**Command prompts**

dir = directory

cd “new directory name” = change directory to “new directory name”

cd .. = move out of current directory to greater umbrella folder

git init = initialize git

git status = check status of git

In git, everything is saved onto a branch = current version of your project. Every time you save, you are saving a point on your timeline. Git also allows you to branch off at different points in your timeline so that you can compare one point on a parallel branch to another point on another branch. If you like one version, you can merge versions.

Git add index.html = add “index.html” to the git saving box

Git add . = add everything

Git commit –m “our first commit”= save file with a message “our first commit”

Git log = shows your messages

**February 7, 2014**

Client-server model: client = you/your browser. Server = machine sending you the data and filing your request. A server is just a computer. Any client could also be a server.

Host~server

Local = hosted on your machine

Remote = hosted on a different machine

**Request-Response Cycle**

4 types of requests: get, post, put, delete.

Get = request information

Post = submit information

Put = update

**CSS**

Selector: list of tags/ids that will conform to rule. Rule fits inside {}. Rule = attribute name: value

e.g.

{

font-size: 18px;

}

p {

font-size:

Any displayed element can be defined as an id or a class. A class is a group of objects that should be styled together. Id is a way of referring uniquely to an object.

Almost all HTML tags can have an id and a class.

< div id = “idname”> - both in css and and js you can refer to this blog as its id name, “idname”

<div id = “idname” class “invisible” > class is a bigger category. You can tag multiple things as the same class and then refer to all of them later.

Ways to refer to selectors: div, #id, .class

e.g.

div{ …}

#id{…}

.class {…}

div.class

use <strong> and <em> over <b> and <i>

div div h1 {} = any h1 nested in two divs

div div.foo h1 {} = any h1 nested in div class foo, all of which is nested in div

div div.foo h1, h2{} = any h1 nested in div class foo, all of which is nested in div **OR** any h2. “,” = everything before OR everything after.

Add css:

<head>

<link rel = “stylesheet” href =”./style.css”>

</head>

Or

<head>

<link rel = “stylesheet” href =”style.css”>

</head>

**COMMON STYLES**

(font)color, background-color

Rgb or hex values

Color:rgb(255,0,255);

Color:#FF00FF = rrggbb (hexadecimal)

Text-decoration: “none” or “underline”

Width, height, font-size: specify as a percentage (of the div element it’s contained in, not overall page) or pixel value

Smallest screen resolution 800px 640px (w, h)

Most 960px wide

To figure out webpage pixel size, use css3 media query.

Border: size style color

**DISPLAY**

display:block;

takes on full available width of whatever element it’s contained in

display:inline;

display:none;

element is not rendered in the browser until the value is changed ~ invisible

display:inline-block

only takes up needed width to not disrupt document flow

allows block-display properties like w,h, top/bottom padding/margin

you can use this for columns

<span>

Display:inline-block

If you don’t define width/height, it will adjust automatically to your text

**POSITION**

position:static

render element relative to neighbor with no offset

default setting, never need to declare in CSS

position:fixed

regardless of scrolling, object will stay set

position:absolute

sets element to be rendered at specific location in first parent element whose position is not static

position:relative;

bootstrap = good css stylesheet