

Assignment 5

Recommended readings:

- Lecture slides as starting literature
- MDN Links within slides for details
- <https://nodejs.org/en/>
- <https://www.typescriptlang.org/docs/home.html>

Note: All exercises must be solved with plain JavaScript or TypeScript and CSS not using additional libraries or frameworks (except the ones proposed).

Exercise 1 – Events

What is *event bubbling* and *event capturing*? How can you specify what method to use? How can you stop the propagation of events? What is the difference between `this`, `event.target` and `event.currentTarget`? What happens if parent elements have event handlers attached? What happens if parent and child elements have event handlers attached?

- Extend the example in `ex1/events.html` for demonstrating bubbling, capturing and `stopPropagation` with event handlers defined on the outer and additionally on the inner div. You may create multiple files for demonstrating the various options.
- Now the event handler should only do the console output if it is called with a second parameter `clicked(e,true)`. How can you pass parameters to event handlers in JS?
- The script in the file `ex1/events2.html` should display the first name and last name of the person as content of the corresponding div, when the div is clicked. Unfortunately, it does not work. Explain why this happens and provide two solutions:
 - Using the `function.bind` method [[MDN](#)].
 - Using an arrow function for calling the actual event handler [[MDN](#)].

Exercise 2 – ES 2015 - Modules

Based on your solution of assignment 4.3, create a module *multimedia* that exports the classes `Video` and `Image`. The class `Image` should be the default export.

Write a test script that imports the default as well as the `Video` export as `Movie` and creates some `Video` and `Image` objects.

Exercise 3 – Node.js

Write a server application using node.js and express with the following functionalities: When `/galleryJSON` is requested on your server you reply with a JSON file using the proper JSON header. The file should have the same structure as the one for assignment 4.4. However, you now need to read the file `gallery.csv` and output the data in the proper JSON output format. Create corresponding Java Script objects and serialize them to JSON for generating the JSON output. Be sure to set the proper content type header. Test your implementation with your existing JSON Image gallery from assignment 4.4. For doing this you will need to host your previous HTML and JS code on the node.js / express server. This test is only mandatory, if you solved assignment 4.4.

Exercise 4 – TypeScript 1

First install TypeScript in node.js and then compile the file `ex4/person.ts` to JavaScript. Execute the JavaScript in node.js. Then compare `person.ts` and `person.js` and explain the difference.

Exercise 5 – TypeScript 2

Start with the file `ex4/person.ts` and modify it such that

- There is a person class implementing the `personI` interface and Professors extend the person class.
- Persons additionally have a birthdate and a method `getAge()` that returns the current age as a number.
- Professors can have a position as a “full professor”, “adjunct professor” or “associate professor”. Use an enumeration type to represent this and modify the `sayHello()` method such that the position is included in the greeting.
- Create an additional class `StudyPlan` to represent study plans. Study plans have a number, a name and an expiration date.
- Students are subclasses of persons. Students have the additional properties `studentNumber` and `studyPlans`. The property `studyPlans` is an array of `StudyPlan` instances. Students also have the additional methods: `enrol(StudyPlan)`, `cancel(StudyPlan)`, and `showStudyPlans()`. The method `showStudyPlans()` returns a list of all enrolled `studyPlan` names as a string. Be sure that one study plan can only be enrolled once.

Create some persons, study plans, students and professors to test your implementation. **Be sure to use proper TypeScript type annotations, suitable visibility modifiers and implement getter and setter methods for all properties.**