# JavaScript
# Lab 1

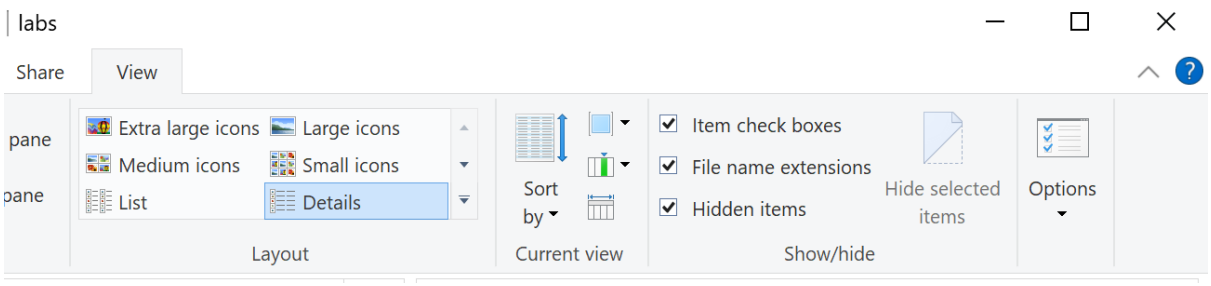


## 1 Lab Setup

You should have received installation instructions on setting up Node as well as a suitable IDE (VS Code / Notepad++).

You should also have received the source code for the lab sessions, which should have been downloaded from the course LMS or a GitHub repo. If you are using VS Code, you can open up the folder containing the JavaScript files for the lab, which should be `labcode/js`.

If you are using Windows, make sure that you have checked the File name extensions in your File Explorer in order for us to be able to directly manipulate the file extensions when creating or modifying files.



# 2 Executing a JavaScript program

There are a multitude of ways to execute a JavaScript program.

1. Using the Node.js runtime environment to directly execute a JavaScript program
2. Including JavaScript into a HTML document
3. Running JavaScript in console tab of Dev Tools
4. Using an online JavaScript playground (https://www.better.dev/javascript-playgrounds-to-use)
5. Using a framework that is based on JavaScript (e.g. Angular, React, Vue)

We will use approach 1 for now and examine the others later on.

To run Node.js, we need a shell terminal. We can use the one directly embedded in the IDE (VS Code) or we can just use the standard command prompt.

Navigate to the `labcode/js` folder in the shell terminal, and open up the file `hello-world.js` in the IDE (or you can simply use Notepad++).

Run the program by simply typing:

`node` *`name-of-file`*

You can choose to include the extension *.js as well if you wish to explicit. Node will automatically assume that extension if none is included.

Files to use: `hello-world.js`

# 3 Basic Syntax

https://www.javascripttutorial.net/javascript-syntax/

# 4 Working with variables

https://www.javascripttutorial.net/javascript-variables/

Files to use: `variables-basic.js`

# 5 Data types

https://developer.mozilla.org/en-US/docs/Glossary/Primitive

https://www.javascripttutorial.net/javascript-data-types/

https://codeburst.io/javascript-whats-the-difference-between-null-undefined-37793b5bfce6

Files to use: `datatypes-basic.js`

# 6 Function Basics

https://www.javascripttutorial.net/javascript-function/

Files to use: `functions-basic.js`

# 7 Object Basics

https://www.javascripttutorial.net/javascript-objects/

https://www.javascripttutorial.net/javascript-object-methods/

Files to use:
`objects-basic.js`
`object-methods-basic.js`

## 7.1 Primitives vs references

https://www.javascripttutorial.net/javascript-primitive-vs-reference-values/

https://www.javascripttutorial.net/javascript-pass-by-value/

Files to use:
`primitive-reference.js`

# 8 string, template literals and String object

https://www.javascripttutorial.net/javascript-string/

String comparison is based on the ASCII code for the respective characters
https://theasciicode.com.ar/

When you attempt to call a method or access a property on a string variable initialized with a primitive string value, automatic conversion to a String object occurs. This is known as autoboxing

https://dev.to/benjaminmock/do-you-know-what-autoboxing-in-js-is-enl

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String

Commonly used String methods

https://www.w3schools.com/js/js_string_methods.asp
https://www.w3schools.com/js/js_string_search.asp

Template literals

https://www.javascripttutorial.net/es6/javascript-template-literals/

Files to use:
`string-methods.js`
`template-literals.js`

# 9 Arrays

https://www.javascripttutorial.net/javascript-array/

https://www.javascripttutorial.net/javascript-array-methods/

https://www.javascripttutorial.net/javascript-array-splice/

Files to use:
`arrays-basic.js`