Intro to HTML/CSS

with Git

define *web development*

the work involved in developing (creating) a website for the internet

examples: web design, web content development, client-side/server-side scripting, network security configuration, database implementation, etc (tons of code!)

website basics

what you need to start:

- a computer
- an internet browser
- some HTML for content, a little CSS for styling

what is HTML?

- HTML : HyperText Markup Language
  programming language used for creating and visually representing a webpage

- HTML5 : the latest evolution of HTML
  contains new elements, attributes and behaviors; larger set of technologies that allows more diverse and powerful web sites and applications

enhancing HTML with CSS

- CSS : Cascading Style Sheets
  stylesheet language used to describe the presentation of a document; used to describe how a structured element must be rendered on screen, paper, speech, other media

- CSS3 : the latest evolution of CSS
  brings a lot of cool properties to the table, like better gradients, transitions, rounded colors, selectors, etc

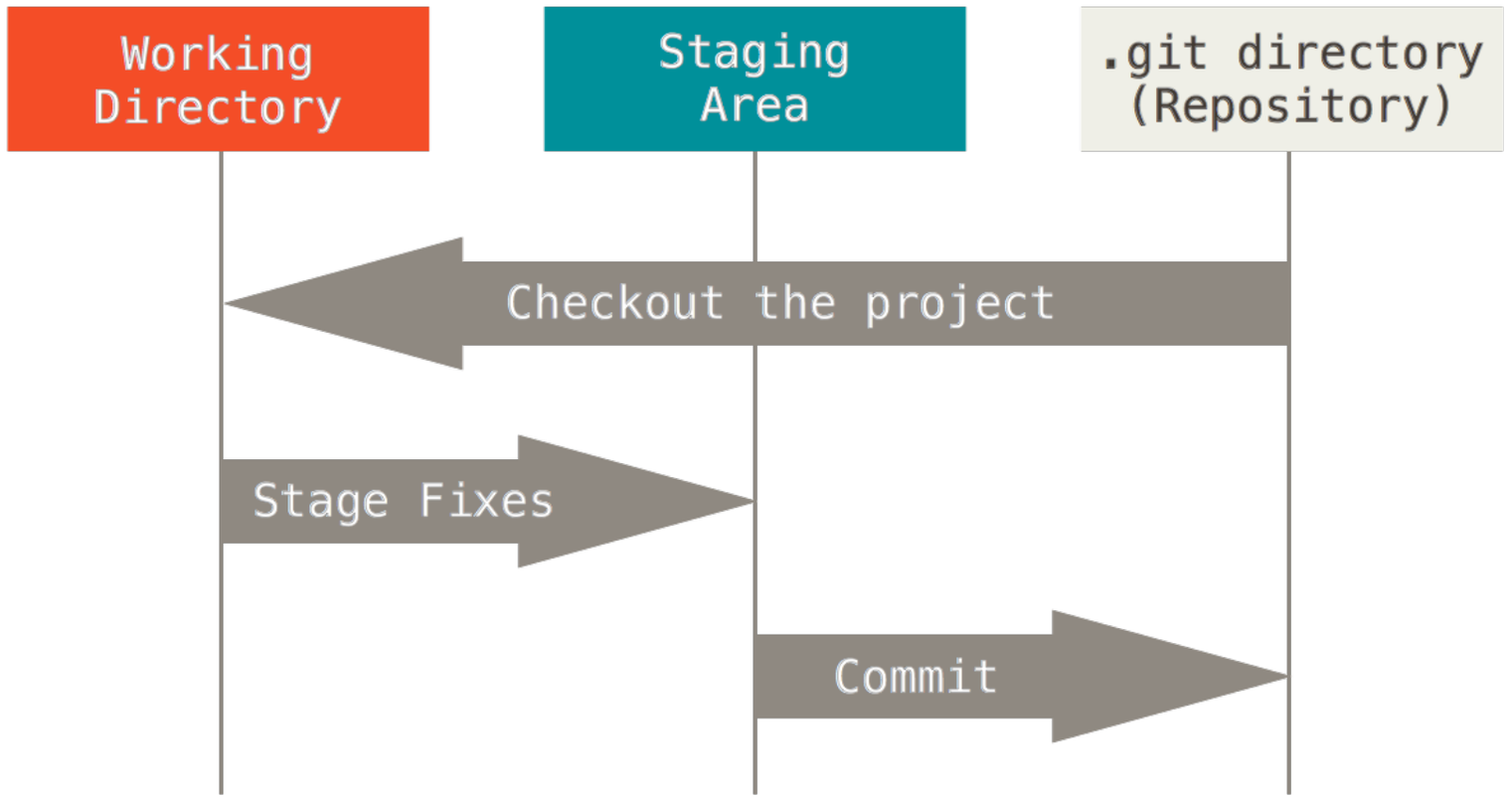
version control with Git

define version control

a system that records your changes to files over time

define Git

the version control system (vcs) created by GitHub. it creates snapshots of your files when changes are committed



> The basic Git workflow goes something like this:
>
> 1. You modify files in your working directory.
> 2. You stage the files, adding snapshots of them to your staging area.
> 3. You do a commit, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.
>
> —Git Documentation

create your own responsive 1-page website while using Git

THE PROJECT

setup

1. download your favorite text editor.
   recommendations: atom or sublime text

2. signup for the student developer pack by GitHub or register for a free account

