives:**
  - Provides a mid-level overview of web development and design for students who already understand HTML and CSS
  - Focus on responsive mobile web development
  - Explore modern CSS layout techniques
- **Format:**
  - Some theory, mostly practical, hands on
  - In-class exercises
  - Lab time
  - Various homework exercises, 2 major projects, 2 exams
- **Readings:**
  - From provided PowerPoint slides
  - Occasionally links will be provided to additional online resources
- **Assignments and Projects:**
  - Homework assignments - small exercises to be handed in throughout course
  - Two major projects - a fully-functioning mobile responsive web site and a mobile responsive web site created as part of a group
- **Exams**
  - Midterm exam on week 6 and final exam on week 12

# Daily Schedule

- Each class is 3 hours
- Most classes will be divided up as follows
  - 0.5 hours of review from previous week
  - 2.25 hours of lecture
    - New material
    - In-class exercises
  - 15 minute break (approximately mid way through the class)
  - 0.5 hours of lab time (some classes)
    - Work on homework assignment
    - One on One Q & A



# What you will learn

- Create multi-page web sites using modern HTML and CSS
- Layout web sites using modern CSS layout technologies such as flexbox and CSS Grid
- Add custom fonts to a web page
- Create mobile responsive web sites
- Animate elements in a web page
- Write CSS using SASS
- Develop projects as part of a team
- Version control your code using Git and GitHub
- Add interactive elements to your web site using jQuery
- Optimize a web server using a .htaccess files

# What you won't learn

- Graphic design
- Multimedia production (video, sound, dynamic graphics)
- database integration ([COMP1630](#))
- Java, or other web programming languages ([COMP1409](#))
  - We will have a brief lessons on JavaScript and jQuery and a brief summary of PHP

# Policy on Plagiarism

- Plagiarism is using another's work and claiming it as your own
- Do not use freely available website templates for submission of course projects and assignments

# Plagiarism

- Acceptable use:
  - Using code snippets from Stackoverflow, Codepen or other code demos for small parts of your web projects
    - a good courtesy is to give credit to your source in your code
- Not Acceptable
  - Grabbing a free website template from the internet, changing a few words and colours and claiming it as your own

# Plagiarism

- If in doubt, ask your instructor

# Plagiarism

- All projects and assignment submissions that were plagiarized will receive a grade of zero for that project or assignment and will be reported to BCIT

# D2L and Course Materials

- All course materials (demo files, PowerPoints, links, images) is available on BCIT's D2L (Desire to Learn)
- You can access D2L by going to the BCIT web page and clicking the "D2L" link in the top navigation bar or going directly to:
  - <https://learn.bcit.ca>
  - You log in with the same credentials you use to log in to the BCIT computers
  - Once logged in, the COMP 1950 course should be listed under the "My Courses" section
    - Click the link for the COMP 1950 course to access all the COMP 1950 course material

# Additional Learning Materials

- There is no textbook for COMP 1950
- Many additional free online resources for learning about web development
- Web sites
  - <https://css-tricks.com/>
  - <https://developer.mozilla.org>
- YouTube
  - Great place for free video learning on a variety of web development topics
  - It's free, so quality can vary
  - Here are some good channels for learning:
    - [Kevin Powell](#)
    - [The Net Ninja](#)
    - [Dev Tips](#)



# Your Development Environment

- Hardware
  - Laptop or Desktop computer
- Software
  - COMP 1950 software development tools are all freely available for download online. Be sure to configure your home computer or laptop with the following tools:
    - Web browsers
    - Text editor
    - FTP client
    - Graphics Editing Software
    - WAMP (Windows) or MAMP (Windows or Mac) web server software
    - Git version control system (optionally GitHub Desktop application)
    - Ruby (Windows only, on Macs Ruby is pre-installed)

# Hardware

- A Mac or Windows PC is fine for web development. Even Linux will work.
  - The beauty of the Web is it is largely platform agnostic
- Hardware Specs
  - Ideal
    - Processor – Core i5 or better
    - RAM – 8 GB or more (get as much as you can afford)
    - Display – 1080p or higher resolution
    - Hard Drive – SSD – 256 GB or more
  - You can get by with a lower spec machine but productivity and the ability to test web sites at different resolutions may suffer

# Web Browsers

- The most popular browser used by Web Developers for developing web sites is Google's Chrome
  - Excellent Developer Tools
- Web Developers should install as many different browsers on their system as possible for testing sites in different browsers. Most are free. Some common browsers to install are:
  - Edge (default browser on Windows 10)
    - Only available on Windows
  - Safari (default browser on Macs)
    - Only available on Macs
  - Chrome
  - Firefox
  - Opera

# Web Browser Links

- BCIT Computers already have Chrome and Firefox installed
- To install Chrome on your computer visit:
  - <https://www.google.com/chrome/index.html>
- To install Firefox on your computer visit:
  - <https://www.mozilla.org/en-US/firefox/>

# Browser Web Development Extensions

- Both Firefox and Chrome allow you to install extensions that extend the capabilities of the browsers
- Both Firefox and Chrome have many extensions that help with web development
- See the following slides for some web developer extensions for Firefox and Chrome
- For Chrome extensions visit this page in the Chrome web browser:
  - <https://chrome.google.com/webstore/category/extensions?hl=en-US>
- For Firefox extensions visit this page in the Firefox web browser:
  - <https://addons.mozilla.org/en-US/firefox/>

# Web Development Firefox Extensions

- [ColorZilla](#)
- [MeasureIt](#)
- [FontFinder](#)
- [Web Developer](#)

# Web Development Chrome Extensions

- [ColorZilla](#)
- [Page Ruler](#)
- [Web Developer](#)
- [WhatFont](#)

# Browser Web Developer Tools

- All the major web browsers include web development tools accessed by various methods that differ depending on the browser (usually something similar to a "right click" then selecting "inspect" will work).
  - Microsoft Edge and Apple Safari's developer tools must first be enabled by the user
- Browser developer tools allow you to view HTML source code, inspect applied CSS rules and troubleshoot JavaScript
- CSS rules can be temporarily turned off or edited for testing. These are only temporary changes and do not affect your local CSS files
- Chrome and Firefox have extra tools for developing mobile responsive web sites



# Text Editors

- Text editors are used to write the HTML / CSS / JS / PHP code that is used on web sites
- You can use plain text editors such as Notepad but you will not be as productive than if you use a dedicated text editor for web development
- Text editors improve a web developers productivity by providing features such as:
  - Syntax highlighting (highlighting special code characters in different colors to allow for easier editing and reading of code)
  - Code completion

# Text Editors

- Text editors are a personal choice. Pick the one that works best for you.
- For COMP 1950 we will use the free Visual Studio Code text editor
  - You can download it for free at:
    - <https://code.visualstudio.com/>
  - The in-class computers have it pre-installed

# Text Editors

- Other Text Editors:
  - Notepad++
  - Visual Studio Code
  - Brackets
  - Sublime Text (paid – but has an unlimited free trial)
  - PHP Storm (paid)
  - Web Storm (paid)
  - Dreamweaver (available with an Adobe Creative Cloud subscription)

# More on Visual Studio Code...

- Visual Studio Code like many text editors can have its features extended by installing extensions
- You can install extensions by opening Visual Studio Code and pressing "Ctrl+Shift+X" (Windows) or "Cmd+Shift+X" (Mac)
- Once the extension manager window is open you can simply search for an extension to install it
- Use the search bar to search for an extension

# Visual Studio Code Recommended Extensions

- There are hundreds of Visual Studio Code extensions. Below are some suggested extensions that we recommend you install on your personal installation of Visual Studio Code (the BCIT computers already have most of these extensions installed)
- Suggested Extensions
  - Live Server
    - Creates a simple web server for serving HTML files
  - open in browser
    - Allows you to right click and open HTML files in a browser
  - Path Autocomplete
    - Completes paths when typing in links to files in HTML files
  - htmltagwrap
    - Wraps a text selection in an HTML tag by selecting text and pressing “Alt+W”
  - SCSS IntelliSense
    - Improves code completion for Sass variables, mixins and functions

# Visual Studio Code - Settings

- To enhance the usability of Visual Studio Code we recommend setting some custom settings (BCIT Computers already have these settings applied)
- You set custom settings in Visual Studio Code by editing a settings JSON file (a type of plain text file).
- You can access this file by going to the menu bar and clicking File->Preferences->Settings
- Once on the Settings screen go to the right most pane and enter your custom settings in between the two "{}"
- See the next slide for the settings to apply

