ECM1417: Web Development

Workshop 07

Diego Marmsoler

Monday, March 7th, 2022

In the following workshop you will learn how to program client side scripts using JavaScript. To this end you will:

- Output text to the browser using JS.
- Work with JS arrays.
- Create and manipulate JS Objects.
- Use closures to restrict access to variables.

1 JavaScript Arrays

In addition to the basic data types, JavaScript provides also a notion of array to organize collections of values. In JavaScript, an array is actually "just" an object that has some array-like characteristics. It converts array subscripts into strings that are used to make properties.

Arrays have their own literal format. For example:

```
var empty = [];
var numbers = [
        'zero', 'one', 'two', 'three', 'four',
        'five', 'six', 'seven', 'eight', 'nine'
];
```

Retrieval and updating of properties work in a similar way as with objects. For example:

```
empty[1]          // 'undefined'
numbers[1]        // 'one'
```

Arrays also have a much more useful set of built-in methods and a property `length`. For example:

```
numbers.push('ten')
empty.length      // 0
numbers.length    // 10
```

2 JavaScript Basics

In this exercise you will use some basic features of JavaScript:

1. First, create a simple JavaScript function `hello` in the header of your HTML document which outputs the string "Hello" followed by the content of a variable x to the console.

2. Then, create a self invoking function which calls `hello` in a way that it prints "Hello World".

3. Finally, ensure that your program also works in strict mode and if not try to change it to make it work in strict mode.

3 JavaScript Functions

For this exercise you should write two JavaScript functions to obtain the maximum value of a sequence of integers:

1. The function `findMax1` takes an array of positive numbers as input and returns the maximum number in the array. For example:

   `findMax([10,2,12,4])=12`

2. The function `findMax2` takes an arbitrary number of positive arguments as input and returns the maximum number. For example:

   `findMax(10,2,12,4)=12`

   or

   `findMax(8,3,12,14,1,7)=14`

4 JavaScript Objects

For this exercise you should create a simple JavaScript application for university students:

- First, create two constructor functions:

  Module which sets the name of a module and an obtained mark.

  Student which sets the name and surname of a student and a list of module objects taken by the student.

- Then, you should create two students with at least two modules taken by both of them.

- Next, you should add a method `average` to the first student which calculates the average mark of the student.

- Now, use the average function of student 1 to calculate the average for student 2 (Do not add the function to student 2 but use an alternative method).

- Finally implement an alternative way to use one average function for both students (Again, do not add the function to student 2 but use an alternative method).

5 JavaScript Functions

So far, modules of a student can easily be changed by everyone which has access to the student object. In this exercise you will create a new student object in which access to the modules property is restricted. To this end, change the implementation in a way that a user of the object should only be able to access the modules of the student via two methods:

addModule Adds a new module to the student

average Calculates the average of all modules of the student.

Test your solution by executing the following code which should return an average of 20:

```
student.addModule(new Module("Web Development", 20));
student.modules=[];
console.log(student.average());
```