MEEC - ADInt

Laboratory 9

In this laboratory students will learn how develop dynamic web pages using JavaScript and JQuery.

1 JavaScript 101

https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript

JavaScript is a programming language that is able to run in the browsers inside a web page.

JavaScript has a syntax close to C, but with some differences

1.1 Types and Variables

New variables in JavaScript are declared using one of three keywords: let, const, or var:

- **let** allows you to declare block-level variables. The declared variable is available from the block it is enclosed in.
- **const** allows you to declare variables whose values are never intended to change. The variable is available from the block it is declared in.
- **var** is the most common declarative keyword. It does not have the restrictions that the other two keywords have. This is because it was traditionally the only way to declare a variable in JavaScript. A variable declared with the var keyword is available from the function it is declared in.

If you declare a variable without assigning any value to it, its type is **undefined**.

JavaScript's types are:

- Number
- String
- Boolean
- Symbol (new in ES2015)
- Object
- Function
- Array

- Date
- RegExp
- null
- undefined

1.2 Numbers

Numbers in JavaScript are "double-precision 64-bit format IEEE 754 values".

The basic operators are available as with any other languages.

More complex math functions are available on the Math object:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math

The **parseInt()** and **parseFloat()** functions parse a **string** until they reach a character that isn't valid for the specified number format, then return the number parsed up to that point.

1.3 Strings

Strings in JavaScript are sequences of Unicode characters.

Strings can be treated as objects with associated methods.

The available methods are described here:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String

1.4 Objects

JavaScript objects can be thought of as simple collections of name-value pairs, similar to Dictionaries in Python.

1.5 Arrays

Arrays in JavaScript work very much like regular objects (numerical properties can naturally be accessed only using [] syntax) but they have one magic property called 'length'. This is always one more than the highest index in the array:

```
var a = ['dog', 'cat', 'hen'];
```

```
a[100] = 'fox';
a.length; // 101
```

1.6 Functions

Functions are defined with the function keyword:

```
function add(x, y) {
  var total = x + y;
  return total;
}
```

If a function is called without passing the parameters it expects, they will be set to **undefined**.

1.7 Control structures

JavaScript has a similar set of control structures to other languages in the C family. Conditional statements are supported by **if** and **else** and can be chained together:

```
var name = 'kittens';
if (name == 'puppies') {
  name += ' woof';
} else
  if (name == 'kittens') {
    name += ' meow';
  } else {
    name += '!';
  }
```

```
var input;
do {
  input = get_input();
} while (notValid(input));
```

```
while (true) {
  // an infinite loop! }
```

```
for (var i = 0; i < 5; i++) {
  // Will execute 5 times
}
```

```
for (let value of array) {
  // do something with value
}
```

```
for (let property in object) {
   // do something with object property
}
```

2 Exercise 2

To open a JavaScript console (where code can be interactively written (just like python) follow these steps on the Chrome browser:

- Select **More Tools** menu
- Select Developer Tools menu
- Select the **Console** tab

It is now possible to execute JavaScript instructions.

Experiment some instructions related to

- Numeric variables declaration and use
- Array and objects creation and manipulation
- String operations

3 JavaScript in the web pages

JavaScript can be executed inside web pages in the browsers

This code is provided to the browsers along the HTML.

To defined the code to run inside a HTML page it is possible to use the <script></script> element.

The JavaScript code can be written inside this element or imported from a file/URL.

The JavaScript code in a web page can access the various HTML elements (buttons, text input, lists, divs) to extract information from them, manipulate them or apply methods.

In order to a HTML element to be accessible from the JavaScript code it is necessary to assign it an identifier.

Identifiers are assigned to HTML element using the id attribute:

```
 <div class="ui calendar" id="standard_calendar">
  <div class="ui input left icon">
    <i class="calendar icon"></i>
    <input type="text" placeholder="Date/Time">
  </div>
</div>
```

From this moment on it is possible to access the various attributes of this object.

These attributes are usually related to the configuration.

To know what attributes are available for such element it is necessary to access its documentations.

3.1 jQuery

https://www.w3schools.com/jquery/default.asp

https://www.tutorialspoint.com/jquery/index.htm

jQuery allows an easy access and manipulation of each of the elements in the HTML page.

If the element contains an identifier (attribute **id=”elementID”**) it is possible to access such element with the jquery selector:

https://www.w3schools.com/jquery/jquery_selectors.asp

https://www.tutorialspoint.com/jquery/jquery-selectors.htm

```
$('#standard_calendar').calendar();
```

This selector also allows to access other elements, and then apply methods to it.

```
$(document).ready(function(){
   $("#buttonShow").hide();
});
```

The previous code defines a function that runs when the whole document is completely loaded/ready.

When the while document is loaded, the button #buttonShow is hidden.