3. download the GitHub Desktop software

first repo and commit

INSTRUCTIONS

1. create a Sites main folder and another folder for your project (use dashes for spaces)

2. open your text editor and create a styles and images folder and 1 index.html file

3. open the GitHub software, add your project folder as a repo (repository) and create your first commit!

4. publish!

first repo and commit

INSTRUCTIONS

5. add a branch called gh-pages

6. open your text editor and create a styles and images folder and 1 index.html file

7. open your GitHub, add your project folder as a repo and create your first commit!

8. sync!

index.html

INSTRUCTIONS

1. add the necessary HTML elements to start a document. you may also know these as tags

2. comment out any links to stylesheets or javascript

3. add some text to the body!

<!doctype html> → INDICATES THE DOCUMENT TYPE (HTML)

<html> → ROOT ELEMENT: ALL OTHER ELEMENTS MUST BE DESCENDANTS

<head> → HEAD ELEMENT: PROVIDES GENERAL INFO, LIKE META DATA & THE TITLE

<meta charset="utf-8"> → CHARACTER ENCODING DECLARATION

<title>Our First GitHub Project</title> → TITLE ELEMENT: TITLE OF THE PAGE

</head> → END HEAD ELEMENT

<body> → BODY ELEMENT: CONTENT. ONLY ONE BODY.

Hello, World! → PARAGRAPH ELEMENT

</body> → END BODY ELEMENT

</html> → END ROOT ELEMENT

index.html

INSTRUCTIONS

4. let's add some more HTML elements to create an image, short bio, and several links to your social media profiles. be sure to use semantic elements like <header>, <section>, etc. we will be using a horizontal list for our links.

   note: the mozilla developer network is great to learn about anything web dev!

5. ensure you commit for each change!

6. open your file or GitHub pages link in your browser and open the developer tools

styles.css

INSTRUCTIONS

1. in your styles folder, create a new file in your text editor called styles.css. commit and sync!