# Visual Studio Code Settings

- Add the text below in between the two "{}" in the right most pane of the settings screen. Include the quotes and the coma

```
"emmet.triggerExpansionOnTab": true,  
"path-autocomplete.extensionOnImport": true
```

# Emmet with Visual Studio Code

- Emmet comes pre-installed with Visual Studio Code
- Emmet is an extension available for most text editors that allows the speeding up of the writing of HTML and CSS by typing abbreviations and then pressing the tab key to create HTML tags or CSS properties
- For example, to create a "p" tag you would normally type: "<p></p>" for a total of 7 key strokes. With Emmet you simply press "p" then the "tab" key and Visual Studio code will create the HTML "p" tag code for you.
- There is a lot more Emmet can do to improve your code editing productivity. To learn more about Emmet click the link below:
  - <https://emmet.io/>



# FTP Clients

- FTP stands for:
  - File Transfer Protocol
  - In plain English
    - It is a standard for transferring files from one computer to another computer
- FTP Clients are graphical user interfaces (GUIs) used for connecting two computers and either sending files to a computer (uploading) or getting files from a computer (downloading)
- In web development we use FTP to deploy our files (upload) to a web server from our local machine
- Once the files are on a web server they can be accessed directly by navigating to them in a web browser

# FTP Clients

- For COMP 1950 we will use the FileZilla FTP Client
- Free software
- Pre-installed on COMP 1950 BCIT computers
- To download to your own machine click the link below. Make sure to click on the "Client" software link and not the "Server" software link
  - <https://filezilla-project.org/>

# Your Web Space for COMP 1950

- Each student will get their own web space for the duration of the program.
- The web space will be deleted without notice at the end of the course.
  - \*\*\*Do not make a copy of a file on the web space your only copy. Keep local copies of all your work.
- To access your web space from a browser navigate to the following URL.  
Change the text in [] to your relevant information
  - [https://comp1950.michaelwhyte.ca/\[the-first-part-of-your-my-bcit-email\]](https://comp1950.michaelwhyte.ca/[the-first-part-of-your-my-bcit-email])
  - For example, for a student with a my.bcit.ca email address of jsmith@my.bcit.ca their URL would be:
    - <https://comp1950.michaelwhyte.ca/jsmith/>

# Your Web Space for COMP 1950

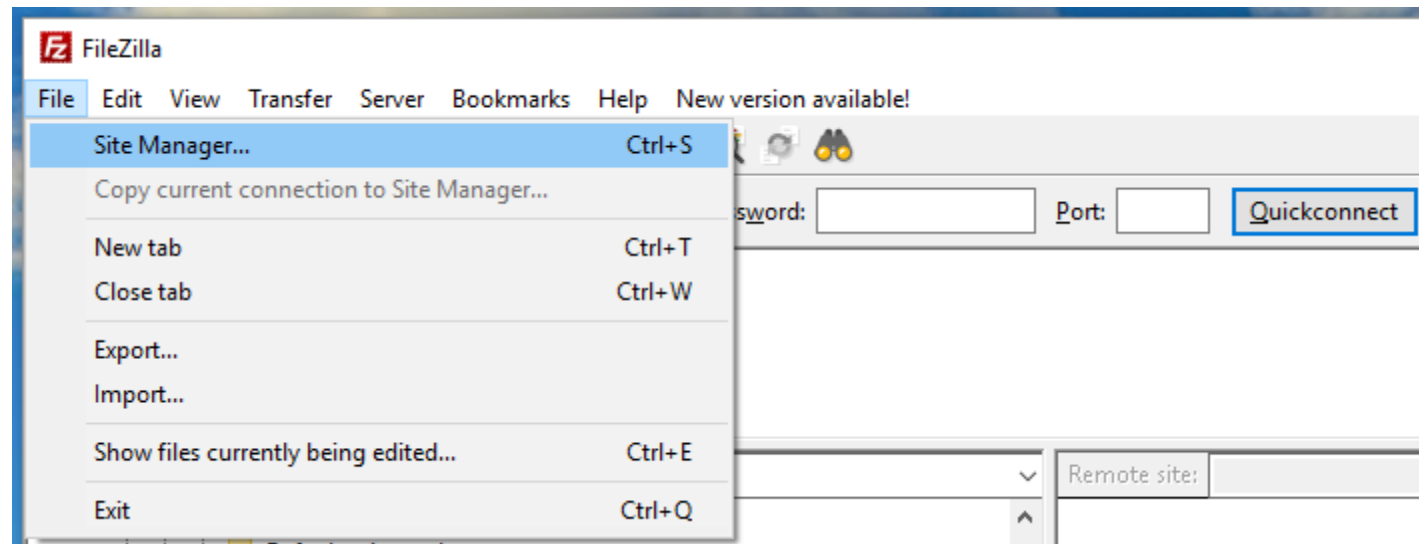
- This space is public. Anyone on the web can see it.
- Do not put private files on your personal web space.
- Do not put files on your personal web space that you do not own the copyright to. This includes images, music files, videos, software, games
- In general the personal web space for COMP 1950 should only be used to store assignment and project files which may include:
  - HTML/CSS/JS/PHP files
  - Images, graphics, audio and video files related to the assignments and projects

# Uploading Files to your Personal Web Space

- To upload files to your personal web space you will use an FTP client
- For COMP 1950 we will use FileZilla
- FileZilla is free software available for Mac and Windows
- You can download FileZilla at:
  - <https://filezilla-project.org/>
- Be careful with the install. The installer may ask you for permission to install a useless and annoying browser toolbar. Do not allow this toolbar to be installed it is not required for FileZilla

