

Distributed Programming (03NQVOC)

Distributed Programming I (03MQPOV)

Laboratory exercise n.5

In order to write HTML or JavaScript files it is enough to use a normal text editor (like Notepad, gedit, nano, or vi). HTML files must be terminated with the suffix **.htm** or **.html**, while for JavaScript files it is suggested to use the **.js** file suffix. A specialized editor (like the one included in the Eclipse IDE for PHP developers) is suggested because it provides additional useful features (e.g. syntax highlighting and checking).

For the exercises involving user input, verify their proper functioning by introducing both valid and non-valid data. For instance:

- by pressing OK in the pop-ups without inserting data
- by closing input pop-ups by pressing the Cancel button
- by closing input pop-ups by pressing the upper right X button
- by introducing wrong data (e.g. in case of numeric data, valid data may be 5, +7, -3, 2.3, 5e3 while wrong data could be “5k”, “hello”, “5 thousand”, “3,2”).

Note: To simplify the debugging of pages containing javascript code, it is advised to work with Firefox and to install the **firebug** extension from the website <http://getfirebug.com>, then click on the “bug” icon to the lower right corner of the window to open the debug screen.

Exercise 5.1.1

Create an HTML page that shows a pop-up window with the “loading...” text when the page is loading, shows in the body of the page some brief information about the author of the page and then it shows another pop-up with the text: “page loaded”.

Exercise 5.1.2

Create an HTML page that requires, through a pop-up, to enter an integer number and then writes in the body of the page the inserted value along with a sentence stating if the number is even or odd.

Exercise 5.1.3

Create an HTML page that requires, through pop-ups, to enter two different numbers and then it outputs, through other two pop-ups, the sum and the difference of the two numbers.

Exercise 5.1.4

Create an HTML page that requires, through pop-ups, to enter two different numbers and then it outputs, in the body of the page, a list of the input values along with their sum and difference.

Exercise 5.1.5

Create an HTML page that requires to enter a number N through a pop-up and then it outputs in the body of the page, in form of a table, the values of square and cube powers of every number from 1 to N .

Exercise 5.1.6

Create an HTML page that requires to enter a number N through a pop-up and then it outputs in the body of the page, in form of a table, a pythagorean table of size N .

Exercise 5.1.7

Create an HTML page that requires to enter a series of numbers (one by one, each one entered by using a pop-up) and it outputs in the body of the page the list of the entered numbers along with their arithmetic mean. The input phase is terminated when the user enters the keyword "end".

Exercise 5.2.1

Create an HTML page that contains an input text field. Then, when the proper input button is pressed, a pop-up window must be shown with the input string, written in the text field, converted in lowercase.

Exercise 5.2.2

Create an HTML page with a multiple choice menu (with radio buttons), in which each choice is paired to a color (red, green, yellow, white). By selecting one of the possible choices, the background color of the page must change accordingly to the selected color.

Exercise 5.2.3

Create an HTML page with a form able to accept two positive integer numbers as input. When a button is pressed, the page must show in a text field (not editable) with the sum of these numbers.

Exercise 5.2.4

Modify the page developed in exercise 5.2.3 by removing the button and by making sure that the sum of the numbers is automatically updated when one of the two numbers changes.

Exercise 5.2.5

Create an HTML page with a form allowing to introduce a year number (e.g. 2011) that automatically calculates in which day of the week will be Christmas that year.

Exercise 5.2.6

Develop a Javascript function called “greatest” that accepts an unspecified amount of numbers as input parameters and outputs the greatest one. Store the function in a file called “functions.js”

Create then an HTML page that uses the file “functions.js” and uses the “greatest” function to calculate the maximum of five numbers entered as form fields. The greatest value must be shown in a pop-up window, activated when a specific button is pressed. The form must also include a “Cancel” button to clear the content of the five input fields.

Exercise 5.2.7

Create an HTML page that can be used to buy three different products (floppy disk, CD-R, DVD-R): the page must contain a form with three different drop-down lists that can be used to specify the quantities of products that can be bought. Each list must contain values ranging from 0 to 10.

When the user presses the proper button of the page it must calculate the total cost of the order. The prices of the three products are, respectively, 0.5, 1.0 and 2.0 Euros. The result must be shown in a non-editable text field, written in the same form.

The page must also contain a “Reset” button that allows, with one click, to set to zero all the values of the dropdown lists (and the total cost as well).