User Interfaces
Computer Science Engineering
Course 2019 / 2020

Programming exercises
(Assignments)

WEB TECHNOLOGIES – JAVASCRIPT & JQUERY

Table of Contents

Introduction

This document presents the **second set of programming exercises**. These exercises aim to provide you with a hands-on introduction to a script language designed to execute code in web browsers. In particular, we will provide a short introduction to the JavaScript language and describe three programming exercises. This section describes how the exercises will be carried out (estimated effort, date of submission), assessed (assessment criteria), and submitted.

Estimated effort

The estimated number of hours each student will need to devote to completing the exercises is 10, which will be distributed along **three** working weeks and sessions. The sessions will take place in small lecture rooms, wherein the lecturers will (i) introduce JavaScript to the students, (ii) go through a number of examples and (iii) solve general doubts or concerns.

Submission

The programming exercises will be submitted on the **8th week of the course** (see the timetable of lab sessions in Aula Global). The submission will consist of one exercise. The exercise to be submitted will be specified at the start of the submission session.

Assessment

In this set of exercises, we will evaluate: **functionality** (i.e. what you are asked to do), **code** (e.g. can a person who is not you understand the code? Have you written comments in the code?) and **user interface design** (e.g. think about a human user interacting with your page – colors, size of text…)

This set of exercises corresponds to 33% of the final mark of the labs of the course, i.e. 10% of the final mark. Students **will not pass** the course if they either copy the exercises from web pages or from another group, or allow them to copy their exercises.

Organization of the document

This document is divided into three sections. Section 2 gives an overview of Script languages and JavaScript. Section 3 describes the exercises and the submission procedure, along with the assessment criteria.

Script languages executed on the client side

Script languages executed on the client allow web developers to run functions and programs in web browsers, and this provides the end-user with a richer interaction with web applications, since these are more dynamic. Script languages are crucial in programming dynamic web applications, whose behavior changes depending on the needs of the end-user, execution conditions or the context of execution (e.g. web browsers). Examples of script languages are VBScript (Visual Basic) and JavaScript. In this document, we will focus on JavaScript.

JavaScript

JavaScript is an interpreted and object-oriented programming language, which is executed in web browsers. The web browser provides JavaScript with an execution context, with predefined objects representing elements of the browser and the page. With respect to user interfaces, JavaScript allows us to:

- Modify the text of an HTML document, as it is possible to insert text into a document (e.g. the value of a variable).
- React to events, as it allows web pages to execute JavaScript code depending on user- and browser-based actions, such as loading a web page or clicking on a button. Event-oriented programming is key to code the dynamic behavior of user interfaces.
- Read and modify HTML labels, since JavaScript enables web developers to add, modify or delete any HTML element. This presents us with an opportunity to modify the structure, content and presentation of a web page. We do this, in JavaScript code, through the DOM (Document Object Model) interface, which enables us to easily manipulate the tree of any HTML document.
- Validate data provided by the end-user. With JavaScript, we can check whether the data provided by the end-user is valid before sending or processing it (the typical example is an online form).

Web browsers might implement different versions of the DOM interface, and this fact often leads to compatibility issues of web pages running JavaScript code. Ensuring the compatibility of web pages in as many web browsers as possible is very important – not all the users go online with the same web browser. Hence, web developers need to be aware of the (un)supported JavaScript elements in different web browsers

The JavaScript tutorial of the w3c schools provides a number of interactive examples and an extensive reference to DOM objects [1]. For more advanced aspects, [2] provides tutorials, and the books [3] and [4], which are available at Safari Books Online, can be used as reference manuals. Another useful reference is [5].

JavaScript libraries

Both an increasing number of functionalities provided by web applications and the need to provide rich interactions in web pages have increased the complexity of developing web interfaces. To solve this problem, a number of JavaScript libraries have been created. These libraries provide us with pre-developed components, which can be used while developing web applications. Examples are Dojo Toolkit, Google Web Toolkit, UI Library and *jQuery*. We will focus on *jQuery*, which is used by companies like Google, Microsoft, IBM and Netflix.

jQuery

jQuery is an open source library designed to help web developers create web interfaces. The aim of *jQuery* can be summarized as: "write less, do more", i.e. to provide web developers with pre-developed components, which allow us to create complex functionalities in one line of code. Moreover, *JQuery* is compatible with nearly all the web browsers we are currently using to access to web pages.

