

# Angular

TechGuide - Alura, FIAP e PM3

---

## Angular

### 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

W3Schools: HTML Introduction  
WebSite ( [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp))

WebSite	W3Schools: HTML Elements (	<a href="https://www.w3schools.com/html/html_elements.asp">https://www.w3schools.com/html/html_elements.asp</a> )
WebSite	W3Schools: HTML Formatting Elements (	<a href="https://www.w3schools.com/html/html_formatting.asp">https://www.w3schools.com/html/html_formatting.asp</a> )
WebSite	W3Schools: HTML Block and Inline Elements (	<a href="https://www.w3schools.com/html/html_blocks.asp">https://www.w3schools.com/html/html_blocks.asp</a> )
WebSite	W3Schools: HTML Layout Elements (	<a href="https://www.w3schools.com/html/html_layout.asp">https://www.w3schools.com/html/html_layout.asp</a> )
WebSite	W3Schools: HTML Lists (	<a href="https://www.w3schools.com/html/html_lists.asp">https://www.w3schools.com/html/html_lists.asp</a> )
WebSite	W3Schools: HTML Tables (	<a href="https://www.w3schools.com/html/html_tables.asp">https://www.w3schools.com/html/html_tables.asp</a> )
WebSite	W3Schools: HTML Forms (	<a href="https://www.w3schools.com/html/html_forms.asp">https://www.w3schools.com/html/html_forms.asp</a> )
WebSite	MDN Web Docs: HTML basics (	<a href="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</a> )
Article	HTML vs CSS: What's the Difference? (	<a href="https://learn.onemonth.com/html-vs-css/">https://learn.onemonth.com/html-vs-css/</a> )
Article	Getting Started with HTML (	<a href="https://towardsdatascience.com/getting-started-with-html-98866c3e5c9e">https://towardsdatascience.com/getting-started-with-html-98866c3e5c9e</a> )
YouTube	Programming with Mosh: HTML Tutorial for Beginners - HTML Crash Course (	<a href="https://www.youtube.com/watch?v=qz0aGYrrlhU">https://www.youtube.com/watch?v=qz0aGYrrlhU</a> )
YouTube	freeCodeCamp.org: Learn HTML – Full Tutorial for Beginners (	<a href="https://www.youtube.com/watch?v=kUMe1FH4CHE">https://www.youtube.com/watch?v=kUMe1FH4CHE</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))

WebSiteMDN Web Docs: Getting started with CSS ([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))

WebSiteMDN Web Docs: CSS selectors ([https://developer.mozilla.org/en-US/docs/Learn/CSS/Building\\_blocks/Selectors](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors))

WebSiteMDN Web Docs: Cascade and inheritance ([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))

WebSiteMDN Web Docs: display property (<https://developer.mozilla.org/en-US/docs/Web/CSS/display>)

WebSiteMDN Web Docs: Basic ([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))

	concepts	
	of	
	Flexbox (	
	MDN Web	
	Docs:	
WebSite	Basic concepts of Grid Layout (	<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> )
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=W6NZfCO5Slk">https://www.youtube.com/watch?v=W6NZfCO5Slk</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=PkZNo7MFNFg">https://www.youtube.com/watch?v=PkZNo7MFNFg</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	<a href="https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html">https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html</a>	)

	New Programmer (	
	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-</a>
WebSiteUnderstanding		)
	TypeScript (	<a href="https://learn.microsoft.com/en-us/archive/msdn-magazine/2015/january/typescript-understanding-typescript">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>
		)
	W3Schools: TypeScript Tutorial: A	<a href="https://learningdaily.dev/typescript-tutorial-a-step-by-step-guide-to-learn-typescript-70f96b7ae7c8">https://learningdaily.dev/typescript-tutorial-a-</a>
Article	step-by-step guide to learn	<a href="https://learningdaily.dev/typescript-tutorial-a-step-by-step-guide-to-learn-typescript-70f96b7ae7c8">step-by-step-guide-to-learn-typescript-</a>
	TypeScript (	<a href="https://learningdaily.dev/typescript-tutorial-a-step-by-step-guide-to-learn-typescript-70f96b7ae7c8">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>
		)

## Angular - Fundamentos:

Angular is an application-design framework and development platform built on TypeScript for creating efficient and sophisticated single-page apps.