2. let's start adding some basic styles and fonts to our HTML elements through CSS

how do you start writing css?

by targeting HTML elements with properties
defined by “rulesets” and “declaration blocks”

two things must be done first

1. Create a CSS file and add some rulesets

2. Link that CSS file in your HTML document

SELECTOR: TARGETS AN ELEMENT OF THE HTML DOCUMENT

```
body {
background: red;
color: white;
font-size: 80px;
}
```

OPEN BRACKET: STARTS THE CSS DECLARATION

RULESET: THE ENTIRE SELECTOR-DECLARATIONS BLOCK

DECLARATION BLOCK: PROPERTIES AND THEIR VALUES

CLOSING BRACKET: ENDS THE CSS DECLARATION

```
<!doctype html>

<html>

  <head>

    <meta charset="utf-8">

    <title>Our First GitHub Project</title>

    <link rel="stylesheet" src="styles/styles.css">

  </head>

  <body>

    <p>Hello, World!</p>

  </body>

  </html>
```

ADDED A “LINK” ELEMENT TO THE HTML SO THAT THE HTML PAGE KNOWS WHERE TO GRAB THE PROPERTIES FROM

CSS

MORE INFO

- CSS follows the rule of cascading and inheritance
  inheritance is the concept that a child element will inherit the properties of the parent element unless it is explicitly overwritten

- for example, we declared that the body should have a red font color, so whatever was inside the paragraph <p> element inherited the red font color

CSS

MORE INFO

- you can create your own class and id selectors and assign them to elements in your HTML
- for example, if I had multiple paragraphs within my HTML document and wanted to change the font color on some of the paragraphs, I could create a class to target those paragraphs. if I wanted to target and manipulate just one element, then I would use an id

Note: it is recommended NOT to use ids to avoid using the same id on multiple elements. also be warned that if you write css for an id for an element and then apply a class, the properties under the id will overwrite anything similar in the class. Ids take precedence.

basic properties of CSS

Property	Definition	Example Values
width, height	specifies the width or height of an element. may use px (pixels) or percentages (%). %s are relative to the parent	width: 100px; height: 100%;
padding	adds space to the inside of an element. may use px or %. note: it adds to the width or height of an element	padding: 10px; padding: 1px 2px 3px 4px; padding-top: 5%; padding-bottom: 15px;
margin	adds spacing to the outside of an element. may use px or % or auto.	margin: 20%; margin: 5px 10px 10px 5px; margin-left: auto; margin-right: auto;
font-size, font-style, font-family	defines the properties of fonts. for font-size, you may use px, Ems, or rems	font-size: 10px; font-family: “Helvetica”; font-style: normal;
color	defines the color of text. may use HEX, RGB, RGBA, HSLA, or actual color names for values	color: red; color: #fff; color: rgb (100%, 0%, 0%);

do’s and don’ts of CSS

Do	Don’t
make sense when you name your classes	start naming any classes with numbers: 1,2,3, etc. it won’t compile
insert a new line for each declaration	compress your declarations in a declaration block. it makes it harder to read at first
check if you’re naming a property right	try to minify properties if you don’t understand it. it’s better to code the long way first
use user-friendly colors	use too many colors or distracting ones
ensure text is readable across all devices	use tiny or big text everywhere. adjust as needed

styles.css

INSTRUCTIONS continued

3. follow me as we now start creating classes and adding properties to those classes to bring your HTML document to life! we will first add some styles to normalize our document and then we will make your profile photo fit in a circle, add some fonts, etc

4. always test and commit for each change!

styles.css

INSTRUCTIONS continued

5. let's test for responsiveness. common breakpoints are: 320px, 480px, 768px, and 1024px. we will stop at 1024px.

6. add media queries if you must!

7. commit commit commit

the final step:
cross-browser testing

cross-browser testing is important for usability and ensures your website works on every device.

always have your developer tools open when testing! a good testing site is browserstack

checklist

- ✓ your HTML is flawless. all tags are closed
- ✓ each CSS declaration has proper opening and closing brackets
- ✓ your site is responsive
- ✓ you've cross-browser tested on at least 3-4 devices
- ✓ the console has no errors
- ✓ your copy has no grammatical errors

congrats!

you're done!

you're on your way to becoming a web guru!!

tutorial resource

http://go.fiu.edu/lhd-webdev-intro

questions?

twitter: anilanorbel facebook: /its.alina linkedin: alina lebron

you can also follow WICSatFIU on Facebook and
get in touch with me there!