jQuery allows us to change a webpage without re-loading it, thanks to manipulating the DOM object and AJAX events, effects and requests. We use $() o jQuery(). Thus, the syntax consists of a selector to select an HTML element followed by an action: **$(*selector*).*action*()**

Key characteristics of *jQuery* are:

- Interaction with HTML documents: selection and manipulation of DOM components and proprieties defined in CSS.
- Management of HTML events: the controllers of events are methods executed when there is a concrete type of interaction with an HTML document. We often talk about an action being triggered or fired by an event. For example, the instruction **$(document).ready(*function*)** will call the *function* when the document (HTML) is *ready*, i.e. the web page has been loaded.
- Animations: special actions which can be associated to HTML elements. Some examples are
    - **hide() / show()**
    - **slideDown() / slideUp() / slideToggle()**
    - **animate()**

For further information about *jQuery*, we recommend the tutorial provided by w3c schools, which provides a number of interactive examples and an extensive reference to actions, effects and animations [6]. The official page of *jQuery* provides lots of documentation and can be used as a reference guide [7]. Another reference is a book available at Safari Books online [8]. Another useful reference is the Mozilla MDN Web Docs [9].

Exercises

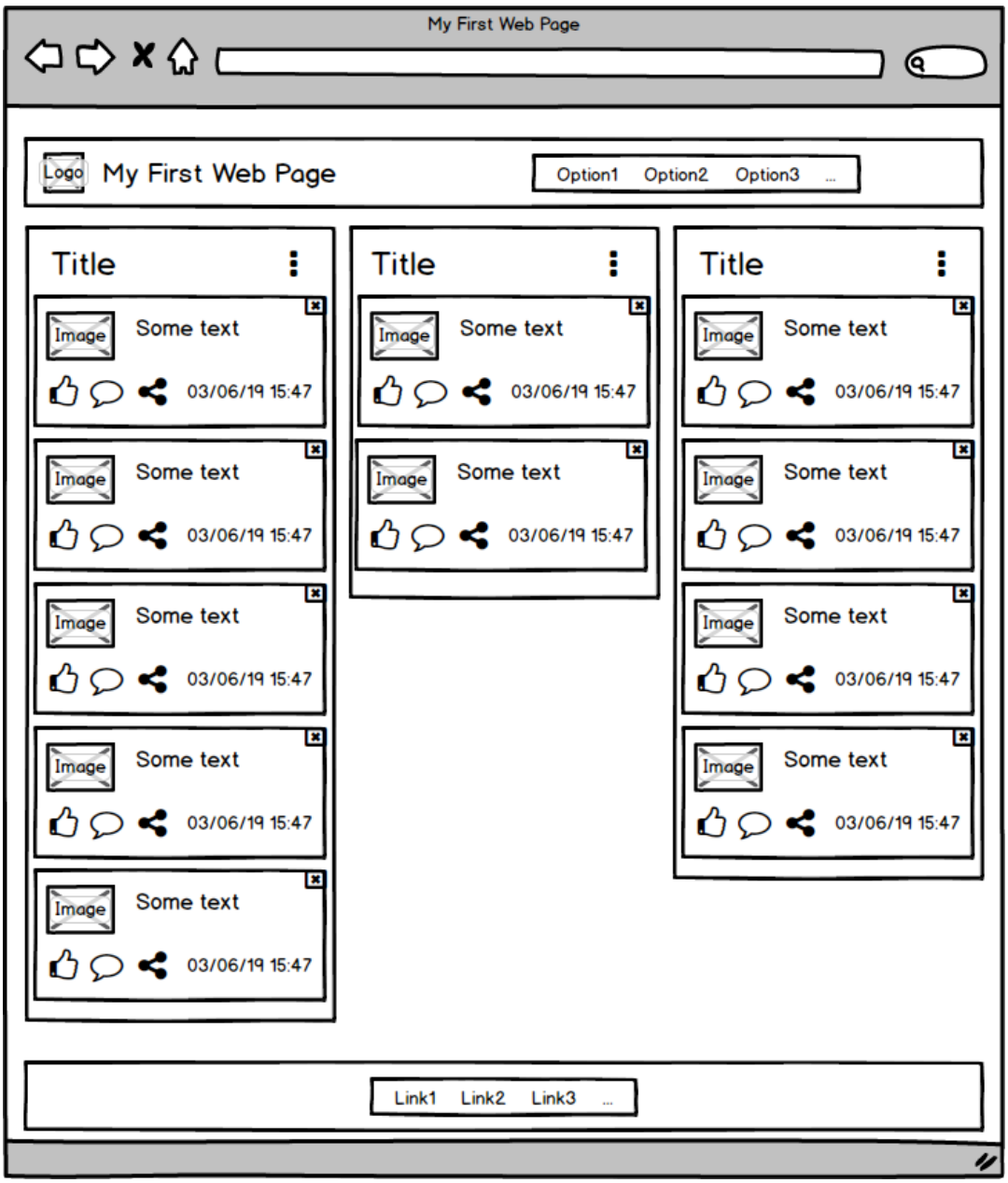
This section consists of 3 programming exercises, all of which are considered mandatory. For each exercise, we describe its main objective, suggest examples and provide supporting material.

