ITMD 441/541
Web Application Foundations

Week 3

FALL 2023 - SEPTEMBER 5, 2023

# Tonight's Agenda

- Git version control and GitHub
- ► HTML and CSS Essentials
- Assignments

# Git version control and GitHub

#### Git

- Git is an open-source distributed version control system.
- Track changes in a set of files
- Originally author was Linus Torvalds in 2005 for use developing the Linux kernel
- Different than client-server VCS systems, every Git directory is a full copy of the repository with history and version tracking without a central server
- Commits are stored with a cryptographic authentication history
- Modified files are staged and then can be committed to the repo
- CLI and GUI implementations
- https://git-scm.com/about

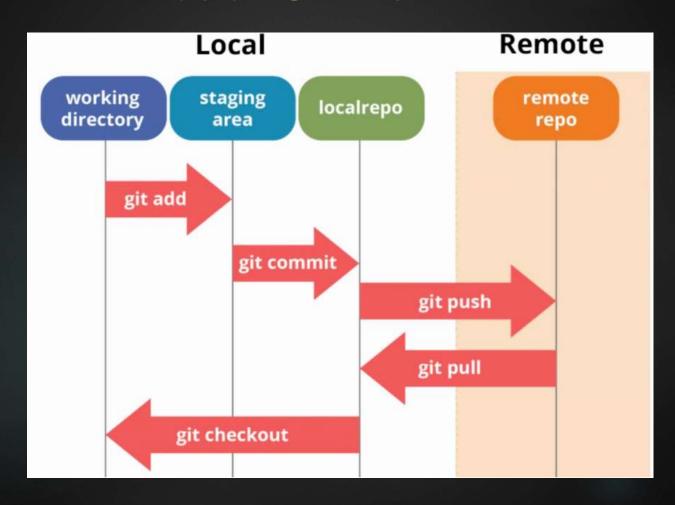
# Git Basics

- Create a local repository from a folder with the init command
  - ▶ git init
- Clone a remote repository from a location to a local copy
  - ▶ git clone [url]
- Browse and inspect the history of the commits
  - ▶ git log
- Stages a file to prepare it for a versioning commit
  - ▶ git add [file]
- Commits staged files to the repository
  - ▶ git commit -m "[description]"

# Git Basics

- Show stage and commit status
  - ▶ git status
- Branches can be created and merged
  - ▶ git branch [name]
  - ▶ git merge [name]
  - git checkout [name] or git switch [name]
- Synchronize changes
  - ▶ git fetch, git merge, git push, git pull
- https://training.github.com/downloads/github-git-cheat-sheet/

# Visual Git Flow



# GitHub

- Web service that is used to host Git repositories and facilitate sharing and social coding
- Founded in Feb 2008 and launched in April 2008
- Largest of the source code hosting platforms
- ► Allows for public or private repositories
- Purchased by Microsoft in June 2018 for \$7.5 billion
- Integrates hosting, version control, issue tracking, discussions
- Code suggestions with pull requests

# GitHub

- GitHub Pages launched in 2008
  - ► Static web hosting service
  - Content stored in a GitHub repository
  - Serve files directly, parse markdown, or integrate with Jekyll
- Gist launched in 2008
  - ► Hosting for small code snippets instead of projects
  - ▶ Way to share snippets with other uses or elsewhere on the web
  - ▶ Hosted as a repository
- https://en.wikipedia.org/wiki/GitHub
- https://github.com/

# Git Resources

- Git website and official book
  - https://git-scm.com/
  - https://git-scm.com/docs/gittutorial
- ▶ Git Cheat Sheets
  - https://training.github.com/
- GitHub getting started with Git
  - https://docs.github.com/en/get-started
  - https://docs.github.com/en
- ▶ Lots of Git tutorials on YouTube

# HTML Essentials STRUCTURE OF A WEB PAGE

# HTML

- ► HTML (HyperText Markup Language)
- Markup language that defines the structure of a web page and its content
- Made up from a series of nested elements
- Elements are made from tags and content
- https://developer.mozilla.org/en-US/docs/Learn/Getting started with the web/HTML basics

#### HTML Element

- Opening Tag
  - Name of the element wrapped in opening and closing angle brackets
- Closing Tag
  - Similar to opening tag but has a forward slash before the element name
- Content
  - Element content, can be text, images, or other elements
- Element
  - Opening and closing tags and content



#### HTML Attributes

# Attribute class="editor-note">My cat is very grumpy

- Attributes add extra data to the element that does not visibly appear on the page
- Should include
  - ▶ Space between it and element name or any other attributes
  - Equal sign after the attribute name
  - Attribute value immediately after the equal sign wrapped in quotes
- Quotes around attribute value are not required if the value doesn't have whitespace or special characters. It is good form to always quote the values anyway for consistency and understandability.

