**JavaScript:**

Closure: <https://www.w3schools.com/js/js_function_closures.asp>

Histing: <https://scotch.io/tutorials/understanding-hoisting-in-javascript>

Strict mode: <https://javascript.info/strict-mode>

Object properties & descriptors: <https://javascript.info/property-descriptors>

"this" keyword: <https://www.geeksforgeeks.org/this-in-javascript/>

Indexed & Keyed collections: <http://webmobtuts.com/javascript/javascript-keyed-and-indexed-collections-array-map-and-set/>

Map, reduce & filter: <https://danmartensen.svbtle.com/javascripts-map-reduce-and-filter>

Memory management: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Memory_Management>

Error handling: <https://www.tutorialspoint.com/javascript/javascript_error_handling.htm>

Debugging: <https://developers.google.com/web/tools/chrome-devtools/javascript/>

**Object Oriented JavaScript:**

Basics: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_JS>

Abstraction, Encapsulation, Inheritance & Polymorphism: <https://javascriptissexy.com/oop-in-javascript-what-you-need-to-know/>

Inheritance & prototype  chain: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Inheritance_and_the_prototype_chain>

SOLID Principals: <https://medium.com/@cramirez92/s-o-l-i-d-the-first-5-priciples-of-object-oriented-design-with-javascript-790f6ac9b9fa>

JavaScript Principals Design Patterns: <https://scotch.io/bar-talk/4-javascript-design-patterns-you-should-know>

==============

**TypeScript:**

Abstraction, Encapsulation, Inheritance & Polymorphism: <https://rachelappel.com/2015/01/02/write-object-oriented-javascript-with-typescript/>

REST, default & optional params: <https://howtodoinjava.com/typescript/functions-rest-optional-default-params/>

Overloading: <https://www.bennadel.com/blog/3339-using-method-and-function-overloading-in-typescript.htm>

Iterators & decorators: <https://www.typescriptlang.org/docs/handbook/iterators-and-generators.html>

Intersections & union types: <https://www.typescriptlang.org/docs/handbook/advanced-types.html>

Decorators: <https://www.typescriptlang.org/docs/handbook/decorators.html>

===============

**Angular:**

Bootstrapping: <https://angular.io/guide/bootstrapping>

Lifecycle hooks: <https://www.cuelogic.com/blog/angular-lifecycle>

Routing: <https://blog.angularindepth.com/the-three-pillars-of-angular-routing-angular-router-series-introduction-fb34e4e8758e>

Router gaurd: <https://codecraft.tv/courses/angular/routing/router-guards/>

Dependency injection (injectors, providers): <https://angular.io/guide/dependency-injection>

Forms: <https://angular-templates.io/tutorials/about/angular-forms-and-validations>

Directives: <https://www.sitepoint.com/practical-guide-angular-directives/>

HostListener & HostBinding: <https://codecraft.tv/courses/angular/custom-directives/hostlistener-and-hostbinding/>

Pipe: <https://scotch.io/tutorials/create-a-globally-available-custom-pipe-in-angular-2>

Component communications: <https://medium.com/@mirokoczka/3-ways-to-communicate-between-angular-components-a1e3f3304ecb>

ViewChildren, ViewChild, ContentChildren & ContentChild: <https://medium.com/@tkssharma/understanding-viewchildren-viewchild-contentchildren-and-contentchild-b16c9e0358e>

Services: <https://angular.io/tutorial/toh-pt4>

HTTP Client: <https://www.techiediaries.com/angular-http-client/>

Web workers: <https://medium.com/@damoresac/using-web-workers-on-angular-6-6fd0490d07b5>

Base project structure & Webpack config: <https://jasonwatmore.com/post/2019/04/24/angular-7-tutorial-part-2-create-base-project-structure-webpack-config>

AOT: <https://angular.io/guide/aot-compiler>

Unit Tesitng: <https://medium.com/@selvarajchinnasamyks/angular-7-unit-testing-97dccfdca900>

Mock backend: <https://jasonwatmore.com/post/2019/05/02/angular-7-mock-backend-example-for-backendless-development>

===============

**NgRx:**

State, actions & reducers: <https://dzone.com/articles/managing-state-in-angular-with-ngrxstore>

<https://blog.angularindepth.com/ngrx-tips-tricks-69feb20a42a7>

===============

**More links for in-depth reference:**

JavaScript:  <https://www.w3schools.com/js/>

TypeScript:  <https://www.typescriptlang.org/docs/handbook/basic-types.html>

Angular:  <https://angular.io/guide/architecture>

Unit Testing:  <https://angular.io/guide/testing>

RxJS:  <https://rxjs-dev.firebaseapp.com/guide/overview>

NgRx:  <https://ngrx.io/guide/store>

Angular

--------------

Why we should use Angular?

How do we organise folder structure.

Angular lifecycle

State management in angular.

If i want to deploy my angular project to prod what are the things need to do with the project.

(--prod, --aot)

Change detection strategy

Httpinterceptor

Data communication between components

Cutom filters, directive

Rxjs operator- forkjoin, switchmap, mergemap etc.

If asked to design a new project what are steps needto take. How do we organise the module.

JavaScript

---------------

Closures

Variable Scope

Prototype

How to create object.

Oops concept

Es6 features

Shallow copy, deep copy of object.

Array functions