

# Front-end

TechGuide - Alura, FIAP e PM3

---

## Front-end

### Level 1

#### HTML - Fundamentals:

HTML is a markup language that defines the structure of your content. HTML consists of a series of elements, which you use to make it appear a certain way, or act a certain way. The enclosing tags can make a word or image hyperlink to somewhere else, can italicize words, can make the font bigger or smaller, and so on.

Learning which tags are required for basic HTML

Creating a text paragraph

Displaying an image

Knowing the difference between 'h1', 'h2', 'h3', etc.

Creating a hyperlinked text

Creating a form with relevant fields

Creating an ordered or unordered list of items

Creating a list of items within a dropdown list

Linking to a CSS file

Creating a table

Adding IDs and classes

#### Contents

WebSiteW3Schools: HTML Introduction ([https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp))

WebSiteW3Schools: HTML Elements ([https://www.w3schools.com/html/html\\_elements.asp](https://www.w3schools.com/html/html_elements.asp))

WebSiteW3Schools: HTML Formatting Elements ([https://www.w3schools.com/html/html\\_formatting.asp](https://www.w3schools.com/html/html_formatting.asp))

WebSiteW3Schools: HTML Block and Inline Elements ([https://www.w3schools.com/html/html\\_blocks.asp](https://www.w3schools.com/html/html_blocks.asp))

WebSiteW3Schools: HTML Layout Elements ([https://www.w3schools.com/html/html\\_layout.asp](https://www.w3schools.com/html/html_layout.asp))

WebSiteW3Schools: HTML Lists ([https://www.w3schools.com/html/html\\_lists.asp](https://www.w3schools.com/html/html_lists.asp))

WebSiteW3Schools: HTML Tables ([https://www.w3schools.com/html/html\\_tables.asp](https://www.w3schools.com/html/html_tables.asp))

WebSiteW3Schools: HTML Forms ([https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp))

WebSiteMDN Web Docs: HTML basics ([https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics))

ArticleHTML vs CSS: What's the Difference? (<https://learn.onemonth.com/html-vs-css/>)

ArticleGetting Started with HTML (<https://towardsdatascience.com/getting-started-with-html-98866c3e5c9e>)

YouTube	Programming with Mosh: HTML Tutorial for Beginners - <a href="https://www.youtube.com/watch?v=qz0aGYrrlhU">https://www.youtube.com/watch?v=qz0aGYrrlhU</a>	)
YouTube	HTML Crash Course ( <a href="https://www.youtube.com/watch?v=kUMe1FH4CHE">https://www.youtube.com/watch?v=kUMe1FH4CHE</a>	)
YouTube	freeCodeCamp.org: Learn HTML – Full Tutorial for Beginners ( <a href="https://www.youtube.com/watch?v=G3e-cpL7ofc">https://www.youtube.com/watch?v=G3e-cpL7ofc</a>	)
YouTube	SuperSimpleDev: HTML & CSS Full Course - Beginner to Pro ( <a href="https://www.youtube.com/watch?v=G3e-cpL7ofc">https://www.youtube.com/watch?v=G3e-cpL7ofc</a>	)

## CSS - Fundamentals:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. CSS can be used for very basic document text styling — for example, for changing the color and size of headings and links. It can be used to create a layout — for example, turning a single column of text into a layout with a main content area and a sidebar for related information. It can even be used for effects such as animation.

Learning the visual structure of a page, with 'margin' and 'padding'

Establishing the size with 'width' and 'height'

Learning about the position of an element ('static', 'relative' or 'absolute')

Learning about the display of an element ('block', 'inline', 'inline-block')

Learning how to position images in relation to text

Learning about alignment

Learning about font style

Learning the differences and advantages of using the different units of measurement in CSS (% , relative, etc)

Connecting to the elements (IDs, classes) of an HTML file

Changing the characteristics of an element when the mouse hovers over it

Learning box-sizing

Learning Flexbox

Learning Grid

## Contents

WebSiteW3Schools: CSS Tutorial (<https://www.w3schools.com/css/>)

WebSiteW3Schools: CSS Introduction ([https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp))

WebSiteW3Schools: CSS Syntax ([https://www.w3schools.com/css/css\\_syntax.asp](https://www.w3schools.com/css/css_syntax.asp))

WebSiteW3Schools: CSS Selectors ([https://www.w3schools.com/css/css\\_selectors.asp](https://www.w3schools.com/css/css_selectors.asp))

WebSiteW3Schools: CSS Box Model ([https://www.w3schools.com/css/css\\_boxmodel.asp](https://www.w3schools.com/css/css_boxmodel.asp))

WebSiteW3Schools: CSS Units ([https://www.w3schools.com/css/css\\_units.asp](https://www.w3schools.com/css/css_units.asp))

WebSite	MDN Web Docs: Getting started with CSS ( <a href="https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/Getting_started">https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/Getting_started</a>	)
---------	--	---

WebSite	MDN Web Docs: CSS selectors ( <a href="https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors">https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors</a>	)
---------	---	---

WebSite	MDN Web Docs: Cascade and inheritance ( <a href="https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance">https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance</a>	)
---------	---	---

WebSite	MDN Web Docs: display property ( <a href="https://developer.mozilla.org/en-US/docs/Web/CSS/display">https://developer.mozilla.org/en-US/docs/Web/CSS/display</a>	)
---------	--	---

WebSite	MDN Web Docs: Basic concepts of <a href="https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox">https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox</a>	)
---------	---	---

	Flexbox (	
	MDN Web Docs: Basic	<a href="https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout">https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout</a>
WebSite	concepts of Grid Layout (	)
Article	CSS Tutorial: Learn the basics of CSS in 5 minutes (	<a href="https://medium.com/free-code-camp/get-started-with-css-in-5-minutes-e0804813fc3e">https://medium.com/free-code-camp/get-started-with-css-in-5-minutes-e0804813fc3e</a> )
Article	The Complete CSS Flex Tutorial (	<a href="https://jst.hashnode.dev/complete-css-flex-tutorial">https://jst.hashnode.dev/complete-css-flex-tutorial</a> )
Article	The Complete CSS Grid Tutorial (	<a href="https://jst.hashnode.dev/css-grid-tutorial">https://jst.hashnode.dev/css-grid-tutorial</a> )
YouTube	Web Dev Simplified: Learn CSS in 20 Minutes (	<a href="https://www.youtube.com/watch?v=1PnVor36_40">https://www.youtube.com/watch?v=1PnVor36_40</a> )
YouTube	freeCodeCamp.org: CSS Tutorial - Zero to Hero (Complete Course) (	<a href="https://www.youtube.com/watch?v=1Rs2ND1ryYc">https://www.youtube.com/watch?v=1Rs2ND1ryYc</a> )
YouTube	SuperSimpleDev: HTML & CSS Full Course - Beginner to Pro (	<a href="https://www.youtube.com/watch?v=G3e-cpL7ofc">https://www.youtube.com/watch?v=G3e-cpL7ofc</a> )

## JavaScript - Fundamentals:

JavaScript is the world's most popular programming language and is one of the core technologies of the World Wide Web, alongside HTML and CSS. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles.

Knowing the primitive types

Declaring variables, considering the difference between 'var', 'let' and 'const'

Using conditional structures ('if', 'else')

Know the assignment and comparison operators ('=', '==', '===')

Using repetition structures and loops ('while', 'for')

Using functions, passing parameters and arguments

Manipulating arrays and lists

Getting data from an API

Making asynchronous calls using 'Async/Await', 'Promise', etc

## Contents

WebSite	W3Schools: JavaScript Syntax (	<a href="https://www.w3schools.com/js/js_syntax.asp">https://www.w3schools.com/js/js_syntax.asp</a> )
WebSite	W3Schools: JavaScript Variables (	<a href="https://www.w3schools.com/js/js_variables.asp">https://www.w3schools.com/js/js_variables.asp</a> )
WebSite	W3Schools: JavaScript Operators (	<a href="https://www.w3schools.com/js/js_operators.asp">https://www.w3schools.com/js/js_operators.asp</a> )
WebSite	W3Schools: JavaScript Comparison and Logical Operators (	<a href="https://www.w3schools.com/js/js_comparisons.asp">https://www.w3schools.com/js/js_comparisons.asp</a> )
WebSite	W3Schools: JavaScript Data Types (	<a href="https://www.w3schools.com/js/js_datatypes.asp">https://www.w3schools.com/js/js_datatypes.asp</a> )
WebSite	W3Schools: JavaScript Arrays (	<a href="https://www.w3schools.com/js/js_arrays.asp">https://www.w3schools.com/js/js_arrays.asp</a> )
WebSite	W3Schools: JavaScript if, else, and else if (	<a href="https://www.w3schools.com/js/js_if_else.asp">https://www.w3schools.com/js/js_if_else.asp</a> )
WebSite	W3Schools: JavaScript Iterables (while, for) (	<a href="https://www.w3schools.com/js/js_iterables.asp">https://www.w3schools.com/js/js_iterables.asp</a> )
WebSite	W3Schools: JavaScript Functions (	<a href="https://www.w3schools.com/js/js_functions.asp">https://www.w3schools.com/js/js_functions.asp</a> )
WebSite	W3Schools: JavaScript Objects (	<a href="https://www.w3schools.com/js/js_objects.asp">https://www.w3schools.com/js/js_objects.asp</a> )
WebSite	W3Schools: JavaScript Classes (	<a href="https://www.w3schools.com/js/js_classes.asp">https://www.w3schools.com/js/js_classes.asp</a> )
WebSite	MDN Web Docs: A first splash into JavaScript (	<a href="https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/A_first_splash">https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/A_first_splash</a> )

