Week 5

1. Introduction

Javascript gives ability to write client side code. This means code that runs on the client instead of the server. It allows manipulation of the document object model (“DOM”).

Javascript is suited to the paradigm of event driven programming. I.E. Something should happen if some event occurs.

1. Useful functions

Document.querySelector -> This allows you to select an element of the DOM

Example

Document.querySelector(‘h1’).innerHTML = …

Function takes 1 argument which is a string of the html element, innerHTML is a property of the html inside.

Document.querySelectorAll(“button”) would return a NodeList similar to an Array. It can be sliced similar to Python.

**Selecting other aspects besides tags in HTMl**

Document.querySelector(‘tag’) -> This means refer to the tag

Document.querySelector(‘#id’) -> This means refer to the ID

Document.querySelector(‘.class’) -> This means refer to the class

**Document**

The document keyword in JS refers to the entire HTML document currently being referenced.

**.addEventListener**

Basically an attribute of document. Takes 2 argument First, a particular event such as ‘click’ or ‘DOMContentLoaded’. Second, a function that is meant to be applied whenever that event occurs.

**Hoisting**

Note that JS on client unlike JS in server, hoisting is not present.

1. Tips

When writing JS code on the client side. Whenever possible you should split the JS code from the HTMl. This means try to avoid stuff like onclick within button elements. However, to still run JS code, you canuse the document.querySelector in your JS code to assign a function to a button and whenever that button is clicked, JS would read it as execute that function.

Note () in JS works similar to Python. If () present it means run function. If not present it means assign function to that element.

Note 2: Note we can only assign a function to a variable like button because JS is a functional language where functions are first class values.

1. Debugging Tips

Whenever you want to debug JS, you can go to the JS console similar to terminal. Go inspect/Console in Chrome.

1. Changing CSS

You can also store data attributes in HTML. Data attributes are just a way to store information within certain HTML attributes. Data attributes always follow this format data-<attributeName>

To access data attributes you must use <element>.dataset.<specialDataSet>

                        document.querySelector('#hello').style.color = button.dataset.color;

1. Types of Event Handlers

* Onclick
* Onmouseover
* Onkeydown
* Onkeyup
* Onload
* Onblur
* …

1. Local Storage

JS also gives you the ability to store information on the client browser. Key commands are

* localStorage.getItem(key)
* localStorage.setItem(key, value)

Never store sensitive data in LocalStorage. This is because every JS file loaded on domain has access to LcoalStorage. When we say domain we mean website. LocalStorage data isn’t sent to server when HTTP request is made. Data in LocalStorage has no expiration time. It can be removed via JS or by clearing the browser’s cache.

1. JSON

Transfer data to and fro API with JSON format.

{

“origin”: “New York”,

“destination”: “London”

}

1. AJAX

AJAX gives you the ability to make web requests even after the web page you are on has loaded from any server. You can query your own server or 3rd party servers.