JavaScript Notes:

Variables can be used to save information into memory and then passed to somewhere else in your webpage.

within <script> element:

var *sock_color* =”Blue”;

var *shoe_color*=”Brown”;

alert(“I am wearing “ + *sock_color* + “ socks with ”+ *shoe_color* + “ shoes.”);

➔I am wearing Blue socks with Brown shoes.

with events:

<h1 onmouseover=”this.innerHTML=*sock_color* + ‘ socks with ‘ + *shoe_color* + ‘shoes’”>My Feet</h1>

➔Blue socks with Brown shoes

This keyword should be used when the element handling the event is the element changed.

Event Handlers

<h1 onmouseover=”this.style.color=’pink’” id=”test1”> My shirt </h1> ➔ My shirt

<h1 onmouseover=”this.innerHTML=’Happy Birthday’;this.style.color=’red’”> Happy </h1>➔ Happy Birthday

<h1 onmouseover=”this.style.color=getElementById(‘test1’).style.color”> My shorts ➔My shorts

getElementById is used when the element handling the event is different from the element changed.

getElementById can be used to copy info from one element to another.

getElementById relies on the id attribute. The referring id must exist for a call to getElementById to complete successfully.

shorthand for **getElementById(‘dog’)** is **dog**

body is used when the entire webpage changed.

Events: actions that automatically trigger reactions within your webpage.

Three mouse events covered in this course: omouseover, onmouseout and onclick.

Event attributes can call JavaScript functions or run JavaScript code.

Here is a link to more information on events https://www.w3schools.com/js/js_events.asp

Javascript:

1. Changing HTML style
    a. Select the object to be changed **(this, getElementById, body)**
    b. Style
    c. CSS property
    d. New Value**(text, another element, variable)**

Examples: this.style.backgroundColor='blue' or getElementById('elem').style.backgroundColor='blue' or

this.style.backgroundColor=*mycolor* where *mycolor* is a variable

2. Changing HTML content
    a. Object to be changed
    b. innerHTML
    c. New Value

Examples: this.innerHTML='Hello' or getElementById('elem').innerHTML='Hello'

3. Changing attribute value of an HTML element
    a. Object to be changed
    b. Attribute name
    c. New value

Examples: this.src='Happy.jpg' or getElementById('**smile**').src='Happy.jpg' or **smile**.src="Happy.jpg"

4. Changing attribute value of an HTML element to another HTML element
    a. Object to be changed
    b. Attribute name
    c. Object to be copied

Examples: this.src= getElementById('**smile**').src or getElementById(**'jump**').src=this.src or

jump.src=this.src

Next we will start JavaScript functions.

Programming with JavaScript functions

Why do we use functions:

1. Actions that are repetitive can be coded once and then called multiple times.
2. Perform a unique action
3. Good programming practices to organize code
4. Easier to make changes to your webpage and eliminate possible errors.
5. Security. Hide code in a function.
6. Programming and Logical Flow
7. Functions can call functions.

Defining a function:

1. within the <script> element.
2. Keyword **function** starts the definition
3. Parts of a function:
    a. Unique Name. Cannot be JavaScript reserved word
    b. Optional Input parameters enclosed by paranthesis()
        i. Input parameters are labels assigned by you. Case Sensitive
        ii. Unique labels beginning with alpha and can contain numbers and underscore
        iii. Good practice is not to use already assigned id's. This can be confusing
    c. The body enclosed by curly brackets {}
        i. Local variables
        ii. Body can hold any JavaScript code we have already reviewed with event handlers
        iii. Can contain multiple JavaScript Code statements separated by semi-colons;
        iv. Update and change HTML elements within webpage
    d. Optional **return** output: only examples. Labs will not focus on these.

Calling/Invoking a JavaScript Function within HTML event attributes

<h1 onmouseover=”change_color();”> Meow </h1>

<h1 id=”dog”> Ruff </h1>

Is the same as……

<h1 onmouseover=” document.getElementById(‘dog’).style.color=’purple’;”> Meow </h1>

<h1 id=”dog”> Ruff </h1>

Examples

```
<script>
// this is a function with no inputs
function change_color (){
document.getElementById(‘dog’).style.color=’purple’;
}
// this is an example of a function with inputs
function change_colors(myId, newColor){
document.getElementById(myId).style.color=newColor;
}
//this is an example of a function with variables and input parameters
function swap_image(id1, id2){
var temp;
temp=document.getElementById(id1).src;
document.getElementById(id1).src=document.getElementById(id2).src;
document.getElementById(id2).src=temp ;
}
 <script>
```