



Tasks to Solve During Seminar 1

Internet Applications, ID1354

Write a document with your answers to the following tasks, and upload it to Bilda at the end of the seminar. The format of the document is not important. Only one document per group is required, each group member uploads the same 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.

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.

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 your 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.



```
1 $(document).ready(function () {
2     $("#someElem").click(function () {
3         $("p").css({display: "none"});
4     });
5 });
```

Listing 1: JavaScript code with callbacks.

Task 2

```
1 function Person(name) {
2     this.name = name;
3     this.handleName = function(callback) {
4         callback(this);
5     }
6 }
7
8 function nameHandler(obj) {
9     alert("name: " + obj.name);
10 }
11
12 var person = new Person('Stina');
13 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.