Description of the exercises

Exercise 1

The aim of this exercise is to get familiar with the main elements of JavaScript for creating a dynamic page that shows activities and events grouped into categories.

The web page will follow the template shown in the image. IMPORTANT: It is mandatory to personalize the colors, the texts, and the icons shown in the page.



It will be divided into three sections: header with a menu, body and footer. These sections must be as follows:

- The header of the page will display a logo and/or a title, and a horizontal menu. The menu will include at least 3 options.
- The body section of the page will be divided into three vertical containers. Each container will show a group of activities or events, a title (i.e. the title has to be different for each one), an icon in the upper right corner (i.e. the hamburger button), and several horizontal boxes (i.e. cards). The boxes will represent the events or activities and they will be designed as follows:
    - The upper line will have an image, a text on the left and an icon (i.e. close button) on the right.
        - The text will have a minimum of three words. If its length is more than one line, the total height of the text has to be the same of the image.
        - The close button (in the upper right corner) will open a pop-up with a confirmation message before removing the box. Once the box has been removed, the other boxes in the same column have to be rearranged to fill the empty spaces (if any). The column can remain completely empty.
    - The lower line will have three icons, the date and the time. The three icons will be the actions "like", "comment", and "share". On mouse over the icons, a tooltip will show a short description of the associated action. By clicking on the icons, the actions will be implemented as follows:
        - By clicking on "like", the icon will change to highlight that the event or activity has received a preference.
        - By clicking on "share", a pop-up will be opened to ask the users where they would like to share the event or activity. The pop-up will contain the name of the event or the activity, and three icons representing three different sharing platforms, like Twitter, Facebook, and the email.
- By clicking on the hamburger button, a menu will be shown with 4 options. Among them, there will be an option to "archive this list". By selecting this option, a confirmation message will be opened before archiving the list of activities or events. Once the list has been archived, the corresponding column will be removed from the page and the remaining columns will be rearranged to fill any empty space.
- The footer of the page will display at least three links (e.g. the copyright and the social networks Facebook and Twitter) separated by a vertical line.

To complete this exercise, we encourage you to look at the examples available at http://www.w3schools.com/js/default.asp, paying special attention to the following examples:

JS Functions (http://www.w3schools.com/js/js_functions.asp)

JS Events (http://www.w3schools.com/js/js_events.asp)

JS HTML DOM (http://www.w3schools.com/js/js_htmldom.asp)

Exercise 2

The aim of this exercise is to get familiar with the main elements of JavaScript to design a form for the registration in a web page.

The webpage will be the same of the first exercise, except for the following changes:

- In the header, there will be an image as the profile image and text as the username. If the user doesn't have a profile image, a default one will be shown.
- One of the menu options will be "Log out". By clicking this option, the central body of the page will be emptied and the options "Register" and "Login" will appear in the menu.
- By clicking on "Register", the central body of the page will show a form with the following fields, some of them required and with a certain format that will depending of the specified information:
    - Account information
        - Username (Required)
        - Password (maximum 8 characters, where the allowed characters are letters [a-z] and digits [0-9]) (Required)
    - Personal information
        - Name and Surname (Required)
        - Email (will follow the format name@domain.extension) (Required)
        - Date of birth (dd / mm / yyyy) (Required)
        - Interests (a list of options, including for example computer science, videogames or cooking) (Optional)
        - Language (a list of options, including for example Spanish, English, or Italian) (Optional)
        - App purpose (for example, work) (Optional)
    - I have read and accept the Terms of Use (Required)
    - "Save" and "Delete" buttons
    - By clicking on the "save" button, a cookie will be stored with all the information contained in the form. If there is a cookie with the same email, the user will be notified that there is already an account associated with the specified email.
    - By clicking on the "delete" button, the initial information of the form will be restored.
    - The validation of the form can be done through: HTML5, JavaScript and jQuery.
- By clicking on "Login", the central body of the page will show a form asking for the email and password. The form will also have a "Login" button. When you

click on the button, a cookie with the entered data will be searched. If the cookie already exists, the page of exercise 1 of this same block will be loaded with the user personal information (username). If the cookie does not exist, the user will be notified that the specified email is not registered.