Building interfaces using HTML, CSS and TypeScript

Creating SPA applications

Building web, mobile or desktop applications

Integrating data with REST APIs

Using composition to create reusable components

Using Resolver services

Manipulating requests by creating Interceptor services

## Contents

WebSiteAngular - Documentation (<https://angular.io/guide/what-is-angular>)

Article	3 steps to understanding Angular (	<a href="https://dev.to/ikatsuba/3-steps-to-understanding-angular-hg0">https://dev.to/ikatsuba/3-steps-to-understanding-angular-hg0</a>	)
	Build A Real World Beautiful	<a href="https://hamedbaatour.medium.com/build-a-real-world-beautiful-web-app-with-angular-6-a-to-z-ultimate-guide-2018-part-i-e121dd1d55e">https://hamedbaatour.medium.com/build-a-real-</a>	
Article	Web APP with Angular — A to	<a href="https://hamedbaatour.medium.com/build-a-real-world-beautiful-web-app-with-angular-6-a-to-z-ultimate-guide-2018-part-i-e121dd1d55e">world-beautiful-web-app-with-angular-6-a-to-z-</a>	
	Z Ultimate Guide — PART I (	<a href="https://hamedbaatour.medium.com/build-a-real-world-beautiful-web-app-with-angular-6-a-to-z-ultimate-guide-2018-part-i-e121dd1d55e">ultimate-guide-2018-part-i-e121dd1d55e</a>	
Article	A practical guide to Angular	<a href="https://dev.to/this-is-angular/a-practical-guide-to-angular-template-driven-forms-bbi">https://dev.to/this-is-angular/a-practical-guide-</a>	
	Template-Driven Forms (	<a href="https://dev.to/this-is-angular/a-practical-guide-to-angular-template-driven-forms-bbi">to-angular-template-driven-forms-bbi</a>	)

YouTube	Fireship: Angular in 100 Seconds (	<a href="https://www.youtube.com/watch?v=Ata9cSC2WpM">https://www.youtube.com/watch?v=Ata9cSC2WpM</a>	)
YouTube	Traversy Media: Angular Crash Course (	<a href="https://www.youtube.com/watch?v=3dHNOWTI7H8">https://www.youtube.com/watch?v=3dHNOWTI7H8</a>	)
YouTube	freeCodeCamp.org: Angular for Beginners Course [Full Front End Tutorial with TypeScript] (	<a href="https://www.youtube.com/watch?v=3qBXWUpoPHo">https://www.youtube.com/watch?v=3qBXWUpoPHo</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-_zl">https://www.youtube.com/watch?v=SiBw7os-_zl</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>	)

## RxJS - Fundamentals:

RxJS is a library for composing asynchronous and event-based programs by using observable sequences. It provides one core type, the Observable, satellite types (Observer, Schedulers, Subjects) and operators inspired by Array methods (map, filter, reduce, every, etc) to allow handling asynchronous events as collections.

Creating asynchronous programs

Creating event-based programs

Understanding the concept of observables and sequences of observables

Understanding how to use Observers, Subscription, Subject and others

## Contents

WebSiteRxJS - Documentation (<https://rxjs.dev/guide/overview>)

WebSiteLearn RxJS - Documentation (<https://www.learnrxjs.io/>)

WebSiteAngular: The RxJS library (<https://angular.io/guide/rx-library>)

Article	Creating Observables in RxJS (	<a href="https://dev.to/thisdotmedia_staff/creating-observables-in-rxjs-2b3i">https://dev.to/thisdotmedia_staff/creating-observables-in-rxjs-2b3i</a>	)
Article	RxJS in Practice (	<a href="https://dev.to/thekiba/rxjs-in-practice-writing-our-own-ngrx-290i">https://dev.to/thekiba/rxjs-in-practice-writing-our-own-ngrx-290i</a>	)
YouTube	Fireship: RxJS Quick Start with Practical Examples (	<a href="https://www.youtube.com/watch?v=3n-WCiSxu2k">https://www.youtube.com/watch?v=3n-WCiSxu2k</a>	)
YouTube	Traversy Media: RxJS Observables Crash Course (	<a href="https://www.youtube.com/watch?v=ei7FsoXKPIQ">https://www.youtube.com/watch?v=ei7FsoXKPIQ</a>	)
YouTube	DesignCourse: Learn RxJS in 60 Minutes for Beginners (	<a href="https://www.youtube.com/watch?v=PhggNGsSQyg">https://www.youtube.com/watch?v=PhggNGsSQyg</a>	)

## Observer pattern:

In software design, the Observer pattern is a software design pattern in which an object, named the subject, maintains a list of its dependents, called observers, and notifies them automatically of any state changes, usually by calling one of their methods.

