Introduction

Module – Advanced JS

Dr. Andrew Errity

Aim

- To provide the students with the theoretical knowledge and practical skills required to design, develop, and test client-side web applications.

Learning outcomes

1. Design, develop and test client-side web applications using JavaScript.
2. Understand the differences in the implementation of JavaScript by the major web browsers in current usage and to create client-side web applications that work across all browsers.
3. Evaluate the JavaScript libraries and frameworks currently available against an application's requirements, and to utilize those libraries in developing client-side web applications.

Learning outcomes

4. Apply modern software engineering techniques, such as the Model-View-Controller architecture and asynchronous programming, in the design and implementation of client-side web applications.
5. Integrate client-side and server-side web applications using best practice programming techniques, including the use of XML, AJAX, JSON and web services.

Indicative topics

JS programming fundamentals

- Variable scope/scope chain.
- Objects and arrays (& iteration).
- Functions, methods, and functions as objects.
  - Closures.
- Classes, constructors and prototypes.
  - Inheritance.
- Modules and namespaces.

Indicative topics

JS web programming

- Manipulating the DOM.
  - Manipulating content, attributes and styles.
  - Selecting and filtering HTML/DOM content.
- Events and event handling.
- Form validation and security/privacy.
- User interface components.
  - drag-and-drop, sorting, pagination, accordions, date and time pickers, dialogs, tabs.
- Client-side graphics – canvas, etc.
- Asynchronous HTTP requests, XML, JSON and AJAX.

Indicative topics

JS web programming

- Developing client-side web applications with the Model-View-Controller architecture, asynchronous event-handling, XML and JSON data handling, web services.
- Integrating client and server-side web applications.
- Testing and debugging client-side web applications.
- JavaScript libraries and frameworks.
  - Angular 2/React/Vue
- Bundling and deploying JS web apps.

Assessment

- CA1 – Develop a JS application [25%]
- CA2 – Develop a modern web app using a JS Framework [25%]
- Exam – 2 hour written exam [50%]

Number of CAs may change…

History of JS

- 1995: JavaScript is born as LiveScript
- 1997: ECMAScript standard is established
- 1999: ES3 comes out and IE5 is all the rage
- 2000–2005: XMLHttpRequest, a.k.a. AJAX, gains popularity in apps such as Outlook Web Access (2000), Gmail (2004) and Google Maps (2005).
- 2009: ES5 comes out (this is what most of us use now) with forEach, Object.keys, Object.create, and standard JSON
- 2015: ES6/ECMAScript2015 comes out

https://webapplog.com/es6/

Books

Crockford, D. (2008). *JavaScript: the good parts*. Sebastopol, C.A.: O'Reilly.

Flanagan, D. (2011). *JavaScript the definitive guide* (6th ed.). Cambridge: O'Reilly.

Haverbeke, M. (2014). *Eloquent JavaScript: a modern introduction to programming* (2nd ed.). San Francisco: No Starch Press.

Osmani, A. (2013). *Learning JavaScript design patterns*. Sebastopol, CA: O'Reilly Media.

Resig, J., Bibeault, B., & Maras, J. (2016). *Secrets of the JavaScript ninja* (2nd ed.). Greenwich: Manning.

Simpson, K. (2015). *You don't know JS: up & going*. O'Reilly Media.

Resources

- JavaScript: The Good Parts – Online course
  - https://www.pluralsight.com/courses/javascript-good-parts
- You Don't Know JS – Online book series
  - https://github.com/getify/You-Dont-Know-JS
- Code Academy – Online course
  - https://www.codecademy.com/learn/introduction-to-javascript
- Mozilla JS docs
  - https://developer.mozilla.org/en-US/docs/Learn/JavaScript

Problem: DesignTech

- Write a JS program that prints the numbers from 1 to 100. But for multiples of three print “Design” instead of the number and for the multiples of five print “Tech”. For numbers which are multiples of both three and five print “DesignTech”

```
1
2
Design
4
Tech
Design
7
8
Design
Tech
11
Design
13
14
DesignTech
16
17
```

Solutions: DesignTech

```
for (let i = 1; i <= 100; i++) {
    if (i % 15 === 0)
        console.log("DesignTech");
    else if (i % 3 === 0)
        console.log("Design");
    else if (i % 5 === 0)
        console.log("Tech");
    else
        console.log(i);
}
```

```
for (let i = 1; i <= 100; i++) {
  let d = i % 3 === 0, t = i % 5 === 0;
  console.log(d ? t ? "DesignTech" : "Design" : t ? "Tech" : i);
}
```

Problem: PasswordStrength

- Write a web page that allows a user to enter a password. If the password length is less than 6 the text should be red, if the length is between 7 and 10 the text should be orange, and if the length is longer than 10 make the text green.

Solution: PasswordStrength

```
<form>
  <input class="pw" type="text" maxlength="64" placeholder="Enter password">
</form>
```

```
$( ".pw" ).keyup(function() {
   console.log( this.value );
  if(this.value.length < 6)
    $( ".pw" ).css("color", "red");
  else if(this.value.length <= 10)
    $( ".pw" ).css("color", "orange");
  else
    $( ".pw" ).css("color", "green");
});
```

```
form {
  margin-bottom: 25px;
  padding: 15px;
  background-color: cyan;
  display: inline-block;
  width: 100%;
  max-width: 340px;
  border-radius: 3px;
}

input {
  display: block;
  margin-bottom: 10px;
  padding: 5px;
  width: 100%;
  border: 1px solid lightgrey;
  border-radius: 3px;
  font-size: 16px;
}
```