#### HTML Attributes

- ▶ All HTML elements have general and specific attributes
- For example, an anchor element (a) has a href attribute that points to the location of the link. An image element (img) has a src attribute that points to the source of the image.
- All HTML elements have a class and id attribute
- Class
  - Non-unique identifier that can be used for targeting with styling or other reasons
  - Can be reused on a page, and element can have multiple class values
- id
  - ▶ Unique identifier that can be used for targeting with styling or other reasons
  - ▶ Must be unique on the page, and element can only have one id
- Boolean attributes may or may not have a value

#### HTML Elements

- Two main categories of HTML elements
- Block elements
  - Appear as new line on the page
  - ▶ Visible "Block" that spans the width of the page
  - ▶ Typically, structural elements: heading, paragraphs, lists, div, etc.
  - ▶ Block elements generally should not be nested in an inline element
- Inline elements
  - Contained in a block element and wrap smaller amounts of content
  - ▶ Typically used with text elements. Things that flow with the content.
- ▶ Behavior can be changed with CSS but still in same category

# HTML Elements

- Some HTML elements are "empty" and do not contain content
- ▶ Image is a good example
  - <img src="url-to-image">
  - ▶ No content and no closing tag
- ► HTML can have comments
  - <!- comment here -->

# HTML5 Page Structure

\*You may still see the following meta tag, but it should be unnecessary at this point, it was a tag for IE version < 11. Some template generators still add it.

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

# Basic Tag

- Here are some basic HTML elements you should already be familiar with
- Headlines: h1, h2, h3, h4, h5, h6
- Paragraphs: p
- Anchor: a
- ► Lists: ul, li, ol
- Images: img
- ▶ Bold and Italic: strong, em
- General block: div
- General inline: span
- ▶ Basic Forms: form, input, textarea
- ▶ We will talk about more later, including HTML5 specific ones

# HTML5 Semantic Structural Elements

- HTML5 provides semantic tags to give areas of your page structural meaning
- main dominant content of the <body> of a document
- article self contained content that could be reusuable
- <u>section</u> generic standalone section in document
- <u>aside</u> content not directly related to main content
- header introductory content that groups content or nav
- <u>nav</u> section of the page meant for navigation
- <u>footer</u> a footer grouping for its nearest ancestor
- https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\_to\_HTML/Document\_and\_website\_str\_ ucture

# CSS Essentials PRESENTATION ON A WEB PAGE

#### CSS

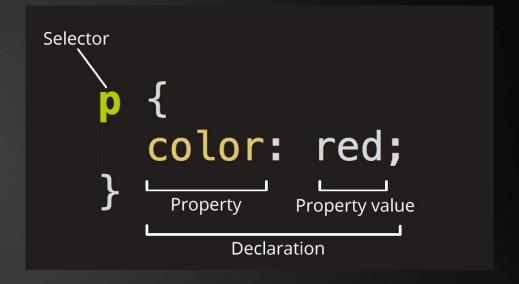
- CSS (Cascading Style Sheets)
- Style sheet language, not markup or code, that you use to decorate or style HTML elements
- Consists of rules that define what should be selected and what styling properties should apply
- https://developer.mozilla.org/en-US/docs/Learn/Getting started with the web/CSS basics

#### CSS

- These are the three primary ways to use CSS on a page in order of preference.
- ▶ Inline
  - CSS is added directly to an element using the style attribute on the tag
- Embedded
  - ► CSS is added to the head section of a page using the <style></style> tags
- External
  - ▶ CSS is added to the page with an external file reference using a link tag
  - ▶ This should be in the head section
  - link href="styles.css" rel="stylesheet">

# CSS Ruleset

- Selector
  - Defines what elements this rule applies to
  - ▶ Many different types of <u>selectors</u>
- Property
- Property value
- Declaration
  - ▶ A property and value in a rule



# **CSS** Position

- https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\_layout/Positioning
- Positioning allows you to change the way elements interact and display compared to the normal document flow
- Static default normal flow
- Relative puts the element in normal flow but allows you to move it after
- Absolute takes element out of the flow and positions it to the first parent that has a positioning context
- ► Fixed positions element based on browser viewport
- Sticky newer combination of relative and fixed

# CSS Box Model

- https://developer.mozilla.org/en-US/docs/Learn/CSS/Building\_blocks/The\_box\_model
- box-sizing
- Standard Model
  - Padding and border is added to set any width or height
- ▶ Border Box Model
  - Padding and border is included in any set width and height
- Some properties only work on block or inline boxes



# CSS Cascade

- https://developer.mozilla.org/en-US/docs/Learn/CSS/Building blocks/Cascade and inheritance
- ▶ This determines how the rules flow and override each other
- Cascade
  - ▶ The origin and order of the CSS rules
- Specificity
  - Algorithm that browsers use to decided what rules to use on an element if multiple selectors apply
- Inheritance
  - Some CSS properties on a parent element are inherited to a child and some are not. (mostly text things)

# CSS Specificity

- Based on a point scoring system
- ▶ id 100 points
- ▶ class 10 points
- ▶ element 1 point
- Inline styles can be considered 1000

