



CSYM019

Internet Programming

Dr Muawya Eldaw
muawya.eldaw@northampton.ac.uk



Week 5

- Quick recap of the last week's topic
- Javascript events
- Setting CSS properties with Javascript
- Moving elements
- Detecting which key is pressed

Javascript

- Javascript is included on web pages using the `<script>` tag
- The `<script>` tag is placed inside the `<head>` element
- The `src` attribute is used to reference the name of a file containing your javascript code

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Web Page!</title>
    <script src="script.js"></script>
  </head>

  <body>
    <h1>Page heading</h1>
    <p>Page content</p>
  </body>
</html>
```

<script> tag

- HTML files are processed from the top to the bottom
- This means the <head> and <script> tags are loaded before the <body> element

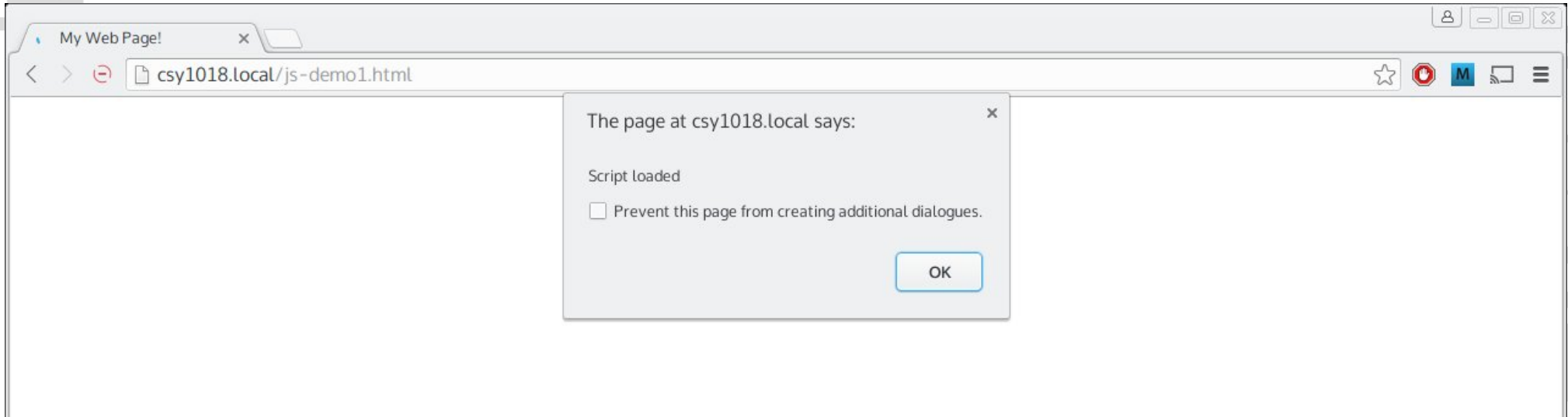
```
<!DOCTYPE html>
<html>
  <head>
    <title>My Web Page!</title>
    <script src="script.js"></script>
  </head>

  <body>
    <h1>Page heading</h1>
    <p>Page content</p>
  </body>
</html>
```

Javascript

- If you add some code to the javascript file it is run before the elements on the page exist

```
alert('Script loaded');
```



Javascript

- Because of this, if you want to write code that uses the elements in some way (e.g. changing the content) you must run the code to do this **after** the page is loaded

Functions

- You can *label* a block of code using a *function*
- This will store the code for later use where it can be referenced and run
- This allows you to write code out of sequence

```
function scriptLoaded() {  
    alert('Script loaded');  
}  
  
function addition() {  
    var num1 = 5;  
    var num2 = 6;  
  
    var num3 = num1 + num2;  
  
    alert(num3);  
}
```

Functions

- The syntax for a function looks like this:

Function keyword

Function name (chosen by you, can be anything)

Opening and closing brackets

```
function scriptLoaded() {  
    alert('Script loaded');  
}
```

Code to run
(As many lines as you like,
Between braces { and })

Functions

- When code is stored inside a function it is not executed it is only defined

```
function scriptLoaded() {  
    alert('Script loaded');  
}  
  
function addition() {  
    var num1 = 5;  
    var num2 = 6;  
  
    var num3 = num1 + num2;  
  
    alert(num3);  
}
```

- This will not display either alert box!

Selecting elements in Javascript

- Javascript contains functions for selecting HTML elements so you can change properties on the (css, attributes, etc)
- The simplest way is to give an element an ID in the HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Web Page!</title>
    <script src="script.js"></script>
  </head>

  <body>
    <h1 id="pageheading">Page heading</h1>
    <p>Page content</p>
  </body>
</html>
```

Selecting elements with Javascript

- Once an element on the page has an ID, you can use the javascript function `document.getElementById()` to select it and store the *element* in a variable

