**Week 2 Notes**

**HyperText Markup Language and Cascading Style Sheets**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>title</title>

</head>

<body>

<!-- page content -->

</body>

</html>

* The above code represents just enough code and meta-data to be considered a valid page
* HTML does NOT affect how a page looks, it is a language to organize data
* CSS is to affect how a webpage looks

**HTML Tags**

Forms

form: everything for a form exists in this space

input: most content will include an input tag, where users can enter in or select options to submit

textarea: function same way as input, but called different

Form attributes –

action: specifies where the form should be sent

method: specifies how the data will be sent (two primary ways, **get** and **post**)

Input Tag

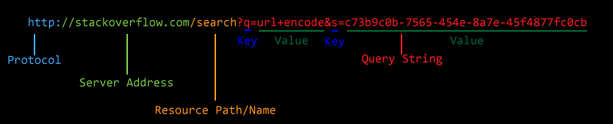
* Has an attribute *type* which changes how it’s displayed
* *name* attribute is how the server will know which part of the form data is associated with e.g. name=”email”

GET and POST

* Sends a key-value pair (e.g. username=alice has the key username and the value alice)

*GET*

* Sends key-value pairs as parts of the URL



http: is the protocol (you might see https: which is the secure version)

//stackoverflow.com is the address of the server we are trying to contact

/search is the path to the resource we are requesting (in this case, only one directory deep)

? indicates that it is part of a query string

key1=value1&key2=value2&keyn=valuen this is the specific format for key-value pairs

* In this case, keys are assigned vals using =, key-val pairs are separated using & symbols

*POST*

* Sends the same kinds of data as a GET
* Instead of sending it as part of the URL, it is sent as part of the request body (you cannot tell what data was sent to a server via a POST request by looking at the URL)

**CSS**

* CSS is a layout tool for specifying how a website looks
* Majority of CSS is about specifying properties and giving those properties values
* Syntax for specifying properties is property: value

Selectors

* Selectors let us set rules for which elements styles will apply to

Syntax



type selector:

span {

}

Class selector

.osu {

}

Id selector

#menu1 {

}

Child selector

Section>span {

}

Descendant selector – applies style to descendants of left item (could do div ul, all ul’s that are descendants of divs)

ul ul {

}

Adjacent sibling (if the two items are directly next to each other, here an h1 followed by an h2)

h1+h2 {

}

General sibling (all siblings after the left item, so all p after an h3 in this case)

h3~p {

}

Priority: ID > class > span

Common CSS Properties

Colors: there are named colors, but you can also use HEX values

Positioning: Uses relative position, like to the screen or to an ancestor element

* Static: default positioning
* Relative: Objects with relative position are placed where they normally would be, and change is measured from that spot
* Absolute: Sets position based on the parent element
* Fixed: fixes the position of the element with respect to the document window (e.g. make the nav bar always be at the top of the page. (suggested not to use: difficult for users of smaller screens)

Measurements: px and em are used to measure pixels. px is an absolute measurement and em is a relative measurement based on page size. You can use % for a section of the page (out of 100)

Space: When laying out documents, everything is a box, and inside of every box is an element. We can specify how far one box is from another and how much space there is between the edge of a box and its content (margin and padding, respectively)