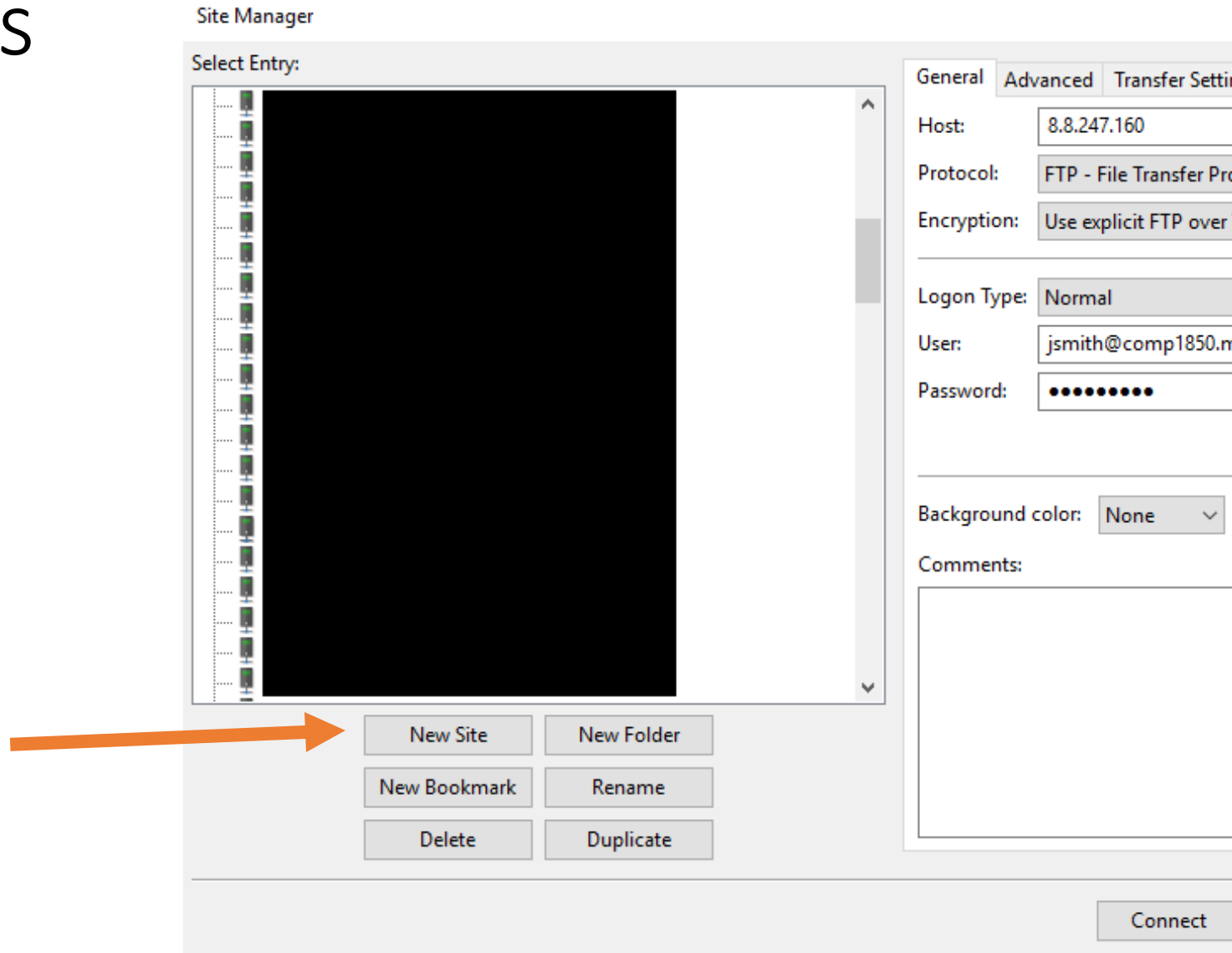
# FileZilla Instructions

1. Open FileZilla and click on "File" in the main toolbar, then click on "Site Manager"



# FileZilla Instructions

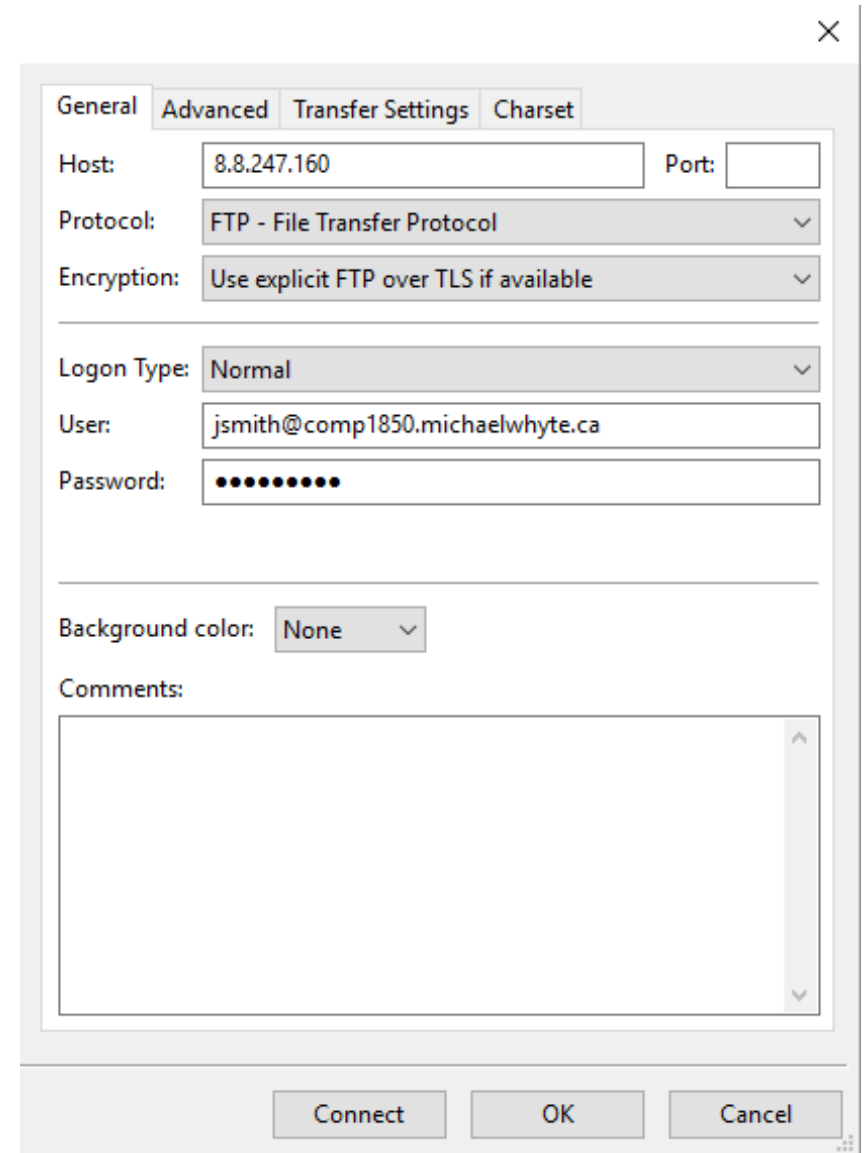
1. Once on the Site Manager screen click on the "New Site" button and give your site a recognizable name
  - For example: COMP1950 – Web Space



# FileZilla Instructions

1. Click over to the right hand side of the window and enter the following information into the fields:

- Host: 8.8.247.160
- User: [first-part-of-your-my-bcit-email]@comp1950.michaelwhyte.ca
- Password: comp1950+[Your BCIT Student Number, with all lowercase letters]
- For Example
  - Student Michael Whyte with a my.bcit.ca email address of mwhyte13@my.bcit.ca with a student number of A00123456
  - Would enter the following:  
Host: 8.8.247.160  
User: mwhyte13@comp1950.michaelwhyte.ca  
Password: comp1950a00123456



The screenshot shows the 'Connect to Site' dialog box in FileZilla, with the 'General' tab selected. The fields are filled with the following information:

- Host: 8.8.247.160
- Port: (empty)
- Protocol: FTP - File Transfer Protocol
- Encryption: Use explicit FTP over TLS if available
- Logon Type: Normal
- User: jsmith@comp1850.michaelwhyte.ca
- Password: (masked with dots)
- Background color: None
- Comments: (empty text area)

At the bottom, there are three buttons: 'Connect', 'OK', and 'Cancel'.



# Practice Exercise

1. Create an HTML file using a Text editor and save it as "hello.html"
2. Add the code "<h1>Hello World</h1>" to the HTML file
3. Upload this file to your personal web space using FileZilla and try and navigate to it in the browser
4. **Note:** the URL will be something like:
  1. <https://comp1950.michaelwhyte.ca/mwhyte13/hello.html>
  2. Change the .../mwhyte13/... part of the above URL to the first part of your my.bcit.ca email address

# File Naming Conventions

- In a regular desktop environment using an operating system such as Windows and MacOS you can get away with using almost any file names
  - For Example:
    - In a desktop environment the file name:
      - Final Project 01 – ver 02.pptx
    - Is a valid file name
- In Web development we utilize server computers and end users of our web sites have to navigate to our site in a web browser
  - For this reason file naming in web development is a little more strict

# File Naming Conventions

- In Web Development file names usually follow these rules
  - Use alpha-numeric characters only
  - Use only lower-case characters only
  - To separate words use the "-" or "\_" character, don't use spaces
- Why only lowercase characters?
  - On server computers the files:
    - My-Web-Site.html
    - my-web-site.html
  - Are considered two different files
  - If you tell your users to navigate to "http://site.com/my-web-site.html" and you uploaded a file as "My-Web-Site.html" the server will return a 404 (file not found) error

# File Naming Conventions

- Use descriptive file names
  - Avoid names such as:
    - `img0002343.jpg`
    - `file01.html`
    - `a23.css`
  - Instead use:
    - `vanier-park.jpg`
    - `about.html`
    - `styles.css`



# File Naming Exercise

- Gather into groups and discuss if the file names on the following slides follow good web design naming conventions

# File Naming Exercise

Web project 01.html

# File Naming Exercise

my\_website\_ver\_02.HTML

# File Naming Exercise

My-favourite-Pic.jpg



# File Naming Exercise

my#logo.ai

# File Naming Exercise

Folder Name: student\_work

# File Naming Exercise

Folder Name: Web projects

# File Naming Exercise

About-me.html.html

# File Naming Exercise

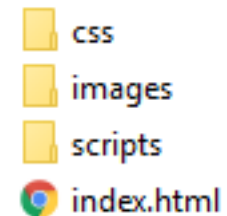
john's-home-page.htm

# File Naming Exercise

Webworks.HTML

# File Organization

- Web sites usually require several files to be linked to HTML documents
- Do not place all your related web files loose in a directory
- A web development best practice is to organize your files by file types into directories and only place html files in the web sites root directory
  - For example
    - All CSS files should be placed inside a "css" or "styles" folder
    - All JavaScript files should be placed inside a "js" or "scripts" folder
    - Images inside an "images" folder
    - Video and audio files inside a "media" folder



