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 0 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 **Query Parameters**

Higher Order Functions

What is JavaScript

Contents

- Overview
- <u>Popularity</u>
- Advantages
- <u>Limitations</u>
- <u>Tutorial</u>
- Exercises

Overview

JavaScript, often referred to as JS, is a scripting language that conforms to the ECMAScript specification.

JavaScript is a text-based language, used both on the client-side and server-side, that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

Common examples of JavaScript that you might use every day include: the search box on Amazon, a news recap video embedded on The New York Times, or refreshing your Twitter feed.

Incorporating JavaScript improves the user experience of the web page by converting it from a static page into an interactive one.

JavaScript adds interactivity to web pages.

Popularity

Over the last few years JavaScript has grown massively in popularity, with many different frameworks that have surfaced. The image below depicts how many frameworks there were in 2009 compared to 2019:



This popularity stems from the limitless possibilities JavaScript allows when working with web pages. For example:

- Showing or hiding more information with the click of a button
- Changing the colour of a button when the mouse hovers over it
- Sliding through a carousel of images on the homepage
- Zooming in or zooming out on an image
- Displaying animations

_	-
0	Web Storage
0	DOM Manipulation
0	Handling Events and Timed Events
0	Asynchronous Programming
0	HTTP-Requests
0	XMLHttpRequests
0	Fetch 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	
Jenkins Pipeline	
Markdown	

IDE Cheatsheet

• Using a drop-down hamburger menu

Advantages

- **Speed** As JavaScript is compiled on the client browser it can be run straight away.
- Simplicity JavaScript is relatively simple to learn and implement.
- Versatility JavaScript has had many new frameworks that can allow a
 website to have a fully functioning backend in JS, such as Node.js and
 Express.js, as well as a frontend implemented using a JS Framework, such
 as Vue.js and React.js.

Limitations

- JavaScript being disabled If JavaScript is disabled then this would prevent JavaScript functionality taking place on the client side. Some browsers by default have JavaScript disabled *e.g. Tor Browser*.
- Client-Side Security Code executes on the user's computer, which could then be exploited for malicious purposes (this is another reason why JavaScript may be disabled).
- Browser Support JavaScript is sometimes interpreted differently by different browsers. Whereas server-side scripts will always produce the same output, client-side scripts can be a little unpredictable. As a result, testing is key and needs to be checked across all major browsers; however, there are services available that can automate this process.

Tutorial

There are no tutorials for this module.

Exercises

There are no exercises for this module.