Understanding what Design Patterns are

Updating several elements simultaneously using Observers

Declaring subjects

Creating event-based programs

## Contents

WebSiteRefactoring Guru: Observer (<https://refactoring.guru/design-patterns/observer>)

Design Patterns — A quick [Article](https://medium.datadriveninvestor.com/design-patterns-a-quick-guide-to-observer-pattern-d0622145d6c2) guide to Observer pattern. ([patterns-a-quick-guide-to-observer-pattern-d0622145d6c2](https://medium.datadriveninvestor.com/design-patterns-a-quick-guide-to-observer-pattern-d0622145d6c2))

[Design Pattern] Observer [Article](https://dev.to/edindevto/design-pattern-observer-pattern-4ipb) Pattern (<https://dev.to/edindevto/design-pattern-observer-pattern-4ipb>)

Observer Design [Article](https://www.digitalocean.com/community/tutorials/observer-pattern-in-java) Pattern in Java (<https://www.digitalocean.com/community/tutorials/observer-pattern-in-java>)

The Observer Pattern in [Article](https://blog.bitsrc.io/the-observer-pattern-in-javascript-the-key-to-a-reactive-behavior-f28236e50e10) JavaScript — the Key to a [Reactive Behavior](https://blog.bitsrc.io/the-observer-pattern-in-javascript-the-key-to-a-reactive-behavior-f28236e50e10) (<https://blog.bitsrc.io/the-observer-pattern-in-javascript-the-key-to-a-reactive-behavior-f28236e50e10>)

The Observer Pattern [Article](https://medium.com/@patrickackerman/the-observer-pattern-with-vanilla-javascript-8f85ea05eaa8) with Vanilla JavaScript (<https://medium.com/@patrickackerman/the-observer-pattern-with-vanilla-javascript-8f85ea05eaa8>)

Angular University: Reactive Angular - The [YouTube](https://www.youtube.com/watch?v=jkp0k3MdW_U) Single Data Observable Pattern ([https://www.youtube.com/watch?v=jkp0k3MdW\\_U](https://www.youtube.com/watch?v=jkp0k3MdW_U))

MattStopaDev: How The Observer [YouTube](https://www.youtube.com/watch?v=WRkw0I72BL4) Pattern Works (<https://www.youtube.com/watch?v=WRkw0I72BL4>)

DevSage: Javascript Design Patterns - [YouTube](https://www.youtube.com/watch?v=45TeJEmcqk8) Observer Pattern (<https://www.youtube.com/watch?v=45TeJEmcqk8>)

## Level 2

### Angular - Templates:

In Angular, a template is a blueprint for a fragment of a user interface (UI). Templates are written in HTML, and special syntax can be used within a template to build on many of Angular's features.

#### Contents

WebSite	Angular Documentation - Understanding templates (	<a href="https://angular.io/guide/template-overview">https://angular.io/guide/template-overview</a> )
WebSite	Angular Documentation - Introduction to components and templates (	<a href="https://angular.io/guide/architecture-components">https://angular.io/guide/architecture-components</a> )
Article	An introduction to Angular Template Syntax (	<a href="https://javascript.plainenglish.io/brief-intro-to-angular-template-syntax-e05fe4f8c1b2">https://javascript.plainenglish.io/brief-intro-to-angular-template-syntax-e05fe4f8c1b2</a> )
Article	Why You Should Never Use Methods Inside Templates in Angular (	<a href="https://dev.to/klementomeri/why-you-should-never-use-methods-inside-templates-in-angular-2p3m">https://dev.to/klementomeri/why-you-should-never-use-methods-inside-templates-in-angular-2p3m</a> )
YouTube	edureka!: Angular Templates (	<a href="https://www.youtube.com/watch?v=jUmlOEyBUrY">https://www.youtube.com/watch?v=jUmlOEyBUrY</a> )
YouTube	Angular University: Angular ng-template - Introduction to Angular Template Partial (	<a href="https://www.youtube.com/watch?v=Qp0ghO3LLYI">https://www.youtube.com/watch?v=Qp0ghO3LLYI</a> )

### Angular - Rendering:

A normal Angular application executes in the browser, rendering pages in the DOM in response to user actions. Angular Universal executes on the server, generating static application pages that later get bootstrapped on the client.