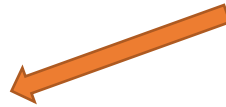
# HTML and HTML5 Review



# A good Basic HTML5 Document

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title></title>
</head>

<body>
</body>
</html>
```



- meta charset="utf-8" tells the computer how to convert your text into a series of bytes and how to convert the bytes back into text again.
- The meta charset stands for "character encoding"
- "utf-8" allows for standard English characters as well as many non-English characters

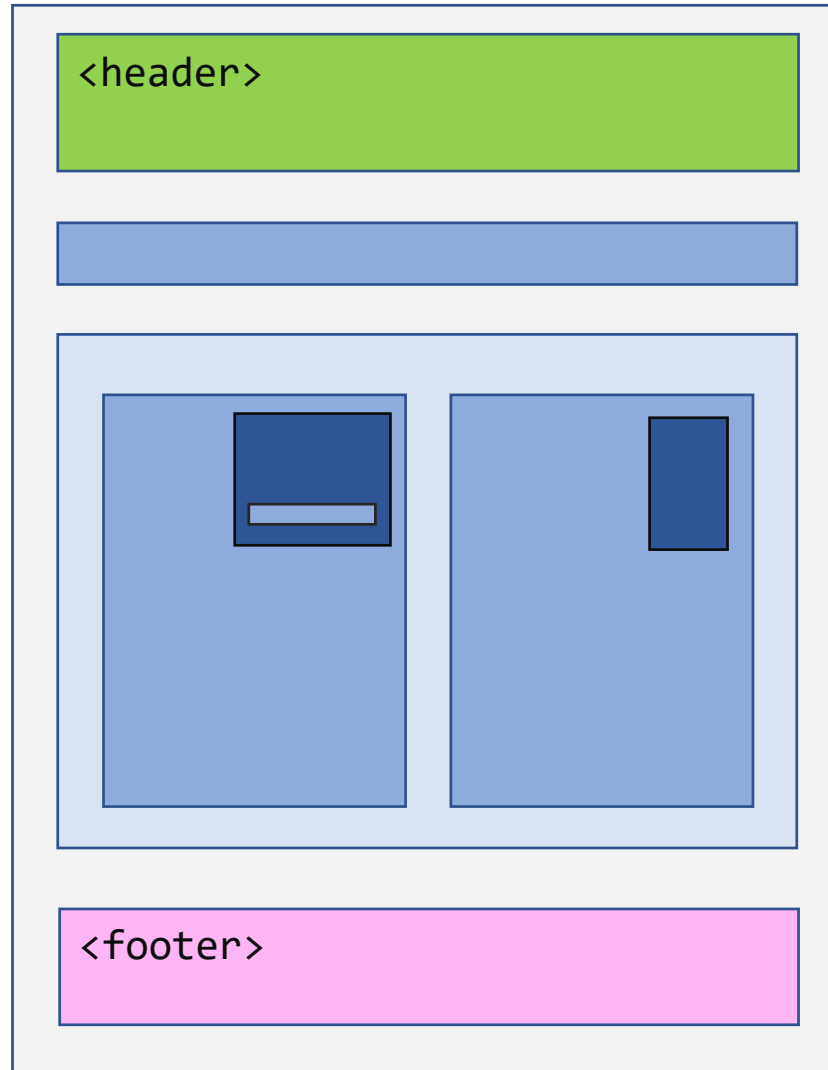
# HTML5 Semantic Markup

- HTML and HTML5 use semantic tags to describe the content
- Key advantages of semantic content:
  - Content is relational - context can be determined easily
  - Extracting important pieces of information is simple (CITE or CODE tags for example)
  - Search engine optimization (SEO)
  - Syndication
  - Big win for frameworks such as jQuery
  - Screen reader friendly
  - Increased Accessibility
  - Separating style (CSS) from structure (HTML), makes it easier to maintain

# Some Common HTML and HTML5 Semantic Tags

- header
- footer
- main
- nav
- section
- article
- aside
- figure
- h1 – h6
- p
- ol, ul, li
- a
- em
- strong

# Header & Footer Tags



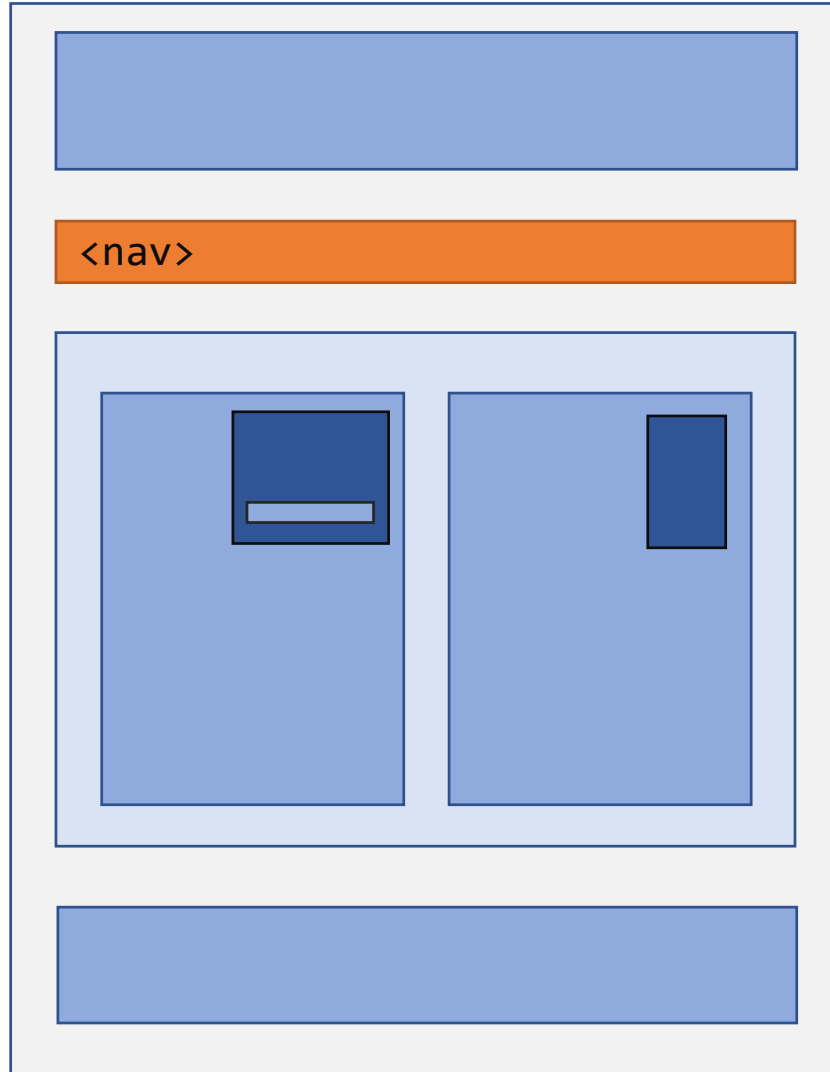
# Header & Footer Tags

- Used to denote the header and footer of a document and or of a section or an article or any section of a document such as an aside
- 90% of the time the header will go at the top of your document and footer will go at the bottom of a document
  - This is not a requirement
- The header tag and the footer tag can be nested inside an article tag or a section tag to denote the head of an article and the footer of an article
- An HTML5 page can have multiple headers and footers
  - An Example:
    - HTML page has a main header and footer
      - Each article in the HTML page can have it's own header and footer (optional)

`<header></header>`

`<footer></footer>`

# Nav Tag

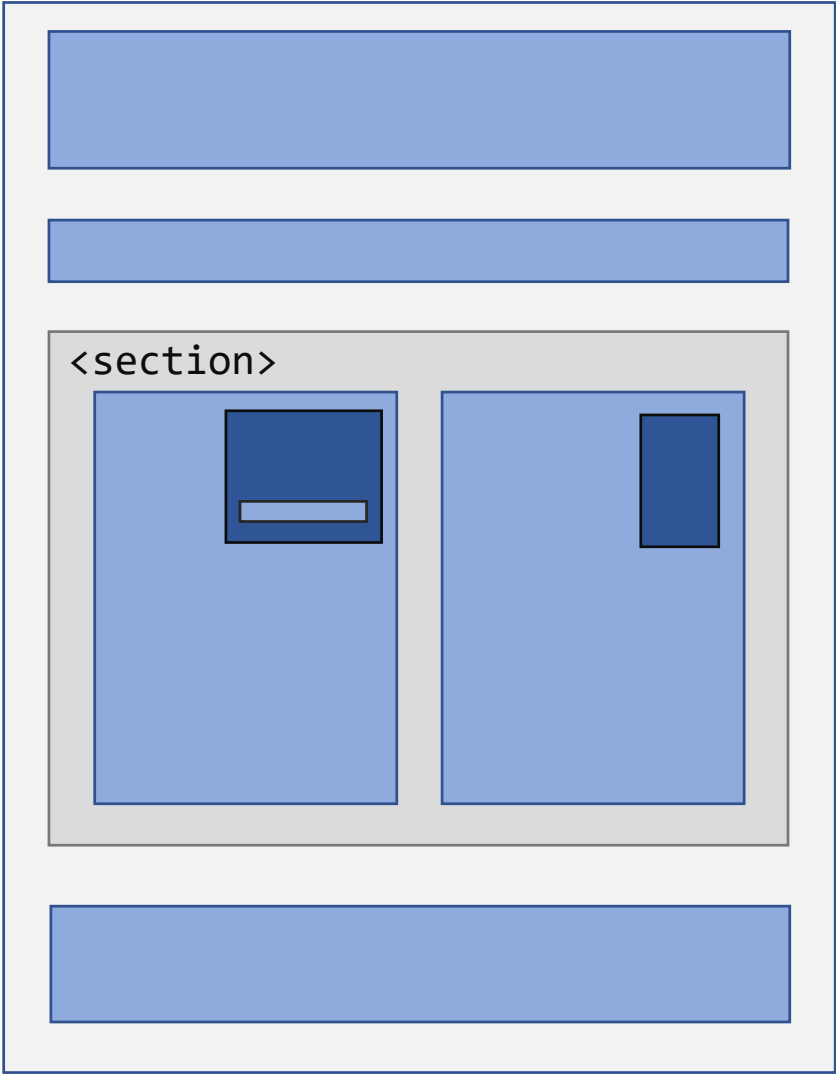


# Nav Tag

- Intended for major navigation items

```
<nav></nav>
```

# Section Tag



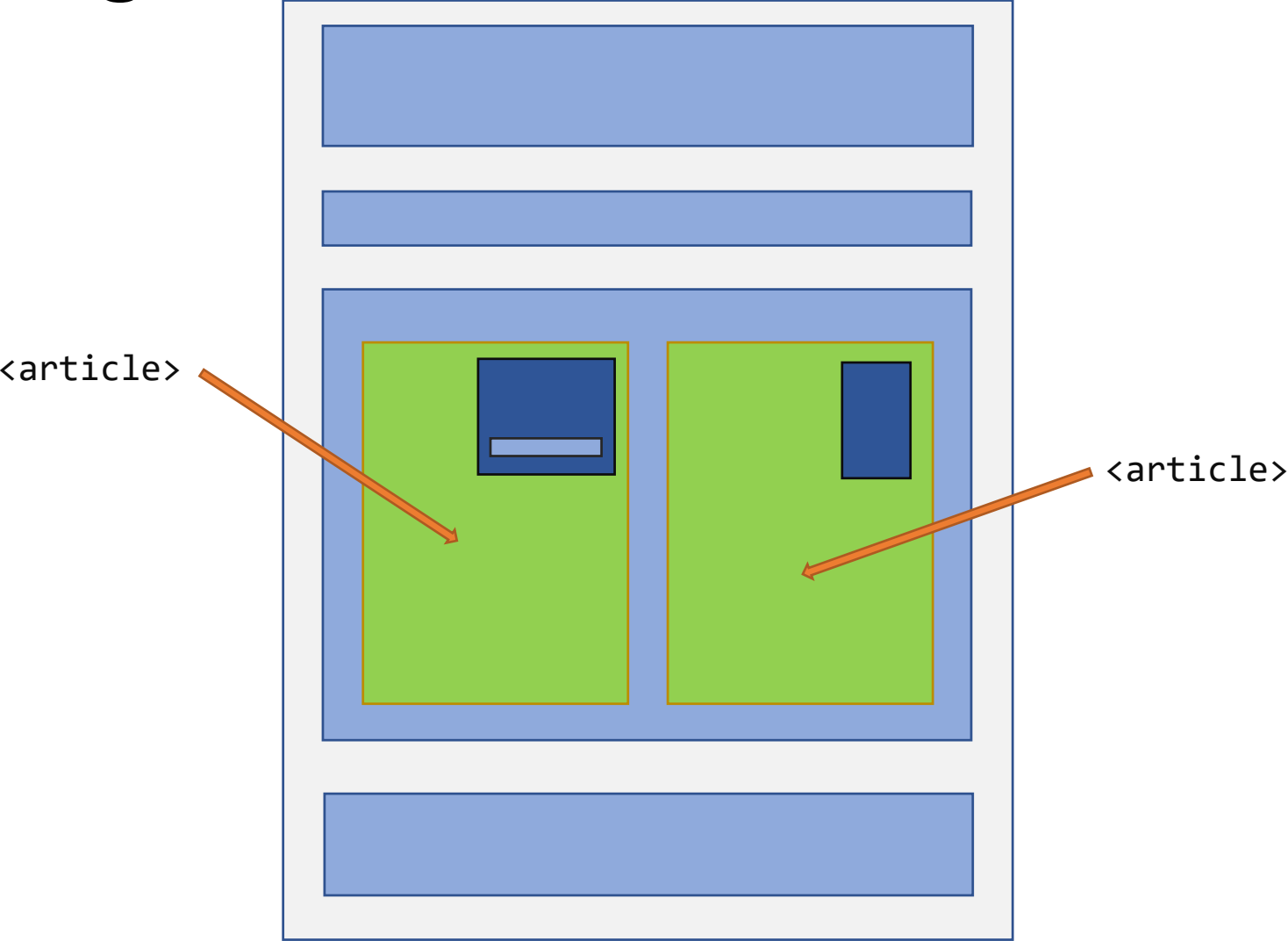


# Section Tag

- A tag used for grouping together content that is related
- Similar to a div tag
  - Differences from a div tag
    - A div tag has no semantic value
    - A section tag should only contain content that is related to each other
- Opinions differ on the use of the section tag versus the article tag

`<section></section>`

# Article Tag

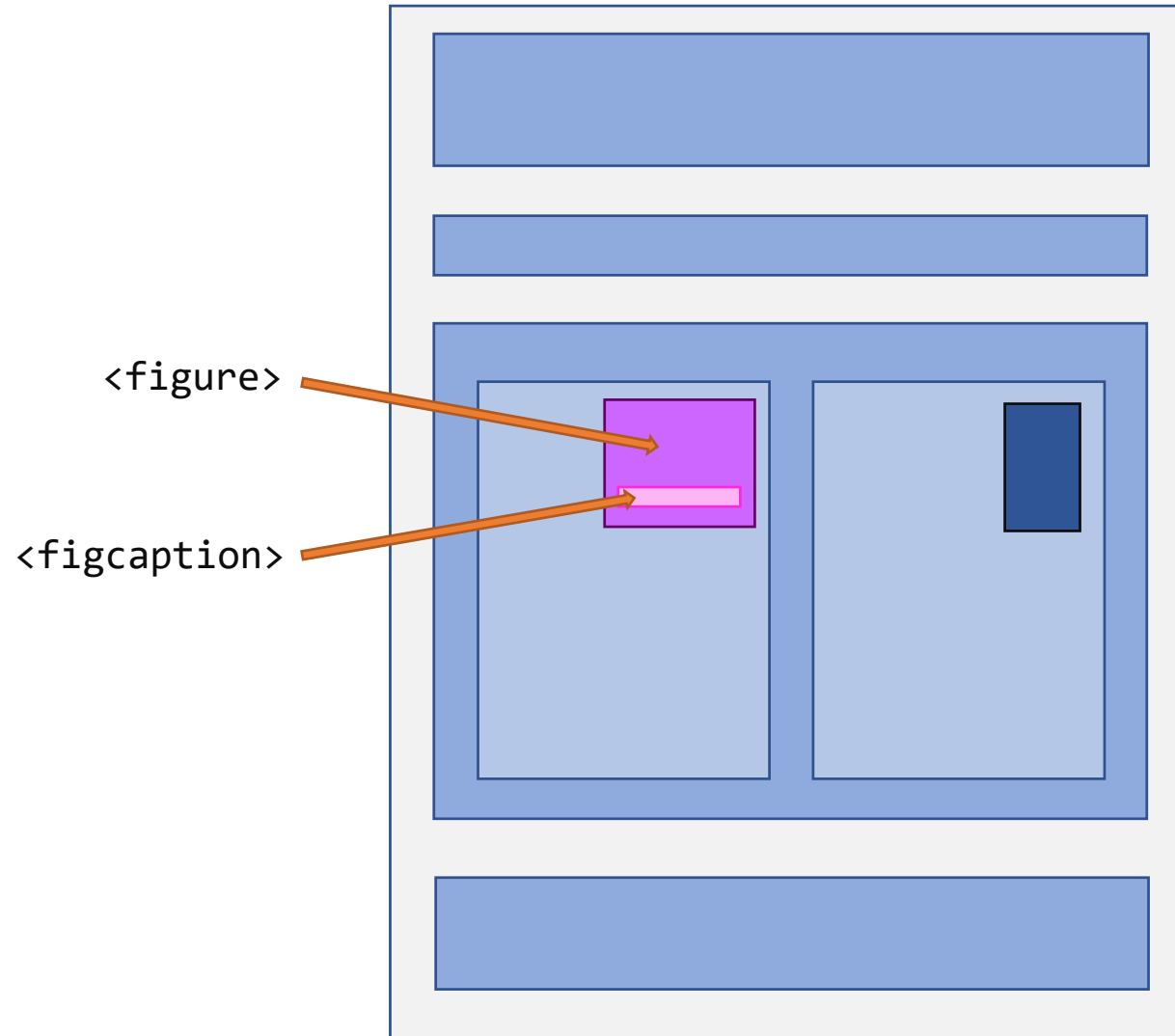


# Article Tag

- A tag used for grouping together content that is related and can be self contained
- Very similar to the section tag
  - Differences from a section
    - This is up for debate – Google – “article vs section tag”
    - Personal thought
      - Use article if the content needs to be portable and taken out of a web page such as for an RSS reader, a forum system, news stories and reviews