WebSite	MDN Web Docs: Handling text — strings in JavaScript (	<a href="https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Strings">https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Strings</a>	)
WebSite	MDN Web Docs: if...else (	<a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else</a>	)
WebSite	MDN Web Docs: while (	<a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/while">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/while</a>	)
WebSite	MDN Web Docs: for (	<a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for</a>	)
WebSite	MDN Web Docs: Arrays (	<a href="https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Arrays">https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Arrays</a>	)
WebSite	MDN Web Docs: Template strings (	<a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals</a>	)
Article	An introduction to JavaScript's async and await (	<a href="https://medium.com/@zellwk/an-introduction-to-javascripts-async-and-await-edb313356677">https://medium.com/@zellwk/an-introduction-to-javascripts-async-and-await-edb313356677</a>	)
Article	Javascript: Map, Filter, and Reduce (	<a href="https://harsh-patel.medium.com/javascript-map-filter-and-reduce-51da071d6dc4">https://harsh-patel.medium.com/javascript-map-filter-and-reduce-51da071d6dc4</a>	)
YouTube	Aaron Jack: Learn JavaScript in just 5 minutes (	<a href="https://www.youtube.com/watch?v=c-I5S_zTwAc">https://www.youtube.com/watch?v=c-I5S_zTwAc</a>	)
YouTube	Topknot Clare: JavaScript Basics in 10 Minutes (	<a href="https://www.youtube.com/watch?v=c-I5S_zTwAc">https://www.youtube.com/watch?v=c-I5S_zTwAc</a>	)
YouTube	Programming with Mosh: JavaScript Tutorial for Beginners - Learn JavaScript in 1 Hour (	<a href="https://www.youtube.com/watch?v=W6NZfCO5SIk">https://www.youtube.com/watch?v=W6NZfCO5SIk</a>	)
YouTube	freeCodeCamp.org: Introduction to JavaScript (	<a href="https://www.youtube.com/watch?v=y9oxzTGERs">https://www.youtube.com/watch?v=y9oxzTGERs</a>	)
YouTube	freeCodeCamp.org: Learn JavaScript - Full Course for Beginners (	<a href="https://www.youtube.com/watch?v=PkZNo7MFNEg">https://www.youtube.com/watch?v=PkZNo7MFNEg</a>	)

## DOM - Fundamentals:

The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects; that way, programming languages can interact with the page.

Understanding how the DOM tree works

Accessing and manipulating HTML and CSS elements

Accessing the parents and children of an element

Inserting a new element to the tree

Removing an element from the tree

Waiting for an event on a certain page element using addEventListener()

## Contents

WebSite	W3Schools: JavaScript HTML DOM Methods (	<a href="https://www.w3schools.com/js/js_htmldom_methods.asp">https://www.w3schools.com/js/js_htmldom_methods.asp</a>	)
WebSite	W3Schools: JavaScript HTML DOM Elements (	<a href="https://www.w3schools.com/js/js_htmldom_elements.asp">https://www.w3schools.com/js/js_htmldom_elements.asp</a>	)
WebSite	W3Schools: JavaScript HTML DOM - Changing HTML (	<a href="https://www.w3schools.com/js/js_htmldom_html.asp">https://www.w3schools.com/js/js_htmldom_html.asp</a>	)

WebSite	W3Schools: JavaScript HTML DOM - Changing CSS (	<a href="https://www.w3schools.com/js/js_htmldom_css.asp">https://www.w3schools.com/js/js_htmldom_css.asp</a> )
WebSite	W3Schools: JavaScript HTML DOM Events (	<a href="https://www.w3schools.com/js/js_htmldom_events.asp">https://www.w3schools.com/js/js_htmldom_events.asp</a> )
WebSite	W3Schools: JavaScript HTML DOM EventListener (	<a href="https://www.w3schools.com/js/js_htmldom_eventlistener.asp">https://www.w3schools.com/js/js_htmldom_eventlistener.asp</a> )
WebSite	W3Schools: JavaScript HTML DOM Elements (Nodes) (	<a href="https://www.w3schools.com/js/js_htmldom_nodes.asp">https://www.w3schools.com/js/js_htmldom_nodes.asp</a> )
WebSite	MDN Web Docs: Introdução ao DOM (	<a href="https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction">https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction</a> )
WebSite	MDN Web Docs: querySelectorAll() (	<a href="https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelectorAll">https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelectorAll</a> )
WebSite	MDN Web Docs: addEventListener() (	<a href="https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener">https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener</a> )
Article	An introduction to the JavaScript DOM (	<a href="https://medium.com/free-code-camp/an-introduction-to-the-javascript-dom-512463dd62ec">https://medium.com/free-code-camp/an-introduction-to-the-javascript-dom-512463dd62ec</a> )
Article	The DOM of Javascript (	<a href="https://javascript.plainenglish.io/the-dom-of-javascript-848506ebf386">https://javascript.plainenglish.io/the-dom-of-javascript-848506ebf386</a> )
YouTube	Zac Gordon: An Introduction to the DOM (Document Object Model) in JavaScript (	<a href="https://www.youtube.com/watch?v=I-0nPnSvbX8">https://www.youtube.com/watch?v=I-0nPnSvbX8</a> )
YouTube	Zac Gordon: An Introduction to DOM Events with JavaScript (	<a href="https://www.youtube.com/watch?v=QE1YQnhntgw">https://www.youtube.com/watch?v=QE1YQnhntgw</a> )
YouTube	Web Dev Simplified: Learn DOM Manipulation In 18 Minutes (	<a href="https://www.youtube.com/watch?v=y17RuWkWdn8">https://www.youtube.com/watch?v=y17RuWkWdn8</a> )

## Javascript - Accessibility:

Web accessibility is the inclusive practice of ensuring there are no barriers that prevent interaction with, or access to, websites by people with physical disabilities, situational disabilities, and socio-economic restrictions on bandwidth and speed.

Writing code with accessibility in mind

### Contents

WebSite	W3: Accessibility ( <a href="https://www.w3.org/standards/webdesign/accessibility">https://www.w3.org/standards/webdesign/accessibility</a> )	
WebSite	MDN Web Docs: CSS and JavaScript accessibility best practices (	<a href="https://developer.mozilla.org/en-US/docs/Learn/Accessibility/CSS_and_JavaScript">https://developer.mozilla.org/en-US/docs/Learn/Accessibility/CSS_and_JavaScript</a> )
WebSite	WebAIM: Accessible JavaScript ( <a href="https://webaim.org/techniques/javascript/">https://webaim.org/techniques/javascript/</a> )	
Article	Writing JavaScript with accessibility in mind (	<a href="https://medium.com/@matuzo/writing-javascript-with-accessibility-in-mind-a1f6a5f467b9">https://medium.com/@matuzo/writing-javascript-with-accessibility-in-mind-a1f6a5f467b9</a> )
YouTube	Flux Academy: Accessible Web Design - What Is It & How To Do It (	<a href="https://www.youtube.com/watch?v=-ao_Kc_8rpE">https://www.youtube.com/watch?v=-ao_Kc_8rpE</a> )
YouTube	SFHTML5: Accessibility Fundamentals with Rob Dodson (	<a href="https://www.youtube.com/watch?v=z8xUCzTof8">https://www.youtube.com/watch?v=z8xUCzTof8</a> )
YouTube	Fullstack Academy: Web Accessibility Guidelines - Accessibility tips for JavaScript Developers (	<a href="https://www.youtube.com/watch?v=z8xUCzTof8">https://www.youtube.com/watch?v=z8xUCzTof8</a> )
YouTube	freeCodeCamp.org: Web App Tutorial - JavaScript, Mobile First, Accessibility, Persistent Data, Sass (	<a href="https://www.youtube.com/watch?v=z8xUCzTof8">https://www.youtube.com/watch?v=z8xUCzTof8</a> )
Course	Web.dev: Learn Accessibility! ( <a href="https://web.dev/learn/accessibility/welcome">https://web.dev/learn/accessibility/welcome</a> )	

## SEO Strategies:

Search engine optimization (SEO) is the process of improving the quality and quantity of website traffic to a website or a web page from search engines.

Choosing keywords

Understanding how Google ranks pages

Knowing the ranking factors

Performing Link Building

### Contents

Google Search Central: WebSite	Search Engine Optimization (SEO) Starter Guide (	<a href="https://developers.google.com/search/docs/fundamentals/seo-starter-guide">https://developers.google.com/search/docs/fundamentals/seo-starter-guide</a>	)
Article	Moz: The Beginner's Guide to SEO	( <a href="https://moz.com/beginners-guide-to-seo">https://moz.com/beginners-guide-to-seo</a> )	
Article	Neil Patel: What is SEO? Your Complete Step-By-Step Guide (	<a href="https://neilpatel.com/what-is-seo/">https://neilpatel.com/what-is-seo/</a>	)
YouTube	Simplilearn: SEO In 5 Minutes   What Is SEO And How Does It Work (	<a href="https://www.youtube.com/watch?v=MYE6T_gd7H0">https://www.youtube.com/watch?v=MYE6T_gd7H0</a>	)
YouTube	Fireship: SEO for Developers in 100 Seconds (	<a href="https://www.youtube.com/watch?v=B58GgsehKQ">https://www.youtube.com/watch?v=B58GgsehKQ</a>	)
YouTube	Ahrefs: Complete SEO Course for Beginners - Learn to Rank #1 in Google (	<a href="https://www.youtube.com/watch?v=xsVTqzratPs">https://www.youtube.com/watch?v=xsVTqzratPs</a>	)

## Responsive Design:

Responsive web design (RWD) or responsive design is an approach to web design that aims to make web pages render well on a variety of devices and window or screen sizes from minimum to maximum display size to ensure usability and satisfaction.