Showing an Angular page on the browser

Performing Server-side rendering

#### Contents

WebSite	Server-side rendering (SSR) with Angular Universal (	<a href="https://angular.io/guide/universal">https://angular.io/guide/universal</a> )
Article	Angular Server-Side Rendering With Firebase — Step-By-Step Guide (	<a href="https://hamedbaatour.medium.com/angular-5-server-side-rendering-with-firebase-step-by-step-guide-ec5f4d8ef38b">https://hamedbaatour.medium.com/angular-5-server-side-rendering-with-firebase-step-by-step-guide-ec5f4d8ef38b</a> )

YouTube	Codeible: Server Side Rendering with Angular, <a href="https://www.youtube.com/watch?v=k7pLxaKkHYs">https://www.youtube.com/watch?v=k7pLxaKkHYs</a>	)
YouTube	Angular Universal, & Firebase (	)
YouTube	Angular: Hybrid Rendering In <a href="https://www.youtube.com/watch?v=RFwjJAZOzOA">https://www.youtube.com/watch?v=RFwjJAZOzOA</a>	)
	Angular (	)

## Angular - Services:

Service is a broad category encompassing any value, function, or feature that an application needs. A service is typically a class with a narrow, well-defined purpose. Angular distinguishes components from services to increase modularity and reusability.

Criar dados que estarão sempre disponíveis

Dividir a aplicação web em diversas partes

### Contents

WebSite	Angular Documentation - Introduction to services and dependency injection (	<a href="https://angular.io/guide/architecture-services">https://angular.io/guide/architecture-services</a>	)
WebSite	W3Schools: <a href="https://www.w3schools.com/angular/angular_services.asp">https://www.w3schools.com/angular/angular_services.asp</a>		)
Article	AngularJS Services (		)
Article	Angular Fundamentals: <a href="https://medium.com/@ZombieCodeKill/angular-fundamentals-creating-reusable-angular-services-405e64a7635b">https://medium.com/@ZombieCodeKill/angular-fundamentals-creating-reusable-angular-services-405e64a7635b</a>		)
Article	Creating Reusable Angular Services (		)
Article	Angular Service's Best Practices (	<a href="https://levelup.gitconnected.com/angular-services-best-practices-f3d26ddaeebd">https://levelup.gitconnected.com/angular-services-best-practices-f3d26ddaeebd</a>	)
YouTube	Codevolution: Angular Tutorial - Services (	<a href="https://www.youtube.com/watch?v=pwuGBvOPFYI">https://www.youtube.com/watch?v=pwuGBvOPFYI</a>	)
YouTube	procademy: Creating a Service in Angular   Services & Dependency Injection (	<a href="https://www.youtube.com/watch?v=cZV3_9oE8to">https://www.youtube.com/watch?v=cZV3_9oE8to</a>	)
YouTube	Simplilearn: Angular Services Tutorial   Angular Services Explained   Angular Tutorial For Beginners (	<a href="https://www.youtube.com/watch?v=0v1mQB4BCIo">https://www.youtube.com/watch?v=0v1mQB4BCIo</a>	)

## Angular - Routing:

In a Single Page Application (SPA), all of your application's functions exist in a single HTML page. As users access your application's features, the browser needs to render only the parts that matter to the user, instead of loading a new page. This pattern can significantly improve your application's user experience. To define how users navigate

through your application, you use routes. Add routes to define how users navigate from one part of your application to another. You can also configure routes to guard against unexpected or unauthorized behavior.

Navigating to a component

Including a route parameter

Controlling your user's navigation flow with route guarding

## Contents

WebSite	Angular Documentation - Angular Routing (	<a href="https://angular.io/guide/routing-overview">https://angular.io/guide/routing-overview</a>	)
WebSite	Angular Documentation - Using Angular routes in a single-page application (	<a href="https://angular.io/guide/router-tutorial">https://angular.io/guide/router-tutorial</a>	)
WebSite	W3Schools: AngularJS Routing (	<a href="https://www.w3schools.com/angular/angular_routing.asp">https://www.w3schools.com/angular/angular_routing.asp</a>	)
Article	Angular Routing Made Easy — Part 1 (	<a href="https://medium.com/coding-blocks/angular-6-routing-made-easy-part-1-f347c8a85947">https://medium.com/coding-blocks/angular-6-routing-made-easy-part-1-f347c8a85947</a>	)
Article	The Three Pillars of Angular Routing (	<a href="https://medium.com/angular-in-depth/the-three-pillars-of-angular-routing-angular-router-series-introduction-fb34e4e8758e">https://medium.com/angular-in-depth/the-three-pillars-of-angular-routing-angular-router-series-introduction-fb34e4e8758e</a>	)
YouTube	Codevolution: Angular Tutorial - Routing and Navigation (	<a href="https://www.youtube.com/watch?v=Nehk4tBxD4o">https://www.youtube.com/watch?v=Nehk4tBxD4o</a>	)
YouTube	Fireship: Angular Router - The Basics and Beyond (	<a href="https://www.youtube.com/watch?v=Np3ULAMqwNo">https://www.youtube.com/watch?v=Np3ULAMqwNo</a>	)

## Angular - CLI (Command-Line Interface):

The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications directly from a command shell.