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Web Page!</title>
    <script src="script.js"></script>
  </head>

  <body>
    <h1 id="pageheading">
      Page heading
    </h1>
    <p>Page content</p>
  </body>
</html>
```

```
var element = document.getElementById('pageheading');
```

Selecting elements

- Once you have an element you can make changes to it
- E.g. to update the content you can use:

```
var element = document.getElementById('pageheading');  
element.firstChild.nodeValue = 'New Heading';
```

Javascript

- Because this code requires the elements to exist on the page, it must be run after the page has loaded
- This can be done in two steps:
 - 1) Move the code into a function so it is not run immediately
 - 2) Tell the browser to call the function when the page loads

Javascript

- 1) Move the code you want to run into a function that get executed only when the page has loaded.

```
function myLoadFunction() {  
    var element = document.getElementById('pageheading');  
  
    element.firstChild.nodeValue = 'New Heading';  
}
```

- Note: This function can have any name you like!

Javascript

2) Inform the browser you want to run this function after the page has loaded

- This can be done using the inbuilt function `document.addEventListener()`

```
function myLoadFunction() {  
    var element = document.getElementById('pageheading');  
    element.firstChild.nodeValue = 'New Heading';  
}  
  
document.addEventListener('DOMContentLoaded', myLoadFunction);
```

DOMContentLoaded
means when the content on the
page is loaded (the elements exist)

The name of the function

Javascript

- `addEventListener()` is a very useful function
- It allows you to run a function when a specific *event* occurs

```
document.addEventListener('DOMContentLoaded', myLoadFunction);
```

When this happens

Run the function with this name

Click events

```
document.addEventListener('click', myLoadFunction);
```

- In last week's exercise 5, if you click **anywhere** on the document, the contents of both elements will be updated
- It's possible to assign a click event to a particular element
- However, the code that associates the event with the element also has to be done **after** the page has loaded

Click Events

- You can call *element.addEventListener()* to add an event to a specific element
- This works exactly the same way as `document.addEventListener()` however it will only call your function when that particular element is clicked on

```
function myClickFunction() {  
    var element = document.getElementById('pageheading');  
    element.firstChild.nodeValue = 'New Heading';  
}  
  
function myLoadFunction() {  
    var element = document.getElementById('pageheading');  
    element.addEventListener('click', myClickFunction);  
}  
  
document.addEventListener('DOMContentLoaded', myLoadFunction);
```

The click function updates
the contents of the h1
element

When the page loads,
run myLoadFunction

When the load function is run,
a click event is added to the h1
Element so that
myClickFunction is run when the
h1 is clicked on

Exercise 1

- 5-10 Minutes
- 1) Download Topic5-Exercise1.zip and extract it
- 2) Add a `<script>` tag to the page so you can run some javascript
- 3) Add an event listener for `DOMContentLoaded` and a function that it runs when the page loads
- 4) In the load function you created, add a click event to the element with the ID ``circle`` so that when the circle is clicked an alert box appears and says "The button was pressed"
- **Note: You will need this for the rest of today's exercises**

Setting CSS Properties

- You can set CSS properties on an element using javascript
- Firstly you have to select the element by using `document.getElementById`
- Once you have a reference to the element in a variable you can change the CSS on it using a *statement* such as the one shown here.

```
element.style.propertyName = 'propertyValue';
```

Setting CSS Properties

- E.g. to set the width and height of an element

```
element.style.width = '50px';  
element.style.height = '50px';
```

- Or Text colour:

```
element.style.color = '50px';
```

- Any CSS property can be used and set
- However, the names are slightly different for some properties

CSS Properties

- Some CSS properties contain hyphens, e.g.
 - background-color
 - border-radius
 - font-family
- In Javascript these are written by removing the hyphen and making the first letter of the second word uppercase:

```
element.style.backgroundColor = 'green';  
element.style.borderRadius = '50px';  
element.style.fontFamily = 'Verdana, Helvetica, Sans-serif';
```

CSS Properties

- The value is placed in quotes as it is a string, however the outcome is the same

```
var element = document.getElementById('paragraph');  
element.style.backgroundColor = 'green';  
element.style.borderRadius = '50px';  
element.style.fontFamily = 'Verdana, Helvetica, Sans-serif';  
element.style.height = '50px';
```

- Will have the same effect as the CSS

```
#paragraph {  
  background-color: green;  
  border-radius: 50px;  
  font-family: Verdana, Helvetica, Sans-serif;  
  height: 50px;  
}
```


CSS in Javascript

- The javascript code is longer but it can be run at **any time**. CSS is applied once, when the page loads
- Javascript can be used to change the look of an element after the page has loaded
- Usually this is useful when an event occurs

Exercise 2

- < 5 minutes
- 1) Building on exercise 1, amend the code so that when the button is clicked its background colour is set to blue