Adjusting your pages to the user's screen size

Learning about Media queries

Knowing the concept of Mobile first

### Contents

WebSite	W3Schools: HTML Responsive Web Design (	<a href="https://www.w3schools.com/html/html_responsive.asp">https://www.w3schools.com/html/html_responsive.asp</a>	)
WebSite	MDN Web Docs: Responsive design (	<a href="https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design">https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design</a>	)
Article	Responsive Web Design: What It Is And How To Use It (	<a href="https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/">https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/</a>	)
Article	The Beginner's Guide to Responsive Web Design (Code Samples & Layout Examples) (	<a href="https://kinsta.com/blog/responsive-web-design/">https://kinsta.com/blog/responsive-web-design/</a>	)
YouTube	Jesse Showalter: Responsive Web Design   10 Basics (	<a href="https://www.youtube.com/watch?v=zF6VSKy4Slc">https://www.youtube.com/watch?v=zF6VSKy4Slc</a>	)
YouTube	freeCodeCamp.org: Introduction To Responsive Web Design - HTML & CSS Tutorial (	<a href="https://www.youtube.com/watch?v=srvUrASNj0s">https://www.youtube.com/watch?v=srvUrASNj0s</a>	)
YouTube	Web Dev Simplified: Learn CSS Media Query In 7 Minutes (	<a href="https://www.youtube.com/watch?v=yU7jJ3NbPdA">https://www.youtube.com/watch?v=yU7jJ3NbPdA</a>	)

## Level 2

### JavaScript - Callbacks and Promises:

A Promise is a proxy for a value not necessarily known when the promise is created. This lets asynchronous methods return values like synchronous methods - instead of immediately returning the final value, the asynchronous method returns a promise to supply the value at some point in the future.

A callback function is a function passed into another function as an argument, which is then invoked inside the outer function to complete some kind of routine or action.

An async function is a function declared with the `async` keyword, and the `await` keyword is permitted within it. The `async` and `await` keywords enable asynchronous, promise-based behavior to be written in a cleaner style, avoiding the need to explicitly configure promise chains.

Understanding the concept of asynchronous programming

Writing asynchronous code understanding the concept of promises in JavaScript

Using JavaScript methods, keywords and objects for handling promises like 'Async/Await', '.then()', 'Promise', etc

Learning in which situations you need to use asynchronous programming

Calling APIs with `fetch()`

### Contents

WebSite	W3Schools: JavaScript Callbacks ( <a href="https://www.w3schools.com/js/js_callback.asp">https://www.w3schools.com/js/js_callback.asp</a> )	
WebSite	W3Schools: JavaScript Promises ( <a href="https://www.w3schools.com/js/js_promise.asp">https://www.w3schools.com/js/js_promise.asp</a> )	
WebSite	W3Schools: JavaScript Async ( <a href="https://www.w3schools.com/js/js_async.asp">https://www.w3schools.com/js/js_async.asp</a> )	
WebSite	W3Schools: Asynchronous JavaScript ( <a href="https://www.w3schools.com/js/js_asynchronous.asp">https://www.w3schools.com/js/js_asynchronous.asp</a> )	
WebSite	W3Schools: JavaScript Fetch API ( <a href="https://www.w3schools.com/js/js_api_fetch.asp">https://www.w3schools.com/js/js_api_fetch.asp</a> )	
WebSite	MDN Web Docs: Promise ( <a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise</a> )	)
WebSite	MDN Web Docs: Callback function ( <a href="https://developer.mozilla.org/en-US/docs/Glossary/Callback_function">https://developer.mozilla.org/en-US/docs/Glossary/Callback_function</a> )	)
WebSite	MDN Web Docs: async function ( <a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function</a> )	)
WebSite	MDN Web Docs: Introducing asynchronous JavaScript ( <a href="https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Introducing">https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Introducing</a> )	)
Article	A practical ES6 guide on how to perform HTTP requests using the Fetch API ( <a href="https://medium.com/free-code-camp/a-practical-es6-guide-on-how-to-perform-http-requests-using-the-fetch-api-594c3d91a547">https://medium.com/free-code-camp/a-practical-es6-guide-on-how-to-perform-http-requests-using-the-fetch-api-594c3d91a547</a> )	)
YouTube	James Q Quick: JavaScript Callbacks Explained in 5 Minutes ( <a href="https://www.youtube.com/watch?v=kz_vwAF4NHI">https://www.youtube.com/watch?v=kz_vwAF4NHI</a> )	)
YouTube	Code with Ania Kubów: Callbacks in JavaScript Explained ( <a href="https://www.youtube.com/watch?v=cNjIUSDnb9k">https://www.youtube.com/watch?v=cNjIUSDnb9k</a> )	)
YouTube	Web Dev Simplified: JavaScript Promises In 10 Minutes ( <a href="https://www.youtube.com/watch?v=DHvZLI7Db8E">https://www.youtube.com/watch?v=DHvZLI7Db8E</a> )	)
YouTube	Traversy Media: Async JS Crash Course - Callbacks, Promises, Async Await ( <a href="https://www.youtube.com/watch?v=PoRJizFvM7s">https://www.youtube.com/watch?v=PoRJizFvM7s</a> )	)



## JavaScript - Testing:

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

Using unit tests

Using integration testing

Using behavioral testing

Using mocks

### Contents

Article	An Overview of JavaScript Testing (	<a href="https://medium.com/welldone-software/an-overview-of-javascript-testing-7ce7298b9870">https://medium.com/welldone-software/an-overview-of-javascript-testing-7ce7298b9870</a>	)
Article	Unit Testing in JavaScript: A Tutorial (	<a href="https://www.browserstack.com/guide/unit-testing-in-javascript">https://www.browserstack.com/guide/unit-testing-in-javascript</a>	)
Article	How to Start Unit Testing Your JavaScript Code (	<a href="https://www.freecodecamp.org/news/how-to-start-unit-testing-javascript/">https://www.freecodecamp.org/news/how-to-start-unit-testing-javascript/</a>	)
Article	Test-Driven Development Tutorial – How to Test Your JavaScript and ReactJS Applications (	<a href="https://www.freecodecamp.org/news/test-driven-development-tutorial-how-to-test-javascript-and-reactjs-app/">https://www.freecodecamp.org/news/test-driven-development-tutorial-how-to-test-javascript-and-reactjs-app/</a>	)
YouTube	Academind: JavaScript Testing Introduction Tutorial - Unit Tests, Integration Tests & e2e Tests (	<a href="https://www.youtube.com/watch?v=r9HdJ8P6GQI">https://www.youtube.com/watch?v=r9HdJ8P6GQI</a>	)
YouTube	Fireship: Test-Driven Development - Fun TDD Introduction with JavaScript (	<a href="https://www.youtube.com/watch?v=Jv2uxzhPFI4">https://www.youtube.com/watch?v=Jv2uxzhPFI4</a>	)
YouTube	Traversy Media: Intro To JavaScript Unit Testing & BDD (	<a href="https://www.youtube.com/watch?v=u5cLK1UrFyQ">https://www.youtube.com/watch?v=u5cLK1UrFyQ</a>	)
YouTube	Web Dev Simplified: Introduction To Testing In JavaScript With Jest (	<a href="https://www.youtube.com/watch?v=FgnxcUQ5vho">https://www.youtube.com/watch?v=FgnxcUQ5vho</a>	)

## JavaScript - Error handling:

Error handling refers to the response and recovery procedures from error conditions present in a software application. In other words, it is the process comprised of anticipation, detection and resolution of application, programming or communication errors.

Knowing and handling the most common exceptions

Knowing the types of errors and in which situations they can occur

Using 'try' and 'catch' for error handling

Learning on what occasions and how to use `throw`

Creating specific exceptions according to your application's needs

### Contents

	MDN	
	Web	
	Docs:	
	Control	
WebSiteflow	<a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Control_flow_and_error_handling</a>	
	and	
	error	



	handling	
	(	
Article	Error Handling in JavaScript: a Quick Guide (	<a href="https://medium.com/swlh/error-handling-in-javascript-a-quick-guide-54b954427e47">https://medium.com/swlh/error-handling-in-javascript-a-quick-guide-54b954427e47</a> )
Article	Better error handling in JavaScript (	<a href="https://iaincollins.medium.com/error-handling-in-javascript-a6172ccdf9af">https://iaincollins.medium.com/error-handling-in-javascript-a6172ccdf9af</a> )
Article	JavaScript Best Practices — Error Handling Techniques (	<a href="https://medium.com/swlh/javascript-best-practices-error-handling-techniques-20a8752d92f6">https://medium.com/swlh/javascript-best-practices-error-handling-techniques-20a8752d92f6</a> )
Article	JavaScript Clean Code: Error Handling (	<a href="https://betterprogramming.pub/javascript-clean-code-error-handling-9cd7e87fdbe3">https://betterprogramming.pub/javascript-clean-code-error-handling-9cd7e87fdbe3</a> )
Article	JavaScript error handling best practices (	<a href="https://medium.com/@raygunio/javascript-error-handling-best-practices-ac5d4e667e09">https://medium.com/@raygunio/javascript-error-handling-best-practices-ac5d4e667e09</a> )
Article	JavaScript Error Handling from Express.js to React (	<a href="https://itnext.io/javascript-error-handling-from-express-js-to-react-810deb5e5e28">https://itnext.io/javascript-error-handling-from-express-js-to-react-810deb5e5e28</a> )
YouTube	freeCodeCamp.org: try, catch, finally, throw - error handling in JavaScript (	<a href="https://www.youtube.com/watch?v=cFTFtuEQ-10">https://www.youtube.com/watch?v=cFTFtuEQ-10</a> )
YouTube	dcode: Basics of Error Handling in JavaScript (	<a href="https://www.youtube.com/watch?v=G8Jux_bslXU">https://www.youtube.com/watch?v=G8Jux_bslXU</a> )
YouTube	Dave Gray: Javascript Error Handling (	<a href="https://www.youtube.com/watch?v=blBolyNhGvY">https://www.youtube.com/watch?v=blBolyNhGvY</a> )

## JavaScript - ES6:

ECMAScript 2015 was the second major revision to JavaScript. ECMAScript 2015 is also known as ES6 and ECMAScript 6.

Knowing the differences and most important features of ES6

### Contents

WebSiteW3Schools: Javascript ES6 ([https://www.w3schools.com/js/js\\_es6.asp](https://www.w3schools.com/js/js_es6.asp))

WebSiteJavascript Tutorial: ES6 Tutorial (<https://www.javascripttutorial.net/es6/>)

Article	JavaScript ES6 — write less, do more (	<a href="https://www.freecodecamp.org/news/write-less-do-more-with-javascript-es6-5fd4a8e50ee2/">https://www.freecodecamp.org/news/write-less-do-more-with-javascript-es6-5fd4a8e50ee2/</a> )
YouTube	Programming with Mosh: ES6 Tutorial - Learn Modern JavaScript in 1 Hour (	<a href="https://www.youtube.com/watch?v=NCwa_xi0Uuc">https://www.youtube.com/watch?v=NCwa_xi0Uuc</a> )
YouTube	freeCodeCamp.org: JavaScript ES6, ES7, ES8: Learn to Code on the Bleeding Edge (Full Course) (	<a href="https://www.youtube.com/watch?v=nZ1DMMsyVyl">https://www.youtube.com/watch?v=nZ1DMMsyVyl</a> )
YouTube	Udemy: ES6 JavaScript Tutorial for Beginners (	<a href="https://www.youtube.com/watch?v=uaBNBWwzV8">https://www.youtube.com/watch?v=uaBNBWwzV8</a> )

## JavaScript - Modularization:

JavaScript modules allow you to break up your code into separate files, which makes it easier to maintain the code-base.

Isolating parts of code in modules

Using import and export

### Contents

WebSiteW3Schools: JavaScript Modules ([https://www.w3schools.com/js/js\\_modules.asp](https://www.w3schools.com/js/js_modules.asp))

WebSite	MDN Web Docs: JavaScript modules (	<a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules</a> )
---------	------------------------------------	---

Article	Basics of Modular JavaScript (	<a href="https://medium.com/@crohacz_86666/basics-of-modular-javascript-2395c82dd93a">https://medium.com/@crohacz_86666/basics-of-modular-javascript-2395c82dd93a</a>	)
Article	How to Modularize Javascript Code (	<a href="https://codesource.io/how-to-modularize-javascript-code/">https://codesource.io/how-to-modularize-javascript-code/</a>	)
Article	Everything I know about writing modular JavaScript applications (	<a href="https://medium.com/dev-bits/everything-i-know-about-writing-modular-javascript-applications-37c125d8eddf">https://medium.com/dev-bits/everything-i-know-about-writing-modular-javascript-applications-37c125d8eddf</a>	)
YouTube	LearnCode.academy: Modular Javascript - Javascript Tutorial on the Object Literal Pattern (	<a href="https://www.youtube.com/watch?v=HkFIM73G-hk&amp;list=PLoYCgNOlyGABs-wDaaxChu82q_xQgUb4f">https://www.youtube.com/watch?v=HkFIM73G-hk&amp;list=PLoYCgNOlyGABs-wDaaxChu82q_xQgUb4f</a>	)
YouTube	Web Dev Simplified: JavaScript ES6 Modules (	<a href="https://www.youtube.com/watch?v=cRHQNNcYf6s">https://www.youtube.com/watch?v=cRHQNNcYf6s</a>	)

## Front-end Semantic Versioning:

SemVer (Semantic Versioning) is a simple set of rules and requirements that dictate how version numbers are assigned and incremented.

Organizing the dependencies of a project

Avoiding the "dependency hell"

### Contents

WebSiteDocumentation: Semantic Versioning (<https://semver.org/>)

Article	Semantic versioning for end-user applications (	<a href="https://uglow.medium.com/making-sense-of-semantic-versioning-for-end-user-software-applications-a3049d97478b">https://uglow.medium.com/making-sense-of-semantic-versioning-for-end-user-software-applications-a3049d97478b</a>	)
Article	Semantic versioning in npm (	<a href="https://javascript.plainenglish.io/semantic-versioning-in-npm-54d5b02c3536">https://javascript.plainenglish.io/semantic-versioning-in-npm-54d5b02c3536</a>	)
Article	Semantic Versioning (	<a href="https://jwnnz.medium.com/semantic-versioning-f0ac90d7a0fe">https://jwnnz.medium.com/semantic-versioning-f0ac90d7a0fe</a> )	
YouTube	Swashbuckling with Code: Understanding Semantic Versioning with Real World Examples (	<a href="https://www.youtube.com/watch?v=1zBzkT7QCmA">https://www.youtube.com/watch?v=1zBzkT7QCmA</a>	)
YouTube	Steve Griffith - Prof3ssorSt3v3: Semantic Versioning with NPM (	<a href="https://www.youtube.com/watch?v=mpkC6MmKgsQ">https://www.youtube.com/watch?v=mpkC6MmKgsQ</a>	)

## Jest:

Jest is a JavaScript test runner that lets you access the DOM via jsdom. It provides a great iteration speed combined with powerful features like mocking modules and timers so you can have more control over how the code executes.

Testing components

### Contents

WebSite	React: Documentation - Testing Overview (	<a href="https://reactjs.org/docs/testing.html#gatsby-focus-wrapper">https://reactjs.org/docs/testing.html#gatsby-focus-wrapper</a>	)
WebSite	Jest: Documentation (	<a href="https://jestjs.io/docs/getting-started">https://jestjs.io/docs/getting-started</a> )	
Article	Testing React Components with Jest and Enzyme - In Depth (	<a href="https://blog.bitsrc.io/how-to-test-react-components-with-jest-and-enzyme-in-depth-145fcd06b90">https://blog.bitsrc.io/how-to-test-react-components-with-jest-and-enzyme-in-depth-145fcd06b90</a>	)
Article	Testing in React with Jest and Enzyme - An Introduction (	<a href="https://rossbulat.medium.com/testing-in-react-with-jest-and-enzyme-an-introduction-99ce047dfcf8">https://rossbulat.medium.com/testing-in-react-with-jest-and-enzyme-an-introduction-99ce047dfcf8</a>	)
Article	Testing components with Jest and React Testing Library (	<a href="https://itnext.io/testing-components-with-jest-and-react-testing-library-d36f5262cde2">https://itnext.io/testing-components-with-jest-and-react-testing-library-d36f5262cde2</a>	)

YouTube	Lama Dev: React Testing Tutorial with React Testing Library and Jest (	<a href="https://www.youtube.com/watch?v=Flo268xRpV0">https://www.youtube.com/watch?v=Flo268xRpV0</a>	)
YouTube	PedroTech: Testing In React Tutorial - Jest and React Testing Library (	<a href="https://www.youtube.com/watch?v=JBSUgDxICg8">https://www.youtube.com/watch?v=JBSUgDxICg8</a>	)
YouTube	Traversy Media: Jest Crash Course - Unit Testing in JavaScript (	<a href="https://www.youtube.com/watch?v=7r4xVDI2vho">https://www.youtube.com/watch?v=7r4xVDI2vho</a>	)
YouTube	Web Dev Simplified: Introduction To Testing In JavaScript With Jest (	<a href="https://www.youtube.com/watch?v=FgnxcUQ5vho">https://www.youtube.com/watch?v=FgnxcUQ5vho</a>	)

## Cypress:

Cypress is a front-end testing tool that allows for setting up, writing, running, and debugging tests.

### Contents

WebSite	Cypress: Documentação (	<a href="https://docs.cypress.io/guides/overview/why-cypress">https://docs.cypress.io/guides/overview/why-cypress</a> )	
Article	Why Using Cypress Is Better Than Unit Testing (	<a href="https://betterprogramming.pub/why-using-cypress-is-better-than-unit-testing-e8234229be81">https://betterprogramming.pub/why-using-cypress-is-better-than-unit-testing-e8234229be81</a>	)
Article	How to Test Your Frontend with the Cypress.io Framework (	<a href="https://medium.com/free-code-camp/how-to-test-your-frontend-with-the-cypress-io-framework-f048070f4330">https://medium.com/free-code-camp/how-to-test-your-frontend-with-the-cypress-io-framework-f048070f4330</a>	)
Article	End to End Testing React Apps With Cypress (	<a href="https://blog.bitsrc.io/testing-react-apps-with-cypress-658bc482678">https://blog.bitsrc.io/testing-react-apps-with-cypress-658bc482678</a>	)
YouTube	LambdaTest: Learn Cypress in 3 Hours (	<a href="https://www.youtube.com/watch?v=jX3v3N6oN5M">https://www.youtube.com/watch?v=jX3v3N6oN5M</a> )	
YouTube	Cypress.io: Cypress in a Nutshell (	<a href="https://www.youtube.com/watch?v=LcGHiFnBh3Y">https://www.youtube.com/watch?v=LcGHiFnBh3Y</a> )	
YouTube	Fireship: Cypress End-to-End Testing (	<a href="https://www.youtube.com/watch?v=7N63cMKosIE">https://www.youtube.com/watch?v=7N63cMKosIE</a> )	

## Level 3

### Data Structures:

In the context of computers, the data structure is a specific way of storing and organizing data in the computer's memory so that these data can be easily retrieved and efficiently used when needed later.

Knowing the main data structures (linked list, stack, queue, tree, etc)

Implementing the main data structures

### Contents

WebSite	W3Schools: Data Structure Tutorial ( <a href="https://www.w3schools.in/data-structures/tutorials/">https://www.w3schools.in/data-structures/tutorials/</a> )	
WebSite	Geeks for Geeks: Data Structures ( <a href="https://www.geeksforgeeks.org/data-structures/">https://www.geeksforgeeks.org/data-structures/</a> )	
WebSite	Data Structures Tutorial ( <a href="https://www.javatpoint.com/data-structure-tutorial">https://www.javatpoint.com/data-structure-tutorial</a> )	
Article	8 Common Data Structures every Programmer must know ( <a href="https://towardsdatascience.com/8-common-data-structures-every-programmer-must-know-171acf6a1a42">https://towardsdatascience.com/8-common-data-structures-every-programmer-must-know-171acf6a1a42</a>	)
YouTube	Internet Made Coder: Data Structures Explained for Beginners - How I Wish I was Taught ( <a href="https://www.youtube.com/watch?v=ZdU4wMyiTSs">https://www.youtube.com/watch?v=ZdU4wMyiTSs</a>	)
YouTube	freeCodeCamp.org: Data Structures Easy to Advanced Course - Full Tutorial from a Google Engineer ( <a href="https://www.youtube.com/watch?v=ZdU4wMyiTSs">https://www.youtube.com/watch?v=ZdU4wMyiTSs</a>	)
YouTube	Aaron Jack: Data Structures you MUST know (as a Software Developer) ( <a href="https://www.youtube.com/watch?v=ZdU4wMyiTSs">https://www.youtube.com/watch?v=ZdU4wMyiTSs</a>	)

## Object-oriented Programming Concepts:

Object-oriented programming (OOP) is a programming paradigm based on the concept of 'objects', which can contain data and code: data in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). A common feature of objects is that procedures (or methods) are attached to them and can access and modify the object's data fields. Some of the main concepts are classes and instances, inheritance, and encapsulation.

How objects work

Creating and using constructors

What classes are

Creating and using Methods

How encapsulation works

What inheritance is

What polymorphism is

How interfaces work

What abstractions are

### Contents

WebSite	MDN Web Docs: Object-oriented programming (	<a href="https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_programming">https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_programming</a>	)
WebSite	Geeks for Geeks: Object Oriented Programming (OOPs) Concept in Java (	<a href="https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/">https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/</a>	)
WebSite	W3Schools: Object-Oriented Programming Style (	<a href="https://www.w3schools.blog/object-oriented-programming-style">https://www.w3schools.blog/object-oriented-programming-style</a>	)
Article	What is object-oriented programming? OOP explained in depth (	<a href="https://www.educative.io/blog/object-oriented-programming">https://www.educative.io/blog/object-oriented-programming</a>	)
YouTube	Programming with Mosh: Object-oriented Programming in 7 minutes (	<a href="https://www.youtube.com/watch?v=QJjY2TNyl-8">https://www.youtube.com/watch?v=QJjY2TNyl-8</a>	)
YouTube	freeCodeCamp.org: Intro to Object Oriented Programming - Crash Course (	<a href="https://www.youtube.com/watch?v=SiBw7os-zI">https://www.youtube.com/watch?v=SiBw7os-zI</a>	)
YouTube	Traversy Media: JavaScript OOP Crash Course (ES5 & ES6) (	<a href="https://www.youtube.com/watch?v=vDjPGenyHaA">https://www.youtube.com/watch?v=vDjPGenyHaA</a>	)

## JavaScript - Storage:

Local storage allows developers to store and retrieve data in the browser. The data stored in local storage will not expire, this means the data will persist even if the tab or the browser window is closed.

Storing data in the front-end with localStorage

Manipulating stored data

Persisting stored data

### Contents

WebSite	MDN Web Docs: Client-side storage (	<a href="https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Client-side_storage">https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Client-side_storage</a>	)
WebSite	MDN Web Docs: Using the Web Storage API (	<a href="https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API">https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API</a>	)

Article	localStorage in JavaScript: A complete guide (	<a href="https://blog.logrocket.com/localstorage-javascript-complete-guide/">https://blog.logrocket.com/localstorage-javascript-complete-guide/</a>	)
Article	How to Use Local Storage with JavaScript (	<a href="https://www.section.io/engineering-education/how-to-use-localstorage-with-javascript/">https://www.section.io/engineering-education/how-to-use-localstorage-with-javascript/</a>	)
YouTube	Web Dev Simplified: JavaScript Cookies vs Local Storage vs Session Storage (	<a href="https://www.youtube.com/watch?v=GihQAC1I39Q">https://www.youtube.com/watch?v=GihQAC1I39Q</a>	)
YouTube	ProMaker Dev: Local Storage Explained In 10 min   Javascript (	<a href="https://www.youtube.com/watch?v=fYTTUBa-IPc">https://www.youtube.com/watch?v=fYTTUBa-IPc</a>	)
YouTube	dcode: How to use Local Storage in JavaScript (	<a href="https://www.youtube.com/watch?v=k8yJCeuP6I8">https://www.youtube.com/watch?v=k8yJCeuP6I8</a>	)

## JavaScript - Concurrency:

Concurrency is a programming paradigm for building programs that make use of the simultaneous execution of several interactive computational tasks, which can be implemented as separate programs or as a set of threads created by a single program.

Executing tasks in parallel

### Contents

WebSite	MDN Web Docs: Thread	<a href="https://developer.mozilla.org/en-US/docs/Glossary/Thread">https://developer.mozilla.org/en-US/docs/Glossary/Thread</a>	
Article	Concurrency In JavaScript 1	<a href="https://dev.to/kalashin1/concurrency-in-javascript-4oho">https://dev.to/kalashin1/concurrency-in-javascript-4oho</a>	
Article	Concurrency in JavaScript 2 (	<a href="https://medium.com/@onejohi/concurrency-in-javascript-f5bb387708d8">https://medium.com/@onejohi/concurrency-in-javascript-f5bb387708d8</a>	)
Article	Multithreading using Javascript (	<a href="https://medium.com/@sarthakvit/multithreading-using-javascript-cf9ad7cf9cfe">https://medium.com/@sarthakvit/multithreading-using-javascript-cf9ad7cf9cfe</a>	)
Article	Multithreading Javascript (	<a href="https://medium.com/techtrument/multithreading-javascript-46156179cf9a">https://medium.com/techtrument/multithreading-javascript-46156179cf9a</a>	)
Article	How JavaScript works: parallelism vs concurrency in JavaScript, what's the difference? (	<a href="https://blog.sessionstack.com/how-javascript-works-parallelism-vs-concurrency-in-javascript-whats-the-difference-99fc3602b9f5">https://blog.sessionstack.com/how-javascript-works-parallelism-vs-concurrency-in-javascript-whats-the-difference-99fc3602b9f5</a>	)
YouTube	Coding Tech: Async/Await - Modern Concurrency In JavaScript (	<a href="https://www.youtube.com/watch?v=NsQ2QlrQShU">https://www.youtube.com/watch?v=NsQ2QlrQShU</a>	)
YouTube	Hack Reactor: Advanced Async and Concurrency Patterns in JavaScript (	<a href="https://www.youtube.com/watch?v=Qg1Svplau6U">https://www.youtube.com/watch?v=Qg1Svplau6U</a>	)
YouTube	JSConf: JavaScript Concurrency and the DOM (	<a href="https://www.youtube.com/watch?v=Qg1Svplau6U">https://www.youtube.com/watch?v=Qg1Svplau6U</a>	)

## TypeScript - Fundamentals:

TypeScript is a strongly typed programming language that builds on JavaScript.

Understanding in depth what types are and the importance of typed programming

Learning what TypeScript is, why it was created, how it works and its relationship with JavaScript

Knowing the TypeScript tools (integration with the code editor, static checker and compiler)

Writing code in TypeScript using its tools (interfaces, enum, decorators, etc.)

### Contents

WebSite	W3Schools: TypeScript Introduction (	<a href="https://www.w3schools.com/typescript/typescript_intro.php">https://www.w3schools.com/typescript/typescript_intro.php</a>	
WebSite	TypeScript Documentation	<a href="https://www.typescriptlang.org/docs/">https://www.typescriptlang.org/docs/</a>	

WebSite	Documentação: TypeScript for the New Programmer (	<a href="https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html">https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html</a>	)
WebSite	TypeScript - Understanding TypeScript (	<a href="https://learn.microsoft.com/en-us/archive/msdn-magazine/2015/january/typescript-understanding-typescript">https://learn.microsoft.com/en-us/archive/msdn-magazine/2015/january/typescript-understanding-typescript</a>	)
Article	W3Schools: Differences Between JavaScript And TypeScript (	<a href="https://www.w3schools.blog/differences-between-javascript-and-typescript">https://www.w3schools.blog/differences-between-javascript-and-typescript</a>	)
Article	W3Schools: TypeScript Tutorial: A step-by-step guide to learn TypeScript (	<a href="https://learningdaily.dev/typescript-tutorial-a-step-by-step-guide-to-learn-typescript-70f96b7ae7c8">https://learningdaily.dev/typescript-tutorial-a-step-by-step-guide-to-learn-typescript-70f96b7ae7c8</a>	)
YouTube	Programming with Mosh: TypeScript Tutorial for Beginners (	<a href="https://www.youtube.com/watch?v=d56mG7DezGs">https://www.youtube.com/watch?v=d56mG7DezGs</a>	)
YouTube	Academind: TypeScript Course for Beginners - Learn TypeScript from Scratch (	<a href="https://www.youtube.com/watch?v=BwuLxPH8IDs">https://www.youtube.com/watch?v=BwuLxPH8IDs</a>	)

## GraphQL:

GraphQL is a new API standard that provides a more efficient, powerful and flexible alternative to REST. It was developed and open-sourced by Facebook and is now maintained by a large community of companies and individuals from all over the world.

Understanding how GraphQL is used in API development

Creating APIs using GraphQL libraries and frameworks

### Contents

WebSite	GraphQL: Documentation ( <a href="https://graphql.org/">https://graphql.org/</a> )	)
WebSite	How to GraphQL: The Fullstack tutorial for GraphQL ( <a href="https://www.howtographql.com/">https://www.howtographql.com/</a> )	)
WebSite	RedHat: What is GraphQL? ( <a href="https://www.redhat.com/en/topics/api/what-is-graphql">https://www.redhat.com/en/topics/api/what-is-graphql</a> )	)
Article	So what's this GraphQL thing I keep hearing about? ( <a href="https://medium.com/free-code-camp/so-whats-this-graphql-thing-i-keep-hearing-about-baf4d36c20cf">https://medium.com/free-code-camp/so-whats-this-graphql-thing-i-keep-hearing-about-baf4d36c20cf</a> )	)
Article	Why GraphQL is the future of APIs ( <a href="https://medium.com/free-code-camp/why-graphql-is-the-future-of-apis-6a900fb0bc81">https://medium.com/free-code-camp/why-graphql-is-the-future-of-apis-6a900fb0bc81</a> )	)
Article	GraphQL: A Success Story for PayPal Checkout ( <a href="https://medium.com/paypal-tech/graphql-a-success-story-for-paypal-checkout-3482f724fb53">https://medium.com/paypal-tech/graphql-a-success-story-for-paypal-checkout-3482f724fb53</a> )	)
Article	GraphQL: Everything You Need to Know ( <a href="https://blog.weblab.technology/graphql-everything-you-need-to-know-58756ff253d8">https://blog.weblab.technology/graphql-everything-you-need-to-know-58756ff253d8</a> )	)
Article	REST vs. GraphQL: A Critical Review ( <a href="https://medium.com/good-api/rest-vs-graphql-a-critical-review-5f77392658e7">https://medium.com/good-api/rest-vs-graphql-a-critical-review-5f77392658e7</a> )	)
YouTube	Web Dev Simplified: Learn GraphQL In 40 Minutes ( <a href="https://www.youtube.com/watch?v=ZQL7tL2S0oQ">https://www.youtube.com/watch?v=ZQL7tL2S0oQ</a> )	)
YouTube	The Net Ninja: GraphQL Tutorial ( <a href="https://www.youtube.com/watch?v=Y0IDGjwRYKw&amp;list=PL4cUxeGkcC9iK6Qhn-QLcXCXPQUov1U7f">https://www.youtube.com/watch?v=Y0IDGjwRYKw&amp;list=PL4cUxeGkcC9iK6Qhn-QLcXCXPQUov1U7f</a> )	)
YouTube	Fireship: GraphQL Explained in 100 Seconds ( <a href="https://www.youtube.com/watch?v=elQh02xuVw4">https://www.youtube.com/watch?v=elQh02xuVw4</a> )	)

## Apollo Client:

Apollo Client is a comprehensive state management library for JavaScript that enables you to manage both local and remote data with GraphQL.

### Contents

WebSite	Documentation: Apollo Client	( <a href="https://www.apollographql.com/docs/react/">https://www.apollographql.com/docs/react/</a> )	
Article	Oh Hello Apollo Client, Goodbye Redux! (	<a href="https://kulkarniankita9.medium.com/oh-hello-apollo-client-goodbye-redux-49dfdeda16d1">https://kulkarniankita9.medium.com/oh-hello-apollo-client-goodbye-redux-49dfdeda16d1</a>	)
Article	Apollo Client, now with React Hooks (	<a href="https://www.apollographql.com/blog/announcement/frontend/apollo-client-now-with-react-hooks/">https://www.apollographql.com/blog/announcement/frontend/apollo-client-now-with-react-hooks/</a>	)
Article	Angular & Apollo Client: Getting Started with GraphQL in Angular (	<a href="https://www.bitovi.com/blog/angular-and-apollo-client-get-started-with-graphql-in-angular">https://www.bitovi.com/blog/angular-and-apollo-client-get-started-with-graphql-in-angular</a>	)
YouTube	PedroTech: GraphQL With React Tutorial - Apollo Client (	<a href="https://www.youtube.com/watch?v=YyUWW04HwKY">https://www.youtube.com/watch?v=YyUWW04HwKY</a>	)
YouTube	Fireship: GraphQL with Apollo Server 2.0 (	<a href="https://www.youtube.com/watch?v=8D9XnnjFGMs">https://www.youtube.com/watch?v=8D9XnnjFGMs</a>	)
YouTube	Laith Academy: React With GraphQL (Apollo Client) Crash Course (	<a href="https://www.youtube.com/watch?v=gAbIQx26wSI">https://www.youtube.com/watch?v=gAbIQx26wSI</a>	)

## Auxiliary Skill: Infrastructure and Back-end

### Git & GitHub - Fundamentals:

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

GitHub is a hosting service for software development and version control using Git.

Creating a repository

Cloning a repository

Committing, pushing and pulling to and from the repository

Reversing a commit

Creating branches and pull requests

Handling merge and conflicts

### Contents

WebSite	Git Reference Book	( <a href="https://git-scm.com/book/en/v2">https://git-scm.com/book/en/v2</a> )	
WebSite	GitHub Documentation	( <a href="https://docs.github.com/en">https://docs.github.com/en</a> )	
WebSite	GitHub Pages Documentation (	<a href="https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages">https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages</a>	)
WebSite	W3Schools: Git Tutorial	( <a href="https://www.w3schools.com/git/default.asp?remote=github">https://www.w3schools.com/git/default.asp?remote=github</a> )	
WebSite	Git School - Visualizing Git	( <a href="https://git-school.github.io/visualizing-git/">https://git-school.github.io/visualizing-git/</a> )	
WebSite	Dangit, Git!?! (	<a href="https://dangitgit.com/">https://dangitgit.com/</a> )	
Article	Git Tutorial - Explore The Commands And Operations In Git (	<a href="https://medium.com/edureka/git-tutorial-da652b566ece">https://medium.com/edureka/git-tutorial-da652b566ece</a>	)
Article	Git and Github Quickstart Tutorial (	<a href="https://medium.com/@prashantramnyc/git-and-github-quickstart-tutorial-654a71594dca">https://medium.com/@prashantramnyc/git-and-github-quickstart-tutorial-654a71594dca</a>	)
Article	Getting Started with Git and GitHub: A Complete Tutorial for Beginner (	<a href="https://towardsdatascience.com/learn-basic-git-commands-for-your-data-science-works-2a75396d530d">https://towardsdatascience.com/learn-basic-git-commands-for-your-data-science-works-2a75396d530d</a>	)
YouTube	Programming with Mosh: Git Tutorial for Beginners - Learn Git in 1 Hour (	<a href="https://www.youtube.com/watch?v=8JJ101D3knE">https://www.youtube.com/watch?v=8JJ101D3knE</a>	)
YouTube	freeCodeCamp.org: Git and GitHub for Beginners - Crash Course (	<a href="https://www.youtube.com/watch?v=RGQJ5yH7evk">https://www.youtube.com/watch?v=RGQJ5yH7evk</a>	)



YouTube	Kevin Stratvert: Git and GitHub for Beginners Tutorial (	<a href="https://www.youtube.com/watch?v=tRZGeaHPoaw">https://www.youtube.com/watch?v=tRZGeaHPoaw</a>	)
YouTube	Tech With Tim: Git Tutorial for Beginners - Git & GitHub Fundamentals In Depth (	<a href="https://www.youtube.com/watch?v=DVRQoVRzMIY">https://www.youtube.com/watch?v=DVRQoVRzMIY</a>	)

## HTTP - Fundamentals:

HTTP stands for Hyper Text Transfer Protocol. Communication between client computers and web servers is done by sending HTTP Requests and receiving HTTP Responses.

Understanding the difference between HTTP verbs

Testing requests and checking the status codes in the browser

Learning how to make a HTTP request on the command line with WGET

Downloading an image with WGET

Performing a POST

### Contents

WebSiteW3Schools: What is HTTP? ([https://www.w3schools.com/whatis/whatis\\_http.asp](https://www.w3schools.com/whatis/whatis_http.asp))

WebSite	MDN Web Docs: An overview of HTTP (	<a href="https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview">https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview</a>	)
---------	-------------------------------------	---	---

WebSite	MDN Web Docs: HTTP request methods (	<a href="https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods">https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods</a>	)
---------	--------------------------------------	---	---

WebSiteHTTP Cats (<https://http.cat/>)

WebSiteHTTP Dogs (<https://http.dog/>)

Article	HTTP codes as Valentine's Day comics (	<a href="https://medium.com/@hanilim/http-codes-as-valentines-day-comics-8c03c805faa0">https://medium.com/@hanilim/http-codes-as-valentines-day-comics-8c03c805faa0</a>	)
---------	--	---	---

Article	Here Are the most popular ways to make an HTTP request in JavaScript (	<a href="https://medium.com/free-code-camp/here-is-the-most-popular-ways-to-make-an-http-request-in-javascript-954ce8c95aaa">https://medium.com/free-code-camp/here-is-the-most-popular-ways-to-make-an-http-request-in-javascript-954ce8c95aaa</a>	)
---------	--	---	---

YouTube	Traversy Media: HTTP Crash Course & Exploration (	<a href="https://www.youtube.com/watch?v=iYM2zFP3Zn0">https://www.youtube.com/watch?v=iYM2zFP3Zn0</a>	)
---------	---	---	---

YouTube	Curious Code: HTTP Request Methods - GET, POST, PUT, DELETE (	<a href="https://www.youtube.com/watch?v=tkfVQK6UxDI">https://www.youtube.com/watch?v=tkfVQK6UxDI</a>	)
---------	---	---	---

YouTube	freeCodeCamp.org: Postman Beginner's Course - API Testing (	<a href="https://www.youtube.com/watch?v=VywxIQ2ZXw4">https://www.youtube.com/watch?v=VywxIQ2ZXw4</a>	)
---------	---	---	---

## JSON:

JSON stands for JavaScript Object Notation. It is a text format for storing and transporting data.

Creating an object

Transforming an object into a string

Transforming a string into an object

Manipulating an object

### Contents

WebSiteW3Schools: JSON ([https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp))

WebSite	MDN Web Docs: JSON (	<a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON</a>	)
---------	----------------------	---	---

WebSite	MDN Web Docs: Working with JSON (	<a href="https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON">https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON</a>	)
YouTube	Web Dev Simplified: Learn JSON in 10 Minutes (	<a href="https://www.youtube.com/watch?v=iiADhChRriM">https://www.youtube.com/watch?v=iiADhChRriM</a>	)
YouTube	Traversy Media: JSON Crash Course (	<a href="https://www.youtube.com/watch?v=w1CWzNtE-M">https://www.youtube.com/watch?v=w1CWzNtE-M</a>	)

## Command Line - Fundamentals:

CLI is a command line program that accepts text input to execute operating system functions.

Knowing the most important commands

### Contents

WebSite	W3Schools: What is Command Line Interface (CLI)? (	<a href="https://www.w3schools.com/whatis/whatis_cli.asp">https://www.w3schools.com/whatis/whatis_cli.asp</a> )
WebSite	Microsoft Docs: Using command line arguments for Windows Terminal (	<a href="https://docs.microsoft.com/en-us/windows/terminal/command-line-arguments?tabs=windows">https://docs.microsoft.com/en-us/windows/terminal/command-line-arguments?tabs=windows</a> )
Article	Advanced CLI: Commands You Should Know as a Developer (	<a href="https://betterprogramming.pub/advanced-cli-commands-you-should-know-as-a-developer-7bc48c752a5e">https://betterprogramming.pub/advanced-cli-commands-you-should-know-as-a-developer-7bc48c752a5e</a> )
YouTube	freeCodeCamp.org: Command Line Crash Course (	<a href="https://www.youtube.com/watch?v=yz7nYlnXLfE">https://www.youtube.com/watch?v=yz7nYlnXLfE</a> )
YouTube	Traversy Media: Command Line Crash Course For Beginners - Terminal Commands (	<a href="https://www.youtube.com/watch?v=uwAqEzhyjtw">https://www.youtube.com/watch?v=uwAqEzhyjtw</a> )

## Cloud - Fundamentals:

Cloud, or cloud computing, is the distribution of computing services over the Internet using a pay-as-you-go pricing model. A cloud is composed of various computing resources, ranging from the computers themselves (or instances, in cloud terminology) to networks, storage, databases, and everything around them. In other words, everything that is normally needed to set up the equivalent of a server room, or even a complete data center, will be ready to use, configured, and run.

Knowing the difference between IaaS, PaaS and SaaS

Knowing the largest cloud providers

Specializing in a specific provider of your choice

### Contents

WebSite	Microsoft Azure: What is cloud computing? (	<a href="https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing/">https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing/</a> )
WebSite	Amazon AWS: What is cloud computing? (	<a href="https://aws.amazon.com/en/what-is-cloud-computing/">https://aws.amazon.com/en/what-is-cloud-computing/</a> )
Article	A beginner's guide to the basics of what cloud computing is about (	<a href="https://scientya.com/a-beginners-guide-to-the-basics-of-what-cloud-computing-is-about-e8b3b7f25a30/">https://scientya.com/a-beginners-guide-to-the-basics-of-what-cloud-computing-is-about-e8b3b7f25a30/</a> )
Article	Cloud Computing for Beginners (	<a href="https://medium.com/hackernoon/cloud-computing-for-beginners-85d168959afb/">https://medium.com/hackernoon/cloud-computing-for-beginners-85d168959afb/</a> )
Article	What are Cloud Computing Services [IaaS, CaaS, PaaS, FaaS, SaaS] (	<a href="https://medium.com/@nnilesh7756/what-are-cloud-computing-services-iaas-caas-paas-faas-saas-ac0f6022d36e">https://medium.com/@nnilesh7756/what-are-cloud-computing-services-iaas-caas-paas-faas-saas-ac0f6022d36e</a> )
YouTube	Simplilearn: Cloud Computing Tutorial for Beginners (	<a href="https://www.youtube.com/watch?v=RWgW-CgdIkQ">https://www.youtube.com/watch?v=RWgW-CgdIkQ</a> )

YouTube	Amazon Web Services: What is Cloud Computing? - Amazon Web Services (	<a href="https://www.youtube.com/watch?v=mxT233EdY5c">https://www.youtube.com/watch?v=mxT233EdY5c</a>	)
YouTube	Ecourse Review: Cloud Computing Services Models - IaaS PaaS SaaS Explained (	<a href="https://www.youtube.com/watch?v=36zducUX16w/">https://www.youtube.com/watch?v=36zducUX16w/</a>	)

## YARN:

Yarn is a package manager for your code. It allows you to use and share code with other developers. Code is shared through something called a package (sometimes referred to as a module). A package contains all the code being shared as well as a package.json file which describes the package.

Managing packages

Managing dependencies

Installing packages offline

Commands

The yarn.lock file

## Contents

WebSiteYARN: Documentation (<https://classic.yarnpkg.com/en/docs>)

Article	NPM vs Yarn Cheat Sheet (	<a href="https://shift.infinite.red/npm-vs-yarn-cheat-sheet-8755b092e5cc">https://shift.infinite.red/npm-vs-yarn-cheat-sheet-8755b092e5cc</a>	)
Article	What are NPM, Yarn, Babel, and Webpack; and how to properly use them? (	<a href="https://medium.com/front-end-weekly/what-are-npm-yarn-babel-and-webpack-and-how-to-properly-use-them-d835a758f987">https://medium.com/front-end-weekly/what-are-npm-yarn-babel-and-webpack-and-how-to-properly-use-them-d835a758f987</a>	)
Article	npm vs Yarn — Choosing the right package manager (	<a href="https://javascript.plainenglish.io/npm-vs-yarn-choosing-the-right-package-manager-a5f04256a93f">https://javascript.plainenglish.io/npm-vs-yarn-choosing-the-right-package-manager-a5f04256a93f</a>	)
YouTube	Traversy Media: Yarn Package Manager Crash Course (	<a href="https://www.youtube.com/watch?v=g9_6KmiBISk">https://www.youtube.com/watch?v=g9_6KmiBISk</a>	)
YouTube	Clever Programmer: NPM vs Yarn - Which is the best Package Manager? (	<a href="https://www.youtube.com/watch?v=0DGCIZD5LEM">https://www.youtube.com/watch?v=0DGCIZD5LEM</a>	)

## Auxiliary Skill: UX and Design

### Design Systems:

A design system is a collection of reusable components, guided by clear standards, that can be assembled together to build applications.

Creating and maintaining libraries that will be consumed and used as a standard for building a project

## Contents

Article	Everything you need to know about Design Systems (	<a href="https://uxdesign.cc/everything-you-need-to-know-about-design-systems-54b109851969">https://uxdesign.cc/everything-you-need-to-know-about-design-systems-54b109851969</a>	)
Article	A comprehensive guide to design systems (	<a href="https://www.invisionapp.com/inside-design/guide-to-design-systems/">https://www.invisionapp.com/inside-design/guide-to-design-systems/</a>	)
Article	What is a UX Design System? (	<a href="https://fuzzymath.com/blog/what-is-ux-design-system/">https://fuzzymath.com/blog/what-is-ux-design-system/</a>	)
Article	Creating a design system in Figma — Tutorial 1 (	<a href="https://medium.com/@johan_ronsse/creating-a-design-system-in-figma-the-not-so-definite-guide-tutorial-1-8aa6801101cb">https://medium.com/@johan_ronsse/creating-a-design-system-in-figma-the-not-so-definite-guide-tutorial-1-8aa6801101cb</a>	)

YouTube	DesignerUp: 5 Best Design Systems and How to Learn (and Steal) From Them (	<a href="https://www.youtube.com/watch?v=BISC15OPeGA">https://www.youtube.com/watch?v=BISC15OPeGA</a>	)
YouTube	DesignCourse: What is a Design System? Design Systems 101 for Designers (	<a href="https://www.youtube.com/watch?v=wc5krC28ynQ">https://www.youtube.com/watch?v=wc5krC28ynQ</a>	)
YouTube	vaexperience: Design Systems - What are They and How to Get Started (	<a href="https://www.youtube.com/watch?v=l6YuGE6EjJA">https://www.youtube.com/watch?v=l6YuGE6EjJA</a>	)

## Figma - Fundamentals:

Figma is a collaborative web application for interface design. The feature set of Figma focuses on user interface and user experience design, with an emphasis on real-time collaboration, utilising a variety of vector graphics editor and prototyping tools.

Creating page layouts and components

### Contents

Article	Getting Started with Figma (	<a href="https://levelup.gitconnected.com/getting-started-with-figma-637f2c868017">https://levelup.gitconnected.com/getting-started-with-figma-637f2c868017</a>	)
Article	Introducing Figma to React (	<a href="https://medium.com/figma-design/introducing-figma-to-react-d2d545cba3cc">https://medium.com/figma-design/introducing-figma-to-react-d2d545cba3cc</a>	)
YouTube	Adrian Twarog: Figma Crash Course (	<a href="https://www.youtube.com/watch?v=lg7w3Ntfqy0">https://www.youtube.com/watch?v=lg7w3Ntfqy0</a>	)
YouTube	AJ&Smart: Figma UI Design Tutorial - Get Started in Just 24 Minutes (	<a href="https://www.youtube.com/watch?v=FTFaQWZBqQ8">https://www.youtube.com/watch?v=FTFaQWZBqQ8</a>	)
YouTube	Jesse Showalter: Intro to Figma - Beginners guide to Figma Basics (	<a href="https://www.youtube.com/watch?v=jk1T0CdLxwU">https://www.youtube.com/watch?v=jk1T0CdLxwU</a>	)
YouTube	Figma: Figma Tutorial - Components - The Basics (	<a href="https://www.youtube.com/watch?v=k74lrUNaJVk">https://www.youtube.com/watch?v=k74lrUNaJVk</a>	)
YouTube	Figma: Figma Components 101 (	<a href="https://www.youtube.com/watch?v=jk1T0CdLxwU">https://www.youtube.com/watch?v=jk1T0CdLxwU</a>	)
Course	LearnUX: Figma Course (	<a href="https://learnux.io/course/figma">https://learnux.io/course/figma</a>	)

## Design components:

Knowing the components that describe a layout or interface

### Contents

Article	Text fields & Forms design — UI components series (	<a href="https://uxdesign.cc/text-fields-forms-design-ui-components-series-2b32b2beebd0">https://uxdesign.cc/text-fields-forms-design-ui-components-series-2b32b2beebd0</a>	)
Article	Design Components for Emails (	<a href="https://blog.prototypy.io/design-components-for-emails-587f6267951d">https://blog.prototypy.io/design-components-for-emails-587f6267951d</a>	)
YouTube	GCFLearnFree.org: Beginning Graphic Design - Layout & Composition (	<a href="https://www.youtube.com/watch?v=a5KYIHnkQB8">https://www.youtube.com/watch?v=a5KYIHnkQB8</a>	)
YouTube	uxable - learning ux together: UI Elements/Components - Types and Importance of UI Elements (	<a href="https://www.youtube.com/watch?v=roF6ocfArK0">https://www.youtube.com/watch?v=roF6ocfArK0</a>	)
YouTube	Dennis Cortes: How to Design Powerful Components and Buttons with Variants and Auto Layout in Figma (	<a href="https://www.youtube.com/watch?v=ODOR5EniJNg">https://www.youtube.com/watch?v=ODOR5EniJNg</a>	)
YouTube	Tutorial Tim: Let's Build a Design System - Understanding Layout (	<a href="https://www.youtube.com/watch?v=nn3tO7gAFR4">https://www.youtube.com/watch?v=nn3tO7gAFR4</a>	)

## Color systems:

Defining a color palette that makes sense for a given interface

## Contents

Article	Color Theory: There is more than one set of universal Primary Colors (	<a href="https://medium.com/upskilling/color-theory-there-is-more-than-one-set-of-universal-primary-colors-debunking-the-myths-28140a7866c9">https://medium.com/upskilling/color-theory-there-is-more-than-one-set-of-universal-primary-colors-debunking-the-myths-28140a7866c9</a>	)
Article	The Science of Color (	<a href="https://medium.com/100-days-of-product-design/the-science-of-color-bd0f057c08ea">https://medium.com/100-days-of-product-design/the-science-of-color-bd0f057c08ea</a>	)
Article	RGB vs. CMYK: A guide to color systems for designers (	<a href="https://medium.com/envato/rgb-vs-cmyk-a-guide-to-color-systems-for-designers-6be8c1ed8554">https://medium.com/envato/rgb-vs-cmyk-a-guide-to-color-systems-for-designers-6be8c1ed8554</a>	)
Article	Color in UI Design: A (Practical) Framework (	<a href="https://medium.com/@erikdkennedy/color-in-ui-design-a-practical-framework-e18cacd97f9e">https://medium.com/@erikdkennedy/color-in-ui-design-a-practical-framework-e18cacd97f9e</a>	)
Article	Basic UI color guide (	<a href="https://blog.prototypr.io/basic-ui-color-guide-7612075cc71a">https://blog.prototypr.io/basic-ui-color-guide-7612075cc71a</a>	)
YouTube	Rachel How: How to pick the right colors for your website or app (	<a href="https://www.youtube.com/watch?v=ewRYw4pnKQU">https://www.youtube.com/watch?v=ewRYw4pnKQU</a>	)
YouTube	Flux Academy: How to Choose Colors (Easy 3-Step Process) (	<a href="https://www.youtube.com/watch?v=KMS3VwGh3HY">https://www.youtube.com/watch?v=KMS3VwGh3HY</a>	)
YouTube	Jesse Showalter: 60-30-10 Color Rule (	<a href="https://www.youtube.com/watch?v=UWwNIMHFdW4">https://www.youtube.com/watch?v=UWwNIMHFdW4</a>	)
YouTube	DesignerUp: Super Practical Guide to Color Theory, Color Models and Perfect Color Palettes (	<a href="https://www.youtube.com/watch?v=GyVMoejbGFg">https://www.youtube.com/watch?v=GyVMoejbGFg</a>	)

## How to use Fonts:

Choosing the most appropriate font for a given project

## Contents

Article	Top 10 UI Fonts for Web & Mobile (	<a href="https://blog.prototypr.io/top-10-ui-fonts-for-web-mobile-a8488e561ce3">https://blog.prototypr.io/top-10-ui-fonts-for-web-mobile-a8488e561ce3</a>	)
Article	Best Fonts for UI Design, I use Daily - Best Typefaces and Font Resources for UI Designers (	<a href="https://uxplanet.org/best-fonts-for-ui-design-i-use-daily-4a7bcffb966c">https://uxplanet.org/best-fonts-for-ui-design-i-use-daily-4a7bcffb966c</a>	)
YouTube	DesignerUp: Best Practices for Choosing Fonts and Font Pairing in UI and Web Design (	<a href="https://www.youtube.com/watch?v=30loKUGaPWQ">https://www.youtube.com/watch?v=30loKUGaPWQ</a>	)
YouTube	DesignerUp: The Difference Between Fonts, Typefaces and Typography for UI Designers (	<a href="https://www.youtube.com/watch?v=OKGTkLqgm58">https://www.youtube.com/watch?v=OKGTkLqgm58</a>	)
YouTube	DesignerUp: Choosing and Pairing Fonts - UI Design (	<a href="https://www.youtube.com/watch?v=uWCWTq1cPW0">https://www.youtube.com/watch?v=uWCWTq1cPW0</a>	)
YouTube	Mizko: The ONLY 8 Fonts UI Designers Need. Forget The Rest. (	<a href="https://www.youtube.com/watch?v=mEAmAFgzQd4">https://www.youtube.com/watch?v=mEAmAFgzQd4</a>	)