

MPT Web Design



Week 4: Introduction to Javascript

What will we do in this class?

- JavaScript + CSS
- Recap on what we learnt last week in JavaScript
 - Syntax
 - DOM Model
 - What it can do
 - Why we would we use it

Class Plan

- WEEK 1
 - Introduction to HTML, Page Structures/layouts, Browsers
 - Homework: Make a HTML page for your competition page. Come up with a design plan
- WEEK 2
 - Introduction to CSS and how to apply style to a web page
 - Homework: Colour and design scheme for your competition page using paletton
- WEEK 3
 - Introduction to JavaScript and dynamic web pages
 - Homework:
- WEEK 4
 - More JavaScript
 - Homework:
- WEEK 5
 - HTML, CSS, JavaScript, complete portfolio site and competition site



JavaScript Recap

Where can we write our JavaScript?

- In the **head** of web page in `<script>` tags
- Or in an **external file** which we link to using `<script>` tag. (similar to CSS)

```
<!DOCTYPE html>

<html lang="en">

  <head>
    <meta charset="utf-8">

    <title>Page Title</title>
    <meta name="description" content="A basic HTML template to use">
    <meta name="author" content="Aaron Bolger">

    <script type="text/javascript">
      alert("Hello From JavaScript");
    </script>
  </head>

  <body>
    <!-- Comments look like this -->
    <h1>HTML Template</h1>
    <p>A basic HTML layout to use for future sites</p>
  </body>

</html>
```

```
<html lang="en">

  <head>
    <meta charset="utf-8">

    <title>Page Title</title>
    <meta name="description" content="A basic HTML template to use">
    <meta name="author" content="Aaron Bolger">

    <script type="text/javascript" src="my/file/path.js"></script>
  </head>

  <body>
    <!-- Comments look like this -->
    <h1>HTML Template</h1>
    <p>A basic HTML layout to use for future sites</p>
  </body>

</html>
```

What are some Golden Rules of JavaScript?

- Keyword **function**
- Columns and indents aren't *required*
- Instead code needs to be in **curly braces**
 - `function myFunc() {code}`
- All lines must end with a **semicolon ;**
- Keyword **var** for declaring variables
 - `var myInt = 0;`
 - `var myString = "Hello World";`
- Note Structure of For Loop

```
for (var i = Things.length - 1; i >= 0; i--) {  
    Things[i]  
};
```


Spot the Syntax Error

6

7

```
var x = 10;
```

8

```
var y = 5;
```

9

```
z = x + y;
```

10

```
5  
6 def myFunction()  
7       
8     alert("Hello World");  
9 }  
10  
11
```

```
5  
6  function myFunction(){  
7  
8      alert(Hello World)  
9  }  
10  
11
```

```
4
5
6  function myFunction(arg1, arg2){
7
8      if(arg1 = 5){
9          return arg1 + arg2;
10     }
11     else() {
12         return arg1 - arg2;
13     }
14 }
15
16
```

```
4
5  function mySumFunction(arg1, arg2){
6
7      if(arg1 <= 5){
8          return arg1 + arg2;
9      }
10     else {
11         return arg1 - arg2;
12     }
13
14
15  function myAlertFunction(arg1, arg2){
16
17      if(arg1 = 5){
18          alert(arg1 + arg2)
19      }
20     else {
21         alert(arg1 - arg2);
22     }
23 }
24
```

What is the DOM Model?

Why do we use it?

- To make out pages interactive
- We need to use the DOM Model (Document Object Model) to access elements (tags)

DOM Example

HTML

```
<body>
  <!-- Comments look like this -->
  <h1>HTML Template</h1>
  <p>A basic HTML layout to use for future sites</p>

  <div id="demo" >
    Hello HTML
  </div>
</body>
```

JavaScript

```
document.getElementById("demo").innerHTML = "Hello JavaScript";
```

Note the use of id

Div id="demo" and document.getElementById("demo")

Note .innerHTML

What are events?
Why do we use them?

HTML **events** are "things" that **happen** to HTML **elements**. Events can be **caused** by the **user** or by the **browser**. JavaScript can **"react"** on these **events**.

Can you list some events?

Event

Description

onchange

An HTML element has been changed

onclick

The user clicks an HTML element

onmouseover

The user moves the mouse over an HTML element

onmouseout

The user moves the mouse away from an HTML element

onkeydown

The user pushes a keyboard key

onload

The browser has finished loading the page



JavaScript + CSS

JavaScript + CSS

- By using Javascript and the DOM we can change or manipulate the style of of our webpage.

For Example:

```
document.getElementById(id).style.property = new style
```

What Style Attributes can we change?

- All of them!
 - Colour
 - Images
 - Sizing
 - Fonts
 - Position
 - Visability

What else? (think back to the week we looked at CSS)

CSS



CSS Recap

CSS3 Animations

- CSS3 animations allows animation of most HTML elements
- An element's style can be gradually animated over time
- To do this we must specify keyframes (discussed more in web animation)
- Simply keyframes mark a change in something over time
- Every keyframe must be bound to an element



CSS3 Animation Examples

```
/* The animation code */
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

```
/* The animation code */
@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

CSS3 Transitions

- CSS3 transitions work differently to animations
- An element's style can be gradually animated over time in the same manner, but it's used in a different way
- We usually use “Events” to fire transitions
- We have to specify the time that the transition will take and the type of transition
- We then specify the changes in the “Event” in CSS

CSS3 Transition Example

```
a
{
  text-decoration: none;
  color: #FFF;
  font-family: Verdana, Geneva, sans-serif;
  margin: 10px;
  padding: 10px;
  display: block;
  background-color: #CCC;
  -webkit-transition: all 1s ease-in 0s;
  -moz-transition: all 1s ease-in 0s;
  -ms-transition: all 1s ease-in 0s;
  -o-transition: all 1s ease-in 0s;
  transition: all 1s ease-in 0s;
}
a:hover {
  background-color: #9C0;
```

```
display: block;
background-color: #CCC;
-webkit-transition: all 1s ease-in 0s;
-moz-transition: all 1s ease-in 0s;
-ms-transition: all 1s ease-in 0s;
-o-transition: all 1s ease-in 0s;
transition: all 1s ease-in 0s;
}
a:hover {
  background-color: #9C0;
  -webkit-transform: scale(1.5);
  -moz-transform: scale(1.5);
  -ms-transform: scale(1.5);
  -o-transform: scale(1.5);
  transform: scale(1.5);
}
</style>
</head>
```

Resources:

W3Schools JavaScript + CSS: https://www.w3schools.com/js/js_htmldom_css.asp

Events in JavaScript: http://www.w3schools.com/js/js_events.asp

DOM: http://www.w3schools.com/js/js_htmldom.asp