Learning the syntax 'ng [optional-arg] [options]'

Getting to know important commands, like 'ng add', 'ng build', 'ng update', 'ng deploy', 'ng new', 'ng test', among others

## Contents

WebSite	Angular Documentation - CLI ( <a href="https://angular.io/cli">https://angular.io/cli</a> )	
Article	Angular Console — The UI for the Angular CLI ( <a href="https://blog.nrwl.io/angular-console-the-ui-for-the-angular-cli-a5d0924240b7">https://blog.nrwl.io/angular-console-the-ui-for-the-angular-cli-a5d0924240b7</a> )	

Article	6 Best Practices & Pro Tips when using Angular CLI (	<a href="https://tomastrajan.medium.com/6-best-practices-pro-tips-for-angular-cli-better-developer-experience-7b328bc9db81">https://tomastrajan.medium.com/6-best-practices-pro-tips-for-angular-cli-better-developer-experience-7b328bc9db81</a>	)
YouTube	Angular: Meet the Angular CLI (	<a href="https://www.youtube.com/watch?v=cRQFZDC7Fs8">https://www.youtube.com/watch?v=cRQFZDC7Fs8</a>	)
YouTube	Fireship: Angular CLI - The Basics (	<a href="https://www.youtube.com/watch?v=IZEoIKjcjks">https://www.youtube.com/watch?v=IZEoIKjcjks</a>	)

## Level 3

### Angular - State Management:

A 'state' is whatever data necessary in order to rebuild the UI at any moment in time. When this data change, it will trigger a redraw of the user interface. State management is the concept of adding, updating, removing and reading these data and their state in an application.

Updating components in real time

Waiting for changes in a component and performing changes

Using Redux, NGXS and others

#### Contents

Article	State Management: don't shoot yourself in the foot before you start an Angular application (	<a href="https://medium.com/@2muchcoffee/angular-state-management-a-must-have-for-large-scale-angular-apps-8b98e5a761c7">https://medium.com/@2muchcoffee/angular-state-management-a-must-have-for-large-scale-angular-apps-8b98e5a761c7</a>	)
Article	Angular State Managment — Don't fear the boilerplate (	<a href="https://codeburst.io/state-management-in-angular-ee2ccb81c283">https://codeburst.io/state-management-in-angular-ee2ccb81c283</a>	)
Article	State Management in Angular Using NGXS (	<a href="https://auth0.com/blog/state-management-in-angular-with-ngxs-part-1/">https://auth0.com/blog/state-management-in-angular-with-ngxs-part-1/</a>	)
Article	Angular State Management Without RxJS — An Experiment (	<a href="https://medium.com/swlh/angular-state-management-without-rxjs-an-experiment-243de024d396">https://medium.com/swlh/angular-state-management-without-rxjs-an-experiment-243de024d396</a>	)
YouTube	Fireship: NGXS - Angular State Management (	<a href="https://www.youtube.com/watch?v=SGj11j4hxmg">https://www.youtube.com/watch?v=SGj11j4hxmg</a>	)
YouTube	Decoded Frontend: State Management in Angular - Do I need Redux & NgRx? (	<a href="https://www.youtube.com/watch?v=2LwuV6s-Cy4">https://www.youtube.com/watch?v=2LwuV6s-Cy4</a>	)

## Angular - Forms:

A web form is an online page that allows for user input. It is an interactive page that mimics a paper document or form, where users fill out particular fields.

Creating forms with Template Forms

Creating reactive forms with Reactive Forms

### Contents

WebSite	Angular Documentation - Introduction to forms in Angular (	<a href="https://angular.io/guide/forms-overview">https://angular.io/guide/forms-overview</a>	)
WebSite	Angular Documentation - Building a template-driven form (	<a href="https://angular.io/guide/forms">https://angular.io/guide/forms</a>	)
WebSite	Angular Documentation - Reactive forms (	<a href="https://angular.io/guide/reactive-forms">https://angular.io/guide/reactive-forms</a>	)
Article	Angular Forms: Useful Tips (	<a href="https://medium.com/angular-in-depth/angular-forms-useful-tips-9f3a9826292e">https://medium.com/angular-in-depth/angular-forms-useful-tips-9f3a9826292e</a>	)
Article	Angular Forms in Depth (	<a href="https://blog.nrwl.io/angular-forms-in-depth-ecb7c58166b5">https://blog.nrwl.io/angular-forms-in-depth-ecb7c58166b5</a>	)
YouTube	Programming with Mosh: Building Forms in Angular Apps (	<a href="https://www.youtube.com/watch?v=hAaoPOx_olw">https://www.youtube.com/watch?v=hAaoPOx_olw</a>	)
YouTube	Simplilearn: Angular Forms Tutorial (	<a href="https://www.youtube.com/watch?v=-bGgjgx3fGs">https://www.youtube.com/watch?v=-bGgjgx3fGs</a>	)
YouTube	Fireship: Reactive Forms - The Basics (	<a href="https://www.youtube.com/watch?v=JeeUY6WaXiA">https://www.youtube.com/watch?v=JeeUY6WaXiA</a>	)

## Angular - Modules:

A Module, unlike a Component, does not control any view. A Module consists of one or more Components. An Angular application must have at least one Module that contains at least one Component.

Using the NgModule class

Declaring which components can be used by components from other modules

Declaring which services will be injected

Learning how to modularize an application

Lazy-loading modules

## Contents

WebSite	Angular Documentation - Introduction to modules (	<a href="https://angular.io/guide/architecture-modules">https://angular.io/guide/architecture-modules</a>	)
WebSite	Angular Documentation - NgModules (	<a href="https://angular.io/guide/ngmodules">https://angular.io/guide/ngmodules</a>	)
Article	A Complete Guide to Angular Modules (	<a href="https://levelup.gitconnected.com/a-complete-guide-to-angular-modules-faf5a85e3e60">https://levelup.gitconnected.com/a-complete-guide-to-angular-modules-faf5a85e3e60</a>	)
Article	Understanding Angular Modules (	<a href="https://itnext.io/understanding-angular-modules-5f1215130bc8">https://itnext.io/understanding-angular-modules-5f1215130bc8</a>	)
Article	Lazy loading modules in Angular (	<a href="https://dev.to/achimoraites/lazy-loading-modules-in-angular-3ln9">https://dev.to/achimoraites/lazy-loading-modules-in-angular-3ln9</a>	)
YouTube	The Net Ninja: AngularJS Tutorial - Modules (	<a href="https://www.youtube.com/watch?v=ybnw5JwBk8g">https://www.youtube.com/watch?v=ybnw5JwBk8g</a>	)
YouTube	Fireship: Learn NgModule in Angular with Examples (	<a href="https://www.youtube.com/watch?v=oqZ4-ULwfbC">https://www.youtube.com/watch?v=oqZ4-ULwfbC</a>	)
YouTube	Code Step By Step: Angular tutorial - Lazy Loading (	<a href="https://www.youtube.com/watch?v=JlQq9lh-Bw">https://www.youtube.com/watch?v=JlQq9lh-Bw</a>	)

## Angular - Dependency Injection:

Dependency Injection (DI) allows you to declare the dependencies of your TypeScript classes without taking care of their instantiation. Instead, Angular handles the instantiation for you. This design pattern allows you to write more testable and flexible code.

Declaring your classes' dependencies

Injecting dependencies into a Component

Getting to know Injection Tokens

Configuring providers

Understanding Hierarchical Dependency Injection

## Contents

WebSite	Angular Documentation - Understanding dependency injection (	<a href="https://angular.io/guide/dependency-injection">https://angular.io/guide/dependency-injection</a>	)
WebSite	Angular Documentation - Dependency injection in action (	<a href="https://angular.io/guide/dependency-injection-in-action">https://angular.io/guide/dependency-injection-in-action</a>	)



WebSite	Angular Dependency Injection: Complete Guide (	<a href="https://blog.angular-university.io/angular-dependency-injection/">https://blog.angular-university.io/angular-dependency-injection/</a>	)
Article	Angular Dependency Injection Explained with Examples (	<a href="https://www.freecodecamp.org/news/angular-dependency-injection/">https://www.freecodecamp.org/news/angular-dependency-injection/</a>	)
YouTube	Codevolution: Angular Tutorial - Dependency Injection (	<a href="https://www.youtube.com/watch?v=OFPIGlxunL0">https://www.youtube.com/watch?v=OFPIGlxunL0</a>	)
YouTube	Decoded Frontend: Angular Dependency Injection – Understanding hierarchical injectors (	<a href="https://www.youtube.com/watch?v=G8zXugcYd7o">https://www.youtube.com/watch?v=G8zXugcYd7o</a>	)

## Angular - Testes:

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

Getting acquainted with Jasmine and/or Karma

## Contents

WebSite	Angular Documentation - Angular Testing (	<a href="https://angular.io/guide/testing">https://angular.io/guide/testing</a> )
Article	A Complete Guide To AngularJS Testing (	<a href="https://dev.to/lambdatest/a-complete-guide-to-angularjs-testing-lf3">https://dev.to/lambdatest/a-complete-guide-to-angularjs-testing-lf3</a> )
Article	Intro to Testing in Angular (	<a href="https://dev.to/prestonjamb/intro-to-testing-in-angular-4o3m">https://dev.to/prestonjamb/intro-to-testing-in-angular-4o3m</a> )
Article	An Angular Testing Cheatsheet (	<a href="https://dev.to/lysofdev/an-angular-testing-cheatsheet-5hj2">https://dev.to/lysofdev/an-angular-testing-cheatsheet-5hj2</a> )
YouTube	Fireship: Angular Testing Quick Start (	<a href="https://www.youtube.com/watch?v=BumgayeUC08">https://www.youtube.com/watch?v=BumgayeUC08</a> )
YouTube	Genka: Angular Unit Test Tutorial - Crash Course for Angular & Jasmine (	<a href="https://www.youtube.com/watch?v=ibatZSCgXLY">https://www.youtube.com/watch?v=ibatZSCgXLY</a> )
YouTube	Oasis Digital: Angular component testing (	<a href="https://www.youtube.com/watch?v=xJ45MGDAi6c">https://www.youtube.com/watch?v=xJ45MGDAi6c</a> )

Angular University: Angular Testing - <https://www.youtube.com/watch?v=7Qd0d4ViiDk&list=PL0a5YlicjJ-XA6uwvBtOLtsVvBHP6g3v>  
YouTube Step-by-Step Test Implementation (Playlist) ( )

## 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 (<https://graphql.org/>)

WebSite How to GraphQL: The Fullstack tutorial for GraphQL (<https://www.howtographql.com/>)

WebSite RedHat: What is GraphQL? (<https://www.redhat.com/en/topics/api/what-is-graphql>)

Article So what's this GraphQL  
Article thing I keep hearing  
Article about? (<https://medium.com/free-code-camp/so-whats-this-graphql-thing-i-keep-hearing-about-baf4d36c20cf>)

Article Why GraphQL is the  
Article future of APIs (<https://medium.com/free-code-camp/why-graphql-is-the-future-of-apis-6a900fb0bc81>)

Article GraphQL: A Success Story  
Article for PayPal Checkout (<https://medium.com/paypal-tech/graphql-a-success-story-for-paypal-checkout-3482f724fb53>)

Article GraphQL: Everything You  
Article Need to Know (<https://blog.weblab.technology/graphql-everything-you-need-to-know-58756ff253d8>)

Article REST vs. GraphQL: A  
Article Critical Review (<https://medium.com/good-api/rest-vs-graphql-a-critical-review-5f77392658e7>)

YouTube Web Dev Simplified: Learn GraphQL In  
YouTube 40 Minutes (<https://www.youtube.com/watch?v=ZQL7tL2S0oQ>)

YouTube The Net Ninja:  
YouTube GraphQL Tutorial (<https://www.youtube.com/watch?v=Y0IDGjwRYKw&list=PL4cUxeGkcC9iK6Qhn-QLcXCXPQUov1U7f>)

YouTube Fireship: GraphQL Explained in 100  
YouTube Seconds (<https://www.youtube.com/watch?v=eIQh02xuVw4>)

## 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

WebSiteDocumentation: Apollo Client (<https://www.apollographql.com/docs/react/>)

Article Oh Hello Apollo Client, <https://kulkarniankita9.medium.com/oh-hello-apollo-client-goodbye-redux-49dfdeda16d1> )  
Goodbye Redux! (

Apollo Client, <https://www.apollographql.com/blog/announcement/frontend/apollo-client-now-with-react-hooks/> )  
Article now with React Hooks (

Angular & Apollo Client: Getting <https://www.bitovi.com/blog/angular-and-apollo-client-get-started-with-graphql-in-angular> )  
Article Started with GraphQL in Angular (

PedroTech: GraphQL With React Tutorial <https://www.youtube.com/watch?v=YyUWW04HwKY> )  
YouTube - Apollo Client (

Fireship: GraphQL with Apollo <https://www.youtube.com/watch?v=8D9XnnjFGMs> )  
YouTube Server 2.0 (

Laith Academy: React With GraphQL (Apollo <https://www.youtube.com/watch?v=gAbIQx26wSI> )  
YouTube Client) Crash Course (

## 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

WebSiteGit Reference Book (<https://git-scm.com/book/en/v2>)

WebSiteGitHub Documentation (<https://docs.github.com/en>)

WebSiteGitHub Pages Documentation ( <https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages> )

WebSiteW3Schools: Git Tutorial (<https://www.w3schools.com/git/default.asp?remote=github> )

WebSiteGit School - Visualizing Git (<https://git-school.github.io/visualizing-git/>)

WebSiteDangit, Git!?! (<https://dangitgit.com/>)

ArticleGit Tutorial - Explore The Commands And Operations In Git ( <https://medium.com/edureka/git-tutorial-da652b566ece> )

ArticleGit and Github Quickstart Tutorial ( <https://medium.com/@prashantramnyc/git-and-github-quickstart-tutorial-654a71594dca> )

ArticleGetting Started with Git and GitHub: A Complete Tutorial for Beginner ( <https://towardsdatascience.com/learn-basic-git-commands-for-your-data-science-works-2a75396d530d> )

YouTubeProgramming with Mosh: Git Tutorial for Beginners - Learn Git in 1 Hour ( <https://www.youtube.com/watch?v=8JJ101D3knE> )

YouTubefreeCodeCamp.org: Git and GitHub for Beginners - Crash Course ( <https://www.youtube.com/watch?v=RGQj5yH7evk> )

YouTubeKevin Stratvert: Git and GitHub for Beginners Tutorial ( <https://www.youtube.com/watch?v=tRZGeaHPoaw> )

YouTubeTech With Tim: Git Tutorial for Beginners - Git & GitHub Fundamentals In Depth ( <https://www.youtube.com/watch?v=DVRQoVRzMIY> )

## 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

WebSite	W3Schools: What is HTTP? (	<a href="https://www.w3schools.com/whatis/whatis_http.asp">https://www.w3schools.com/whatis/whatis_http.asp</a> )
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> )
WebSite	HTTP Cats (	<a href="https://http.cat/">https://http.cat/</a> )
WebSite	HTTP Dogs (	<a href="https://http.dog/">https://http.dog/</a> )
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

WebSite	W3Schools: JSON (	<a href="https://www.w3schools.com/js/js_json_intro.asp">https://www.w3schools.com/js/js_json_intro.asp</a> )
---------	-------------------	---

WebSite	MDN Web Docs: <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=wl1CWzNtE-M">https://www.youtube.com/watch?v=wl1CWzNtE-M</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-CgdIk0">https://www.youtube.com/watch?v=RWgW-CgdIk0</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>	)

## 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>	)
Articles	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.prototypr.io/design-components-for-emails-587f6267951d">https://blog.prototypr.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 - <https://www.youtube.com/watch?v=nn3tO7gAFR4> )  
Understanding Layout (

## Color systems:

Defining a color palette that makes sense for a given interface

### Contents

Color Theory: There is <https://medium.com/upskilling/color-theory-there-is-more-than-one-set-of-universal-primary-colors-debunking-the-myths-28140a7866c9> )  
Article more than one set of universal Primary Colors (

The Science of <https://medium.com/100-days-of-product-design/the-science-of-color-bd0f057c08ea> )  
Article Color (

RGB vs. CMYK: A guide to <https://medium.com/envato/rgb-vs-cmyk-a-guide-to-color-systems-for-designers-6be8c1ed8554> )  
Article color systems for designers (

Color in UI Design: A <https://medium.com/@erikdkennedy/color-in-ui-design-a-practical-framework-e18cacd97f9e> )  
Article (Practical) Framework (

Basic UI color guide <https://blog.prototypr.io/basic-ui-color-guide-7612075cc71a> )  
Article (

Rachel How: How to pick the right colors for <https://www.youtube.com/watch?v=ewRYw4pnKQU> )  
YouTube your website or app (

Flux Academy: How to Choose Colors <https://www.youtube.com/watch?v=KMS3VwGh3HY> )  
YouTube (Easy 3-Step Process) (

Jesse Showalter: 60-30-10 Color <https://www.youtube.com/watch?v=UWwNIMHFdW4> )  
YouTube Rule (

DesignerUp: Super Practical Guide to Color <https://www.youtube.com/watch?v=GyVMoejbGFg> )  
YouTube Theory, Color Models and Perfect Color Palettes (

## How to use Fonts:

Choosing the most appropriate font for a given project

### Contents

Top 10 UI Fonts for Web & <https://blog.prototypr.io/top-10-ui-fonts-for-web-mobile-a8488e561ce3> )  
Article Mobile (

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>	)

---

TechGuide - Alura  
Alura, PM3 e FIAP  
O Techguide.sh é um projeto open source