CS7025 Programming for Digital Media

Lesson 4 – DOM Manipulation

Semicolon

In Javascript the semicolon ';' is optional. It is good practice however to end a statement with one, to increase legibility of your code and to avoid confusion;

Don't use a semicolon when:

▶ Declaring a function	`function doSomething(){}`
▶ A conditional comparison	`if(true){}else{}`
▶ A loop	`for(let i=0; i<5; i++){}`

Scope

Variables should only be accessible within scope;

```
let students = 31;
function countAbsentStudents(){
    let absent = 3;
    students -= absent;
}
countAbsentStudents();
console.log("students", students);
console.log("absent", absent); //throws error: absent is not defined
```

Events

Events are the things that are happening on your web page or in your application.

Examples:

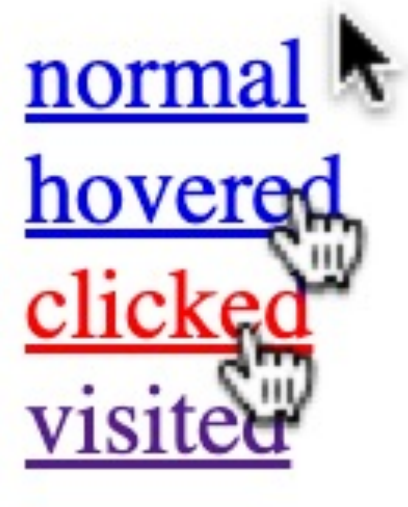
- Loading a page
- Clicking on a hyperlink
- Entering text in a text field / pressing keys on the keyboard
- Checking a checkbox
- Submitting a form

State

State goes hand in hand with events. Events usually alter the state of your web page or application. State is whatever describes the UI at a given point in time or when you are in control of your own states, a specific stage of a process.

In the case of a hyperlink for instance, it can be

- 'normal' the link is there but no interaction
- 'hovered' the mouse cursor is over the link
- 'clicked'
- 'visited'



The Document Object Model

The Document Object Model (DOM) represents the structure of a document (HTML).

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Document</title>
    </head>
    <body>
    </body>
</html>
```

Manipulating the DOM

Let the browser wait with executing code within scope until the document is completely loaded into memory

```
document.addEventListener("DOMContentLoaded", () => {
    //do things
});
```

Manipulating the DOM

Traversing the DOM

```
<body>
    <div id="shopping_list">
        <ul id="items">
            <li class="item">item 1</li>
            <li class="item">item 2</li>
            <li class="item">item 3</li>
        </ul>
    </div>
</body>
```

`li.item` is a child of `ul#items` which is a child of `div#shopping_list` which is a child of `body`

Manipulating the DOM

https://developer.mozilla.org/en-US/docs/Web/API/Document

The dot notation is used to traverse through the DOM

```
document.querySelector('#theSelector');
```

In English:

"Of the document object, use the query selector for the element with an ID of 'theSelector' "

Manipulating the DOM

Different ways of addressing elements within the DOM

```
getElementById()
// <div id="student"></div> document.getElementById('student');

getElementsByClassName()
// <div class="student"></div> document.getElementByClassName('student');

getElementsByName()
// <input type="text" name="up" /> document.getElementsByName('up');

getElementsByTagName()

// <h1>A Title</h1>  document.getElementsByTagName('h1');

querySelector()
// <div id="student"></div> document.querySelector('#student');

querySelectorAll()
// <div class="student"></div> <div id="location">…</div>
// document.querySelectorAll('.student #location');
```

Manipulating the DOM

Adding and removing elements

The way to add new elements to your document is by creating a new element (store it in memory)

```
let paragraph = document.createElement("p");
```

```
paragraph.textContent = "Hello World!";
```

And then add it to the document (make it available)

```
document.body.appendChild(paragraph);
```

To remove an element, use the `remove()` function:

```
document.getElementById(<THE ID>).remove();
```

Manipulating the DOM

Attributes

```
<div id="students" class="cs7025" data-counter="3">…</div>
```

Get the value of an attribute by invoking the `getAttribute(attribute)` function

Set the value of an attribute by invoking the `setAttribute(name, value)` function

```
document.getElementById('students').setAttribute('id','teachers');
```

Manipulating the DOM

Read data from an external JSON file

```
let dataSource = "externalFile.json";

fetch(dataSource)
    .then((response) => {
        return response.json();
    })
    .then((data) => {
        // now you have the json data, so this is the place where
        // you can work with it
    })
    .catch((err) => {…});
```

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Manipulating the DOM

Read data from an external API

```
let dataSource = "https://newsapi.org/v2/everything?q=<TOPIC>&from=<YYYY-MM-
DD>&sortBy=publishedAt&apiKey=<API_KEY>";

fetch(dataSource)
    .then((response) => {
        return response.json();
    })
    .then((data) => {
        // now you have the json data, so this is the place where
        // you can work with it
    })
    .catch((err) => {…});
```

jQuery

write less, do more.

Third party library jQuery which uses css selectors to address elements in the DOM

https://jquery.com/

```
document.querySelector('.student')
```

```
$('.student')
```

```
button.addEventListener('click',
(event)=>{
    …
});
```

```
$(document).on('click','button',
()=>{
    …
});
```

Try it yourself

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Thank You

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin