Tasks to Solve During Seminar 4

Internet Applications, ID1354

Write a document with your answers to the following tasks, and upload it to Canvas at the end of the seminar. The format of the document is not important. You solve the tasks together in the group, but each group member must write and upload their own document.

Mandatory Task

All members of the group shall, in turn, explain and motivate their program to the other group members. Write a brief summary of differences, and comment on advantages and disadvantages of the different solutions. Do not write about minor layout differences. Below are instructions on what to evaluate.

- Evaluate that your web sites meet the requirements specified in the lab tasks, and that all PHP and JavaScript code is well-structured and easy to understand. Also evaluate that your reports meet the requirements in the report template and in the lab tasks.
- Verify that no HTML code is sent in response to an AJAX request.
- Try to explain the main difference(s) between an application using Knockout (or another framework introducing a view model), and an application not using such a framework. You may skip this if no group member used a JavaScript framework.
- Now that an HTTP response does not contain any HTML, does it matter if the HTTP response, generated by a PHP `echo` statement, is sent by the same PHP file that received the HTTP request?

Optional Tasks

Solve as many of these tasks as time allows, and write your solutions in the same document as the mandatory task. If the solution is a piece of code, just paste the code in the document, without bothering about figures or formatting. The tasks may be solved in any order, you do not have to start from task one.

Task 1

Callback functions are used very frequently in JavaScript. Here are some exercises to give you a better understanding of callbacks.

a) Try to understand the JavaScript `map` function, which uses a callback to map each element in one array to another element in another array, see for example `http://www.w3schools.com/jsref/jsref_map.asp` Write a function that takes an array as parameter and uses `map` to add the index number to each element. It should change the array `["a", "b", "c"]` to `["0:a", "1:b","2:c"]`

b) Now write you own implementation of the `map` function itself. Try it with the program you wrote in bullet a) above.

c) Explain exactly the meaning of each line in listing 1 below. Also explain when each line is executed.

```
$(document).ready(function () {
   $("#someElem").click(function () {
      $("p").css({display: "none"});
   });
});
```

Listing 1: JavaScript code with callbacks.

Task 2

```
function Person(name) {
    this.name = name;
    this.handleName = function(callback) {
        callback(this);
    }
}

function nameHandler(obj) {
    alert("name: " + obj.name);
}

var person = new Person(’Stina’);
person.handleName(nameHandler);
```

Listing 2: Callback and objects

Listing 2 is an attempt to use objectoriented design best practices, by keeping objects intact.

a) What happens on line 4, and is there any reason to pass `this` instead of `name`? What does `this` contain?

b) Can you see any reason why the function `handleName` (lines 3-5) should *not* be replaced by the function `nameHandler` (lines 8-10)? This would make the code shorter and easier to understand, and the outcome of running the program would be exactly the same.