CHT2520 Advanced Web Programming

Matthew Mantle m.e.mantle@hud.ac.uk

No Classes Next Week

- All classes next week (w/c 2nd December) have been cancelled
- Replacement sessions will be run in w/c 6th January
  - Hand-in the assignment on 10th January
  - These sessions are optional (i.e. not attendance monitored)

Today's Session - JavaScript

JavaScript

- JavaScript is a programming language that runs in the browser and adds interactivity to web pages
- Interactivity? Changing the page after it has loaded
  - Changing CSS e.g. hiding/showing elements
  - Changing the HTML content adding/removing elements
- Improves the user experience

JavaScript - Some key points

- JavaScript has the usual programming features - variables, conditionals, arrays, functions, objects etc.
- Variables are declared using `const` or `let`
    - `const` for varables that won't change value
    - `let` for variables that can

```
let decade = 2010;
if(decade === 2000){
  updateFilmList(filmsFrom2000s);
}else if(decade === 2010){
  updateFilmList(filmsFrom2010s);
}
```

JavaScript - Some key points

- You may not have seen arrow function expressions before

```
const showColour = (col)=>{
  console.log(col);
}
```

- Is equivalent to:-

```
function showColour(col){
  console.log(col);
}
```

- `console.log()` displays a message in the browsers console
    - Useful for debugging

JavaScript - Some key points

- We often use arrow function expressions as anonymous functions

```
const colours = ['red', 'green', 'blue'];
colours.forEach((col) => {
  console.log(col);
});
```

JavaScript in the Browser - `querySelector()`

- In a browser we can use JavaScript to get hold of parts of the HTML page e.g. `document.querySelector()`

```html
<div id="filmListHolder">...
```

```javascript
const filmListHolder = document.querySelector("#filmListHolder");
```

JavaScript in the Browser - `querySelectorAll()`

- `querySelectorAll()` can be used to select multiple elements

```html
<div id="decadeNavHolder">
  <a href="/films" class="decade-link">2000</a>
  <a href="/films" class="decade-link">2010</a>
</div>
```

```javascript
const decadeLinks = document.querySelectorAll(".decade-link");
```

JavaScript in the Browser - Changing Content

- We can use `innerHTML` to access the content of an element

```html
<button id="btn">Nice Button</button>
```

```javascript
const btn = document.querySelector("#btn");
console.log(btn.innerHTML); // Nice Button
```

- We can also use it to change the content of an element

JavaScript in the Browser - Changing Content

- We can create elements and insert them into the HTML page.

```
const filmHolder = document.querySelector("#filmHolder");
const filmPara = document.createElement("p");
filmPara.innerHTML = "Jaws (1975)";
filmHolder.appendChild(filmPara);
```

- Would generate

```
<div id = "filmHolder">
  <p>Jaws (1975)</p>
</div>
```

JavaScript in the Browser - Events

- We can listen for user events e.g. clicking on a button

```
<button id="btn">Nice Button</button>
```

```
function doSomething(){
  console.log("You clicked the button");
}
const btn = document.querySelector("#btn");
btn.addEventListener("click", doSomething, false);
```

JavaScript in the Browser - Events

- The event object can tell us about the event

```
<button id="btn">Nice Button</button>
```

```
function doSomething(evnt){
  //You clicked <button id="btn">Nice Button</button>
  console.log("You clicked: "+event.target);
}
const btn = document.querySelector("#btn");
btn.addEventListener("click",doSomething,false);
```

JavaScript - Loading External Data

- We can use `fetch` to load external data

```
async function getFilms(decade) {
  const url = "./data/" + decade + ".json";
  try {
    const response = await fetch(url);
    const films = await response.json();
    console.log(films);
  } catch (error) {
    console.error(error.message);
  }
}
```

- Fetch is asynchronous and promise based

Loading External Data - JSON

- Often data loaded into JS is structured as JSON (JavaScript Object Notation)
- It looks like JavaScript arrays/objects (isn't exactly the same)

```
[
    {"id":4, "title": "The Incredibles", "year":2004},
    {"id":7, "title":"Spirited Away", "year":2001},
    {"id":13, "title": "Mean Girls", "year":2004}
]
```

- JSON **isn't** part of the JavaScript language
    - It is simply a way of structuring data so it can be exchanged between different languages/applications.

Loading External Data - JSON

- We can get Laravel to provide a JSON response

```
function index()
{
    $films = Film::all();
    return response()->json($films);
}
```

- Our JavaScript apps can then be database driven
- JSON provides a common format for transferring data from a back-end to JavaScript.

Learning JavaScript

- There are lots of resources on the web
  - e.g. https://eloquentjavascript.net/ is very good
- JavaScript is a language that has changed a lot over the years
  - Should be looking at ES2015
  - Avoid jQuery

Practical Work

- Using some JavaScript in the Films App
  - First as a front-end only example
  - Then integrating with Laravel