- Opinions differ on the use of the section tag versus the article tag

`<article></article>`

# Figure and figcaption



# Figure and figcaption

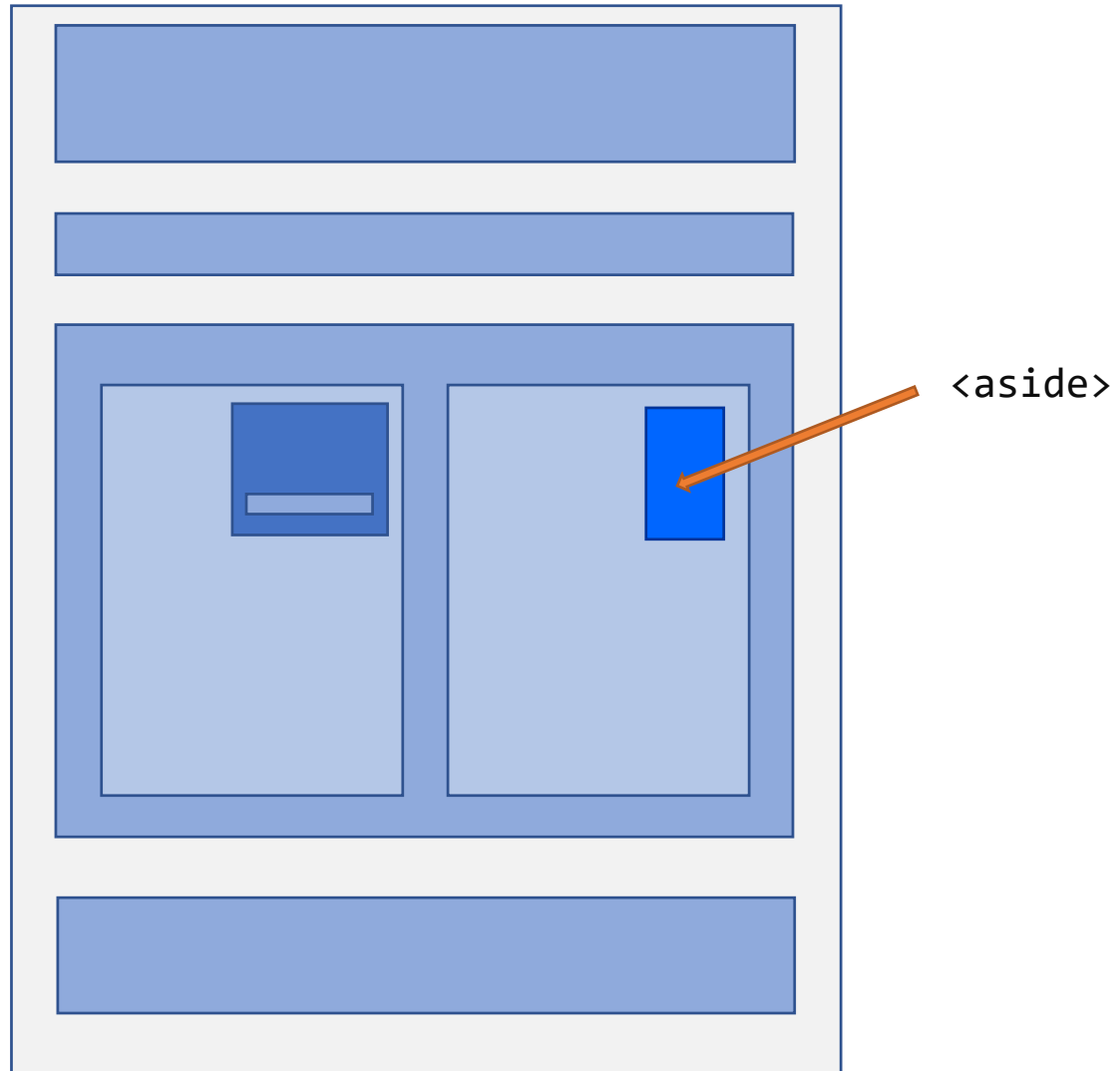
- “figure” elements are used for content that is related to an article or document but that could be removed from the article without affecting the meaning of the article or document
- “figcaption” elements provide captions to figures.
- “figures” do not have to have a “figcaption” element
- Only one “figcaption” element can appear inside a “figure” element
- Content can appear before or after the “figcaption” element inside the “figure” element
- “figure” elements can contain all types of content, including text, images, tables, audio, and video