Jquery allows the applications of certain effects to the HTML elements:

- Hide/show
- Fade
- Animate

3.2 Events

https://www.w3schools.com/jquery/jquery_events.asp

https://www.tutorialspoint.com/jquery/jquery-events.htm

When a user interacts with a web pages, events are continuously being generated and handled by the browsers:

- document ready
- click on a button
- move over an element
- change the value of a field
- ...

jQuery allows the registration of callbacks to such events

Callbacks are functions that execute when a certain event occurs:

```
$("#buttonHide").click(function(){
   $("#textToHide").hide()
});
```

This Jquery code configures buttonHide to execute certain code when the user clicks it with the mouse.

The instruction **$("#textToHide").hide()** is only executed when the user clicks on the **buttonHide**

More events and examples of code can be seen here:

https://www.w3schools.com/jquery/jquery_ref_events.asp

The following code exemplifies the programming of button click handlers and the hiding of other elements:

```
<!DOCTYPE html>
<html>
  <head>
    . . .
    <script>
        $(document).ready(function(){
            $("#buttonShow").hide();
            $("#buttonHide").click(function(){
                $("#textToHide").hide()
            });
        });
    </script>
  </head>
  <body>
    <button class="ui button" id ="buttonShow">Show</button>
    <button class="ui button" id ="buttonHide">Hide</button>
    <br><br>
```

```
    <div id="textToHide"> this text will (dis)apear </div>
  </body>
</html>
```

4 Exercise 3

Modify the file **jquery-1.html** so that:

- when the user clicks the button **buttonHide**:
  - the text is hidden
  - the button **buttonShow** is made visible
  - the button **buttonHide** is hidden
- when the user clicks the button **buttonShow**:
  - the text is made visible
  - the button **buttonHide** is made visible
  - the button **buttonShow** is hidden
- Whenever the mouse is inside the text area (** . . . **) a notification should appear (for instance INSIDE) on another div.

5 User input/Output

Some of the HTML elements are static but always contain information/text (, <p>, <table>) while some others allow user input (input).

In these two types of HTML elements it is possible from the JavaScript read the values that are present in such elements and even change them. For other elements such as checkboxes it is possible to query their state

5.1 Retrieving HTML values

There are a set of special methods that can be applied to the retrieve data from element:

https://www.w3schools.com/jquery/jquery_dom_get.asp

The methods .**text()** and .**html()** retrieve the content of regular elements (except input).

The difference between these is the format of the content (plain text or html)

To retrieve the text that the user typed in a input box, it is necessary to use the val() method.

5.2 Retried Fomantic widgets user input

The complex widgets from Fomantic (that need to be initialized with JavaScript) have a special method to retrieve the values that the user inserted.

These widgets have what is call **behaviors** and is accessible with the following generic JavaScript:

```
$('.#widgetID').widgetClass(behavior, argument1, argumentTwo, ...);
```

where:

- widgetID should be replace with the identifier assigner when the widget was created
- widgetClass should be replaces by the name that was used when initializing the widget
- behavior should be replaced by a string dependent of the operation.

The behaviors are related to the state of the widget (selected, enabled, checked, ….)

The documentation for these behaviors can be accessed in each widget page (on section usage→ behavior)

The more relevant are:

- calendar – get date - $("#calendar-1").calendar("get date")
- slider – get value - $("#slider-1").slider("get value")
- dropdown - get value - $("#dropDown-1").dropdown("get value")
- checkbox – is checked - $("#checkbox-1").checkbox("is checked")

Other elements (input, textareas use the regular .val() method to retrieve the typed value.

6 Exercise 4

Open the resolution of the **Exercise 1** on the browser, open the JavaScript Console (as described in Exercise 3) and apply the **.text()**, **.html()** and **.val()** or the **fomantic-ui behavior** methods to every element with an id.

7 Modifying HTML values

The same methods that read the HTML values, can also be used to modify them.

https://www.w3schools.com/jquery/jquery_dom_set.asp

For regular HTML elements the methods **.text()** and **.html()** replace the values that are inside the element tags. Applying .html to a div will replace the whole content that was defined inside . . . .

For input elements, by applying the method **.val()** it is possible yo replace what the user typed.

7.1 Exercise 5

Open the resolution of the **Exercise 1** on the browser, open the JavaScript Console (as described in Exercise 3) and modify the contents of every element with an **id** by applying the **.text()**, **.html()** and **.val()**.

For each change observe the result returned

If necessary reload the page.

8 Exercise 5

Modify the result of exercise 1 so that the when the user clicks the button **button1** the values typed by the user of every input element are put in the **randomText** element.

The text to put in the randomText should be formatted: a <h3> subtitle should separate each value.

https://fomantic-ui.com/modules/calendar.html#/usage

https://fomantic-ui.com/modules/slider.html#/usage

https://fomantic-ui.com/modules/dropdown.html#/usage

https://fomantic-ui.com/modules/checkbox.html#/usage