We encourage you to look at the following entries available at http://www.w3schools.com/js/default.asp to complete this exercise:

Forms Validation (http://www.w3schools.com/js/js_validation.asp)

HTML Forms and Input – About the form fields

HTML5 Input Types – About the new input fields of HTML5

DOM Css (http://www.w3schools.com/js/js_htmldom.asp)

JS Functions (http://www.w3schools.com/js/js_functions.asp)

JS Events (http://www.w3schools.com/js/js_events.asp)

JS HTML DOM (http://www.w3schools.com/js/js_htmldom.asp)

JS Cookie (http://www.w3schools.com/js/js_cookies.asp)

Exercise 3

The aim of this exercise is to get familiar with jQuery library and its benefits. With this aim, the drag&drop mechanism will be added to the Exercise 1.

The structure of the page must be the same of the Exercise 1, except for the following changes:

- Each column can be dragged and dropped to change its order. Once a column has been dragged and dropped, all the contained boxes will be also dragged and dropped.
- Inside each column, the boxes can be dragged and dropped to change their order. The boxes could be also dragged and dropped from a column to another.
- It is possible to use one of the available jQuery plugins.

We encourage you to look at the examples available at the official page of jQuery (http://jqueryui.com/).

Material

To complete these exercises, you are not allowed to use any HTML page editor, frameworks or specific tools, such as Dreamweaver.

Whereas no specific editor or tool is required to complete these exercises, we recommend that you use free tools such a Notepad++, HTML-Kit, Visual Studio Code, Sublime Text, Brackets or Atom. We also encourage you to use a JavaScript editor to debug your code, such as Firebug for Firefox. The lecturers will not help the students to use these tools.

Norms

The realization and submission of the programming exercises is guided by the following set of rules. If you do not comply with them, your mark **won't be more than 3** in the exercises.

Conducting the exercises

The exercises will be carried out in groups of two people.

The members of each group will belong to the same lab group.

The members of the group cannot be altered throughout the course.

The exercises will be carried out by using HTML5 and CSS3.

The exercises will be tested with either Mozilla Firefox version 16 (or above) or Chrome 26 (or above).

IMPORTANT: The resolution of problems of a particular nature will be made during office hours .The lecturers will not solve problems via e-mail.

Submitting the exercises

The exercises will be submitted **at the beginning of the session** indicated in the introduction of this document. Exercises submitted afterwards will not be considered.

The submission norms are:

- All the files will be submitted through Aula Global.
- All the files will be either zip or rar files, with the following filename:

Ep02_grXX.zip

- XX is the ID of your group. For example, group 5 will submit ep01 as:

Ep02_gr05.zip

The zip or rar files will have the following structure:

- ExN. Root folder. HTML files.
- ExN/style. CSS styles.
- ExN/images. Images and material.

N = number of exercise (1 – 3).

IMPORTANT: Class attendance is mandatory for at least one of the members of the practice group. Exercises submitted by email, during office hours, or outside the hours of the corresponding session will not be accepted.

References

[1] "JavaScript Tutorial", Tutorial JavaScript de W3 Schools, available at http://www.w3schools.com/js
[2] "JavaScript tutorials", available at http://www.javascriptkit.com/javatutors
[3] "The JavaScript PocketGuide", Lenny Burdette, Ed. PeachPit Press, 2010
[4] "JavaScript Step by Step", Steve Suehring, Ed. Microsoft Press, 2008
[5] "Eloquent JavaScript", available at http://eloquentjavascript.net
[6] "jQuery Tutorial", Tutorial jQuery de W3 Schools, available at http://www.w3schools.com/jquery/default.asp
[7] "jQuery Official WebSite", available at http://jquery.com
[8] "Learning jQuery", Jonathan Chaffer, Ed. Pckt Publishing, 2011
[9] "MDN Web Docs – Mozilla" available at https://developer.mozilla.org/es/