```
<figure>
```

```
    <figcaption></figcaption>
```

```
</figure>
```

# Aside Tag

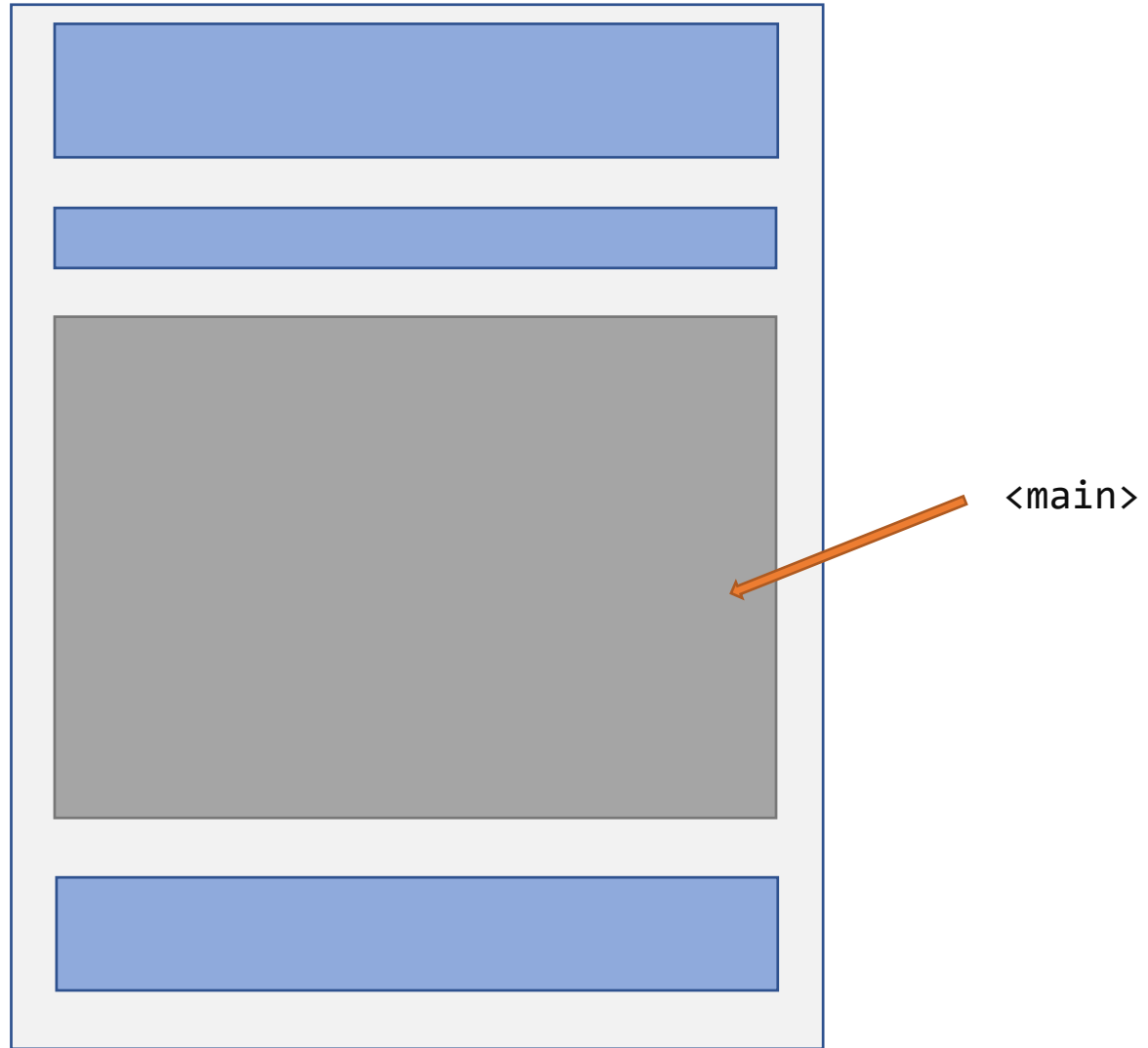


# Aside tag

- The <aside> tag is not a sidebar
- Used for content that is somewhat related to the article that the <aside> element is used with
- Items placed inside the <aside> tag could be removed from the page without having a major effect on the overall meaning of the article
- It is the content not the position, that determines what should go inside an <aside> tag
- An aside can be positioned left, right, top and or bottom

<aside></aside>

# Main Tag





# Main Tag

- New element introduced in February 2013
- The main element is meant to wrap the main content section of a web page
- Only a single main element is allowed per page
- The main element must not be a child element of an article, aside, footer, header or nav element<sup>1</sup>

1. <http://www.w3.org/html/wg/drafts/html/master/grouping-content.html#the-main-element>

# CSS Review

# Cascading Style Sheets

- CSS or Cascading Style Sheets is the presentational layer of front-end web development
- All typography styles and layout styles are handled by CSS
- For most web sites the common way to include a CSS file in an HTML file is via a "link" tag

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

# CSS Selectors

- Element selectors
  - Select elements by their tag name
    - `p {}`, `header {}`, `nav {}`
- Class selectors
  - Select elements by a class name
  - Use a "." before the class name
    - `.some-class {}`
- Id selectors
  - Select an element by its Id name
  - Discouraged in CSS to use Id's as they are difficult to overrule
  - A good rule of thumb is to use classes in CSS and save Ids for JavaScript or internal page links
  - Use a "#" before the id name
    - `#some-id {}`

# CSS Selectors

- Descendant Selectors

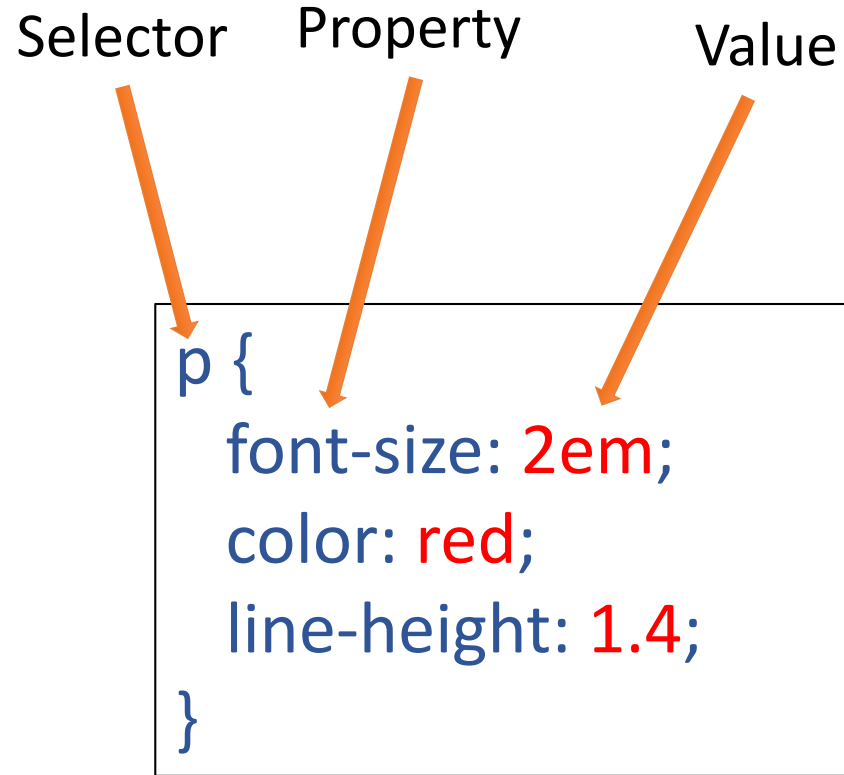
- Used to select elements that are descendants of other elements in an HTML document
- For example:
  - To select only "h1" tags inside a "header" element you would write:
    - `header h1 {}`

- Multiple Selectors on one rule set

- To apply styles to multiple elements use the "," character between selectors
- For example:
  - To select "p" and "li" elements you would write:
    - `p, li {}`

# CSS Syntax

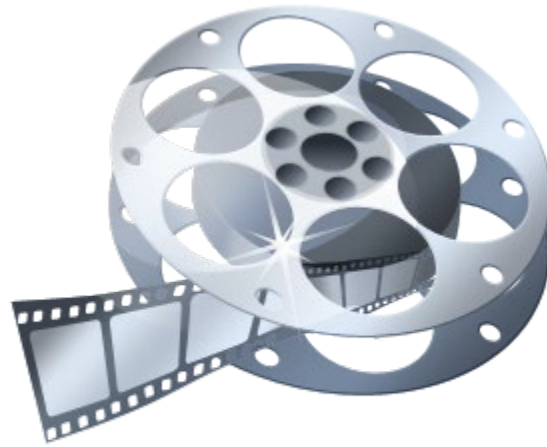
- CSS files are composed of "rules" that describe the styling to be applied to the selected element(s)
- Inside the "{}" of a CSS rule set are properties and value pairs that determine how an element should be "styled" on an HTML page



# In-Class Exercise 01

- Open the "in-class-exercise-01-start" folder in Visual Studio Code
- Create a CSS file and attach it to the "index.html" file
- Write CSS code to make the "index.html" file look as close as possible to the completed file shown on the projector (or open the "in-class-exercise-01-screenshot.png" file as a reference)
- Hints
  - The wrapper has a width of 800px
  - The fonts used are:
    - Arial / Helvetica, sans-serif for the headings and navigation items
    - Georgia, serif for the content text and the footer text
    - Visit CSS Font Stack to get the font stack code for these fonts
      - <https://www.cssfontstack.com/>

# HTML5 Audio and Video





# HTML5 Media Tags

- HTML5 adds native video and audio tags
- Common audio formats used for HTML5 include:
  - MP3 (Safari, Edge, IE, Chrome, Firefox)
  - Ogg Vorbis (Opera, Firefox, Chrome)
- Common video formats used for HTML5 include:
  - MP4 (Safari, Edge, IE, Chrome, Firefox)
  - WebM (Opera, Firefox, Chrome)
- Use the pre-load attribute to tell the browser to pre-load media files or not
- Use the <source> tag to utilize multiple formats for the audio and video tags
- Optional:
  - Provide a hyperlink to the content for worse case scenarios

# Audio Tag

- Used to play audio files in the browser without the use of a plug-in
- Can be controlled with JavaScript for use in games or other interactive pages

`<audio></audio>`

# Audio Tag Attributes

- autoplay attribute

- If you put the autoplay attribute inside the audio tag then audio will play automatically when the web page loads

```
<audio autoplay></audio>
```

# Audio Tag Attributes

- controls attribute
  - If you put the controls attribute inside the audio tag then the browser will display user controls to play and pause the audio