Selector	Identifiers	Classes	Elements	Total specificity
h1	0	0	1	0-0-1
h1 + p::first-letter	0	0	3	0-0-3
<pre>li &gt; a[href*="en-US"] &gt; .inline- warning</pre>	0	2	2	0-2-2
#identifier	1	0	0	1-0-0
button:not(#mainBtn, .cta)	1	0	1	1-0-1

# CSS3

- Newest standard being developed
- Unlike CSS2, CSS3 was split into smaller modules that could be developed and published independently
- https://www.w3.org/Style/CSS/Overview.en.html
- https://www.w3.org/Style/CSS/specs.en.html
- Includes new properties and features such as
- CSS Selectors, border radius, box shadow, box sizing, opacity, gradients, multiple backgrounds, transitions, transforms, animations, flexbox, grid, media queries, variables, and more.
- https://en.wikipedia.org/wiki/CSS
- https://www.css3.info/modules/

# CSS Variables

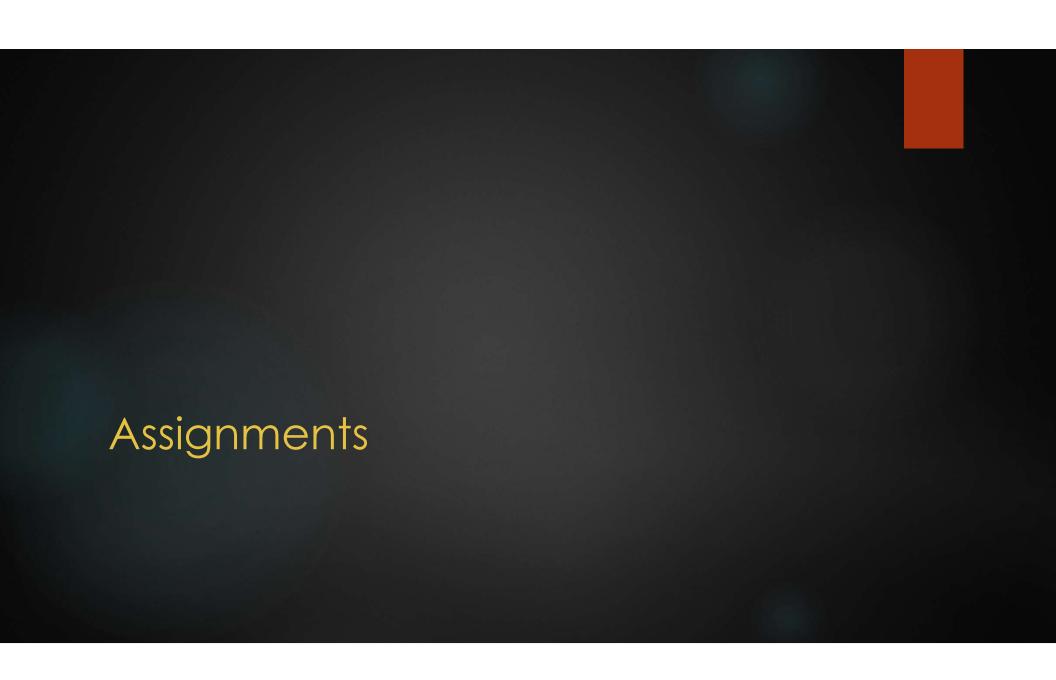
- CSS custom properties or variables have been added as part of CSS3
- Allows you to define a custom property with a value and reuse it
- Typically defined on the :root pseudo-class and they inherit all the way down but doesn't have to.
- Begin with double hyphen (--)
- Use the custom property with the var() function
- https://developer.mozilla.org/en-US/docs/Web/CSS/Using CSS custom properties

```
:root {
    --main-color: red;
}
Element {
    color: var(--main-color);
}
```

#### **CSS** Reference

- https://css-tricks.com/almanac/properties/
- https://developer.mozilla.org/en-US/docs/Learn/CSS
- https://www.tutorialrepublic.com/css-tutorial/
- https://tutorial.techaltum.com/css3.html
- https://css-tricks.com/snippets/css/a-guide-to-flexbox/
- https://css-tricks.com/snippets/css/complete-guide-grid/

A lot of videos out there that will help you review the steps for incorporating CSS and HTML



# Reading/Assignments

- Quiz 2 Week 2 Content Due end of day September 7 before midnight
- ▶ Lab#1: GitHub Repo and index page for our course
  - Create a GitHub account if you don't have one
  - Create a repo for this course
  - Create an index.html page in the root of the repo and it will be an intro page about you and a table of contents for future class examples. Add CSS and do the best you can with layout.
  - Create a GitHub Pages from the repo so the site is hosted
  - ▶ Due Tuesday, September 19, see blackboard
- Read MDN Guides on HTML and CSS (especially if you have limited experience)
  - https://developer.mozilla.org/en-US/docs/Learn/HTML
  - https://developer.mozilla.org/en-US/docs/Learn/CSS
- Quiz 3 Week 3 Content Assigned by end of day Friday Due September 14.

# https://missanita2.github.io/

This is what your URL that you are uploading to Blackboard should look like. The "missanita2" should be the username **you** used when creating your GitHub account.