SWA1Work Sheet:
Asynchronous Programming

Topic

In this module we will work with the various methods of asynchronous programming: Callbacks, promises, and async/await. To have something asynchronous to work with, we will look at calling a server from a client using either XmlHttpRequest or fetch. In order to avoid some of the pitfalls with the asynchronous programming methods, we will have to look deeper into the JavaScript runtime model.

Functional programming solves many of the problems inherent in asynchronous programming, so we will look at functional programming in JavaScript first.

Period

Week 39 – 41

Literature

- [Haverbeke] 5, 11, 18
- Eric Elliot: Curry and Function Composition
- Video: Anjana Vakil: Learning Functional Programming with JavaScript - JSUnconf 2016
- Concurrency model and Event Loop
- Video: What the heck is the event loop anyway?

Goals

- define functional programming
- use functional programming in JavaScript
- relate functional programming to imperative programming in JavaScript
- identify pitfalls of asynchronous programming
- describe the methods for client/server programming
- apply asynchronous programming using Promises to implement client/server programming
- apply asynchronous programming using XmlHttpRequest to implement client/server programming
- Exam question 3 – 5

Course Assignment

Course assignment 2: Asynchronous programming in JavaScript.

Solve the course assignment in groups of 2 or 3. Hand in by Sunday, 19 October before midnight.