COURSEWARE

Professional Skills Agile Fundamentals Jira Git Databases Introduction Java Beginner Maven Testing (Foundation) Java Intermediate HTML CSS Javascript What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Colosures Callbacks and Promises Cookies Hoisting Prototypes Query Parameters			
Git Databases Introduction Java Beginner Maven Testing (Foundation) Java Intermediate HTML CSS Javascript What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	Pro	ofessional Skills	
Git Databases Introduction Java Beginner Maven Testing (Foundation) Java Intermediate HTML CSS Javascript What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	Agile Fundamentals		
Databases Introduction Java Beginner Maven Testing (Foundation) Java Intermediate HTML CSS Javascript What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	Jira		
Java Beginner Maven Testing (Foundation) Java Intermediate HTML CSS Javascript What is JavaScript Cetting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	Git		
Maven Testing (Foundation) Java Intermediate HTML CSS Javascript What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	Databases Introduction		
Testing (Foundation) Java Intermediate HTML CSS Javascript What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	Java Beginner		
Testing (Foundation) Java Intermediate HTML CSS Javascript What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	Maven		
Java Intermediate HTML CSS Javascript What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Callbacks and Promises Cookies Hoisting Prototypes			
HTML CSS Javascript What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	resuring (Foundation)		
CSS Javascript What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	Java Intermediate		
Javascript What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	HTML		
 What is JavaScript Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	CSS		
 Getting started with JS Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	Javascript		
 Variables Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	0	What is JavaScript	
 Data types ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	0	Getting started with JS	
 ASI Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	0	Variables	
 Strict mode Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	0	Data types	
 Iteration Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	0	ASI	
 Conditionals with Truthy / Falsey Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	0	Strict mode	
 Objects, Arrays + JSON Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	0	Iteration	
 Structuring JS Code Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	0	Conditionals with Truthy / Falsey	
 Destructuring Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	0	Objects, Arrays + JSON	
 Scope Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	0	Structuring JS Code	
Functions, function expressions and arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	0	Destructuring	
arrow functions The ECMAScript 6 Specification OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes	0	Scope	
 OOP in JavaScript Best Practices Closures Callbacks and Promises Cookies Hoisting Prototypes 	0	•	
Best PracticesClosuresCallbacks and PromisesCookiesHoistingPrototypes	0	The ECMAScript 6 Specification	
ClosuresCallbacks and PromisesCookiesHoistingPrototypes	0	OOP in JavaScript	
Callbacks and PromisesCookiesHoistingPrototypes	0	Best Practices	
CookiesHoistingPrototypes	0	Closures	
HoistingPrototypes	0	Callbacks and Promises	
Prototypes	0	Cookies	
	0	Hoisting	
Query Parameters	0	Prototypes	
	0	Query Parameters	

Higher Order Functions

Structuring JS Code

Contents

- Overview
- <u>Tutorial</u>
 - Structure within a script
 - Modular Code
- Exercises

Overview

Stucturing one's code in a reasonable manner saves time, headaches and processing power in every language - JavaScript is no exception to the rule.

Tutorial

Structure within a script

With any new script that is created, ensure that it is kept out of the global scope - to do this, wrap it in a function.

If it is something that we can explicitly run later or at a specific time, then wrap it in a *named function* that can be called later:

```
let myScript = function () {
    // insert fantastic code
}

// Later on...
myScript();
```

Else, wrap it in an <u>IIFE (Immediately Invoked Function Expression)</u>.

```
(function() {
    // insert fantastic code
})();
```

Within scripts, it is important to break up code into three sections:

- 1. Variables
- 2. Methods
- 3. Initialisations and event listeners

This helps keep code predictable and organised.

_	-		
O V	Veb Storage		
0 [OOM Manipulation		
O F	landling Events and Timed Events		
O A	synchronous Programming		
O F	HTTP-Requests		
	MLHttpRequests		
O F	etch API		
Spring Boot			
Selenium			
Sonarqube			
Advanced Testing (Theory)			
Cucumber			
MongoDB			
Express			
NodeJS			
React			
Express-Testing			
Networking			
Security			
Cloud Fundamentals			
AWS Foundations			
AWS Intermediate			
Linux			
DevOps			
Jenkins Introduction			
Jenki	Jenkins Pipeline		
Markdown			

IDE Cheatsheet

```
(function(){
    /*
    *
    * Variables
    *
    */
    let element = document.querySelector('#app');

    /*
    *
    * Methods
    *
    */
    let handleClicks = function(event){
        console.log(event.target);
    }

    /*
    *
    * Inits & Event Listeners
    *
    */
    element.addEventListener('click',handleClicks);
})();
```

Modular Code

Do not keep all of your code in one giant JavaScript file. That would be madness!

Ensure you create a /js directory in your project where you keep all the individual script files.

There are module bundlers such as <u>gulp</u> which you can use to combine fines (called concatenation) and minify them.

Exercises

Revisit any previous JS work, and ensure it follows the correct structure.