# Comp 324/424 - Client-side Web Design

Spring Semester 2017 - Week 8

Dr Nick Hayward

# Contents

- DEV Week presentation and demo
- DEV Week peer review
- AJAX and JSON

brief presentation or demonstration of current project work

- ~ 5 to 10 minutes per group

- analysis of work conducted so far
  - *eg: during semester & DEV week*

- presentation, demonstration, or video overview...
  - *outline current state of web app*
  - *show prototypes and designs*
  - *explain what works & does not work*
  - *anything else considered relevant to your research or development...*

# DEV Week peer review

- take notes for each presentation
- complete peer review per project presentation
  - *individually, **NOT** as a group*
- peer review **MUST** be completed this week
  - *part of this week's assignment*
- collated, anonymised results sent to each group
- carefully review peer review results
  - *consider modifications, updates...*

# AJAX and JSON - part 1

- AJAX is a simple way to load data
  - *often new or updated data*
  - *into a current page without having to refresh the browser window*

- common form of data for work with AJAX is JSON

- many common usage scenarios and examples for AJAX
  - *autocomplete in forms*
  - *live filtering of search queries*
  - *real-time updates for content and data streams*

- also use AJAX to help us load data behind the scenes
  - *preparing content for our users before a specific request is received*
  - *helps to speed up page responses and data load times*

- AJAX uses an asynchronous model for processing requests

- user can continue to perform various tasks, queries, and work
  - *whilst the browser itself continues to load data*

- inherent benefit of AJAX should include
  - *a more responsive site, intuitive usage and interface experience*

# AJAX and JSON - part 2

*asynchronous model*

- traditional synchronous model normally stops a page
  - *until it has loaded and processed a requested script*

- AJAX enables a browser to request data from the server
  - *without this synchronous pause in usage*

- AJAX's **asynchronous processing model**
  - *often known as **non-blocking***
  - *allows a page to load data and process user's interactions*

- server responds with the requested data
  - *an event will be fired by the browser*
  - *event can then call a function to process the data*
  - *often JSON, XML, or simply HTML*

- browser will use an **XMLHttpRequest** object to help handle these AJAX requests

- browser will not wait for a response

# JSON and jQuery - get a file - part I

*initial setup*

- try some AJAX with a JSON file

```
{
  "country":"France",
  "city":"Marseille"
}
```

- save this content to our `docs/json/trips.json` file

- run on a server, local or remote
  - *browser security restrictions for JavaScript*
  - *local server such as XAMPP, Python's SimpleHTTPServer, Node.js...*

```
python -m SimpleHTTPServer 8080
```

- initially use the `getJSON()` function to test reading this content

```
$.getJSON("docs/json/trips.json", function(trip) {
  console.log(trip);
});
```

- console output is expected JSON object

```
Object { country: "France", city: "Marseille" }
```

# JSON and jQuery - get a file - part 2

- now use this return object to load our data as required within a site

```
//overall app logic and loader...
function loadJSON() {
  "use strict";

  $.getJSON("docs/json/trips.json", function(trip) {
    //element for trip data
    var $tripData = $("<p>");
    //add some content from json to element
    $tripData.html(trip.city + ", " + trip.country);
    //append content to .note-output section
    $(".note-output").append($tripData);
  });
};

$(document).ready(loadJSON);
```

- DEMO - AJAX 1 - AJAX - demo 1

# JSON and jQuery - get a file - part 3

*array for trips...*

- need to store multiple trips
  - *multiple countries, multiple cities, and so on...*

- need to work with JSON arrays
  - *update `trips.json` file for cities*

```json
{
    "cities": [
        {
            "name": "Marseille",
            "region": "Provence-Alpes-Côte d'Azur"
        },
        {
            "name": "Paris",
            "region": "Île-de-France"
        }
    ]
}
```

# JSON and jQuery - get a file - part 4

*load an array for trips...*

- update JavaScript to load array and set data as required

```javascript
//overall app logic and loader...
function loadJSON() {
  "use strict";

  $.getJSON("docs/json/trips.json", function(trips) {
    //element for trip data
    var $cityData = $("<ul>");

    //iterate over cities array - trips.cities...
    var $cities = trips.cities;
    $cities.forEach(function (item) {
      var $city = $("<li>");
      $city.html(item.name + " in the region of " + item.region);
      $cityData.append($city);
    })
    //append list to .note-output
    $(".note-output").append($cityData);
  });
};


$(document).ready(loadJSON);
```

- DEMO - AJAX 2 - AJAX - demo 2

# Demos

## AJAX

- DEMO 1 - AJAX - demo 1
- DEMO 2 - AJAX - demo 2

# References - JS & Libraries

- jQuery
  - *jQuery*
  - *jQuery API*
  - *jQuery - .getJSON()*

- Lint options
  - *JSLint - JavaScript Validator*
  - *JSONLint - JSON Validator*

- MDN
  - *MDN - JS*
  - *MDN - JS Objects*

- W3
  - *W3 - JS Object*