```
<audio controls></audio>
```

# Audio Tag Attributes

- loop attribute

- If you put the loop attribute inside the audio tag then the browser will loop the audio continuously

```
<audio loop></audio>
```

# Audio Tag Attributes

- preload attribute
  - Tells the browser whether or not to preload the audio
  - preload="auto" – the audio will preload
  - preload="metadata" – only the audio metadata will preload
  - preload="none" – the browser will not preload the audio

```
<audio preload="auto"></audio>
```

```
<audio preload="metadata"></audio>
```

```
<audio preload="none"></audio>
```

# Using the Audio Tag with the Source Tag

- Some older versions of browsers require different audio formats
- Use the source tag to provide links to the audio files formatted in different formats
- Browsers will ignore unsupported formats
- Anything inside the audio tag that is not a source element will be exposed to non-supported browsers (mainly older versions of IE) but will be hidden in browsers that support the audio tag
  - This provides a means of creating fallback text that will only be displayed in older non-supported browsers

```
<audio controls preload="none">  
  <source src="my_audio_file.ogg" />  
  <source src="my_audio_file.mp3" />  
</audio>
```

# Writing an HTML5 Audio Tag with Fallbacks

```
<audio controls preload="none">
```

```
  <source src="my_audio_file.ogg">
```

```
  <source src="my_audio_file.mp3">
```

```
  <a href="media/la_pegatina_penjat.mp3">Download the song</a>
```

```
</audio>
```

or

```
<audio controls preload="none">
```

```
  <source src="my_audio_file.ogg">
```

```
  <source src="my_audio_file.mp3">
```

```
</audio>
```

```
<p><a href="media/la_pegatina_penjat.mp3">Download the song</a></p>
```



# Video Tag

- Used to play video files in the browser without the use of a plug-in

`<video></video>`

# Video Tag Attributes

- The video tag attributes are similar to the audio tag attributes
  - The following attributes work the same as they do with the audio tag
    - autoplay
    - loop
    - preload
    - src
  - Since movies take up space you can use some additional attributes to set the width and the height of the video tag
  - You can also set a poster attribute to display an image that displays in the video tag until the video is played
    - poster="some\_image.jpg"
    - IE9 will only display the poster image if the preload attribute is set to "none"

# Writing an HTML5 Video Tag with Link fallback

```
<video controls width="360" height="240" poster="placeholder.jpg">
```

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

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

```
  <a href="movie.mp4">Download the movie</a>
```

```
</video>
```

Or

```
<video controls width="360" height="240" poster="placeholder.jpg">
```

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

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

```
</video>
```

```
<p><a href="movie.mp4">Download the movie</a></p>
```

# CSS Filters

# CSS Filters

- CSS Filters are a powerful tool that authors can use to achieve varying visual effects (sort of like Photoshop filters for the browser).
- The CSS filter property provides access to effects like blur or color shifting on an element's rendering before the element is displayed.
- You can apply multiple filters to a single element

## Filter Types

- Blur
- Brightness
- Contrast
- Drop-shadow
- Grayscale
- Hue-rotate
- Invert
- Opacity
- Saturate
- Sepia
- url – not available in Microsoft Edge

# CSS Filters - Blur

- Applies a Gaussian blur to the input image.
- The value of 'radius' defines the value of the standard deviation to the Gaussian function, or how many pixels on the screen blend into each other, so a larger value will create more blur.
- If no parameter is provided, then a value 0 is used. The parameter is specified as a CSS length, but does not accept percentage values.

```
.blur {  
    -webkit-filter: blur(5px);  
    filter: blur(5px);  
}
```



# CSS Filters - Brightness

- Applies a linear multiplier to input image, making it appear more or less bright.
- A value of 0% will create an image that is completely black. A value of 100% leaves the input unchanged.
- Other values are linear multipliers on the effect.
- Values of an amount over 100% are allowed, providing brighter results. If the "amount" parameter is missing, a value of 100% is used.

```
.brightness {  
    -webkit-filter: brightness(1.5);  
    filter: brightness(1.5);  
}
```



# CSS Filters - Contrast

- Adjusts the contrast of the input.
- A value of 0% will create an image that is completely black.
- A value of 100% leaves the input unchanged.
- Values of amount over 100% are allowed, providing results with less contrast.
- If the “amount” parameter is missing, a value of 100% is used.

```
.contrast {  
    -webkit-filter: contrast(50);  
    filter: contrast(50);  
}
```





# CSS Filters – Drop Shadow

- Applies a drop shadow effect to the input image. A drop shadow is effectively a blurred, offset version of the input image's alpha mask drawn in a particular color, composited below the image. The function accepts a parameter of type (defined in CSS3 Backgrounds), with the exception that the 'inset' keyword is not allowed.
- This function is similar to the more established box-shadow property; the difference is that with filters, some browsers provide hardware acceleration for better performance.
- Another difference to box-shadow is the drop-shadow filter will not follow an images rectangular outline instead the drop-shadow will follow an images pixels that are not transparent. This is useful in applying drop-shadows to PNG files with alpha transparent regions

```
.drop-shadow {  
    -webkit-filter: drop-shadow(3px 3px 8px black);  
    filter: drop-shadow(3px 3px 8px black);  
}
```



# CSS Filters - Grayscale

- Converts the input image to grayscale.
- The value of “amount” defines the proportion of the conversion.
- A value of 100% is completely grayscale.
- A value of 0% leaves the input unchanged.
- Values between 0% and 100% are linear multipliers on the effect.
- If the “amount” parameter is missing, a value of 100% is used.
- Negative values are not allowed.

```
.grayscale {  
    -webkit-filter: grayscale(1);  
    filter: grayscale(1);  
}
```



# CSS Filters – Hue Rotate

- Applies a hue rotation on the input image.
- The value of “angle” defines the number of degrees around the color circle the input samples will be adjusted.
- A value of 0deg leaves the input unchanged.
- If the “angle” parameter is missing, a value of 0deg is used.
- Maximum value is 360deg.

```
.hue-rotate {  
    -webkit-filter: hue-rotate(90deg);  
    filter: hue-rotate(90deg);  
}
```



# CSS Filters – Invert

- Inverts the samples in the input image.
- The value of “amount” defines the proportion of the conversion.
- A value of 100% is completely inverted.
- A value of 0% leaves the input unchanged.
- Values between 0% and 100% are linear multipliers on the effect.
- If the “amount” parameter is missing, a value of 100% is used. Negative values are not allowed.

```
.invert {  
    -webkit-filter: invert(0.9);  
    filter: invert(0.9);  
}
```



# CSS Filters – Opacity

- Applies transparency to the samples in the input image.
- The value of “amount” defines the proportion of the conversion.
- A value of 0% is completely transparent.
- A value of 100% leaves the input unchanged.
- Values between 0% and 100% are linear multipliers on the effect.
- This is equivalent to multiplying the input image samples by amount.
- If the “amount” parameter is missing, a value of 100% is used.
- This function is similar to the more established opacity property; the difference is that with filters, some browsers provide hardware acceleration for better performance.
- Negative values are not allowed.

```
.opacity {  
    -webkit-filter: opacity(0.2);  
    filter: opacity(0.2);  
}
```





# CSS Filters – Saturate

- Saturates the input image.
- The value of “amount” defines the proportion of the conversion.
- A value of 0% is completely unsaturated.
- A value of 100% leaves the input unchanged.
- Other values are linear multipliers on the effect.
- Values of amount over 100% are allowed, providing super-saturated results.
- If the “amount” parameter is missing, a value of 100% is used.
- Negative values are not allowed.

```
.saturate {  
    -webkit-filter: saturate(12);  
    filter: saturate(12);  
}
```



# CSS Filters – Sepia

- Converts the input image to sepia.
- The value of “amount” defines the proportion of the conversion.
- A value of 100% is completely sepia.
- A value of 0 leaves the input unchanged.
- Values between 0% and 100% are linear multipliers on the effect.
- If the “amount” parameter is missing, a value of 100% is used.
- Negative values are not allowed.

```
.sepia {  
    -webkit-filter: sepia(0.7);  
    filter: sepia(0.7);  
}
```



# CSS Filters – Multiple

- You can use multiple filters on an element by space separating them.

```
.grayscale-blur {  
    -webkit-filter: grayscale(1) blur(10px);  
    filter: grayscale(1) blur(10px);  
}
```

