**What is Angular Framework?**

Angular is a Typescript-based open-source front-end platform which is maintained by Google.

**Steps to setup Angular**

* Install Node.js
* Install Visual Studio Code

To run angular project from VS Code you need to setup the Execution Policy. Below is the command to set the execution policy is :

**Set-ExecutionPolicy RemoteSigned -Scope CurrentUser**

* **Install Angular CLI**

npm install –g @angular/cli

* **Create new application**

*ng new MyFirstAngular*

* **To run the application**

First go to project root folder then run **ng serve –o**

* **To see the Angular version**

Go to project root folder then run **ng –v**

## **1. Differentiate between Angular and AngularJS.**

|  |  |  |
| --- | --- | --- |
| **Feature** | **AngularJS** | **Angular** |
| ***Architecture*** | Supports MVC design model | Uses components and directives |
| ***Language*** | Recommended Language: JavaScript | Recommended Language: TypeScript |
| ***Expression Syntax*** | Specific ng directive is required for the image/property and an event | Uses () to bind an event and [] for property binding |
| ***Mobile Support*** | Doesn’t provide any mobile support | Provides mobile support |
| ***Routing*** | $routeprovider.when() is used for routing configs | @RouteConfig{(…)} is used for routing config |
| ***Dependency Injection*** | Doesn’t supports the concept of Dependency Injection | Supports hierarchical Dependency Injection with a unidirectional tree-based change detection |
| ***Structure*** | Less manageable | Simplified structure and makes the development and maintenance of large applications easier |
| ***Speed*** | With two-way data binding development effort and time are reduced | Faster than AngularJS with upgraded features |
| ***Support*** | No support or new updates are provided anymore | Active support and frequent new updates are made |

**What do you mean by Cross-Platform?**

In computing, cross-platform software (also multi-platform software or platform-independent software) is computer software that is implemented on multiple computing platforms. ... For example, a cross-platform application may run on Microsoft Windows, Linux, and macOS.

**What are the advantages of using Angular?**

A few of the major advantages of using Angular framework are listed below:

* It supports two-way data-binding
* It follows MVC pattern architecture
* It supports static template and Angular template
* You can add a custom directive
* It also supports RESTfull services
* Validations are supported
* Client and server communication is facilitated
* Support for dependency injection
* Has strong features like Event Handlers, Animation, etc.

**What are Angular expressions or string interpolation?**

{{ expression }}

**How can you differentiate between Angular expressions and JavaScript expressions?**

|  |  |
| --- | --- |
| **Angular Expressions** | **JavaScript Expressions** |
| They can be written inside the HTML tags. | 2. They can’t be written inside the HTML tags. |
| They do not support conditionals, loops, and exceptions. | 3. They do support conditionals, loops, and exceptions. |
| They support filters. | They do not support filters. |

**Explain jQLite.**

jQlite is also known as jQuery lite is a subset of jQuery and contains all its features. It is packaged within Angular, by default. It helps Angular to manipulate the DOM in a way that is compatible cross-browser. jQLite basically implements only the most commonly needed functionality which results in having a small footprint.

**What are the different types of filters in Angular?**

Below are the various filters supported by Angular:

**currency**: Format a number to a currency format.

**date**: Format a date to a specified format.

**filter**: Select a subset of items from an array.

**json**: Format an object to a JSON string.

**limit**: To Limits an array/string, into a specified number of elements/characters.

**lowercase**: Format a string to lower case.

**number**: Format a number to a string.

**orderBy**: Orders an array by an expression.

**uppercase**: Format a string to upper case.

**Differentiate between one-way binding and two-way data binding.**

In One-Way data binding, the *View or the UI part does not update automatically whenever the data model changes.* You need to manually write custom code in order to update it every time the view changes.

Whereas, in Two-way data binding, the View or the UI part is updated implicitly as soon as the data model changes. It is a synchronization process, unlike One-way data binding.

**What do you understand by dirty checking in Angular?**

In Angular, the digest process is known as dirty checking. It scans the entire scope for changes. In other words, it compares all the new scope model values with the previous scope values. Since all the watched variables are contained in a single loop, any change/update in any of the variable leads to reassigning of rest of the watched variables present inside the DOM. A watched variable is in a single loop(digest cycle), any value change of any variable forces to reassign values of other watched variables in DOM

**Differentiate between DOM and BOM.**

|  |  |
| --- | --- |
| **DOM** | **BOM** |
| 1. Stands for Document Object Model | 1. Stands for Browser Object Model |
| 2. Represents the contents of a web page | 2. Works a level above web page and includes browser attributes |
| 3. All the Objects are arranged in a tree structure and the document can be manipulated & accessed via provided APIs only | 3. All global JavaScript objects, variables & functions become members of the window object implicitly |
| 4. Manipulates HTML documents | 4. Access and manipulate the browser window |
| 5. W3C Recommended standard specifications | 5. Each browser has its own implementation |

**How to perform animation in Angular?**

In order to perform animation in an Angular application, you need to include a special Angular library known as Animate Library and then refer to the ngAnimate module into your application or add the ngAnimate as a dependency inside your application module.

**List some tools for testing angular applications?**

* Karma
* Angular Mocks
* Mocha
* Browserify
* Sion

**What is a singleton pattern and where we can find it in Angular?**

Singleton pattern in Angular is a great pattern which restricts a class from being instantiate more than once. Singleton pattern in Angular is majorly implemented on dependency injection and in the services.

**What do you understand by REST in Angular?**

REST stands for REpresentational State Transfer. REST is an API (Application Programming Interface) style that works on the HTTP request. In this, the requested URL pinpoints the data that needs to be processed. Further ahead, an HTTP method then identifies the specific operation that needs to be performed on that requested data. Thus, the APIs which follows this approach are known as RESTful APIs.

**What do you understand by constants in Angular?**

In Angular, constants are similar to the services which are used to define the global data. Constants are declared using the keyword “constant”. They are created using constant dependency and can be injected anywhere in controller or services.

**What are Angular Global APIs?**

Angular Global API is a combination of global JavaScript functions for performing various common tasks like:

* Comparing objects
* Iterating objects
* Converting data

There are some common Angular Global API functions like:

* angular.lowercase: Converts a string to lowercase string.
* angular.uppercase: Converts a string to uppercase string.
* angular.isString: Returns true if the current reference is a string.
* Angular.isNumber: Returns true if the current reference is a number.

**In Angular, describe how will you set, get and clear cookies?**

For using cookies in Angular, you need to include a module called ngCookies angular-cookies.js.

**To set Cookies** – For setting the cookies in a key-value format ‘put’ method is used.

**cookie.set('nameOfCookie',"cookieValue");**

**To get Cookies** – For retrieving the cookies ‘get’ method is used.

**cookie.get(‘nameOfCookie’);**

**To clear Cookies** – For removing cookies ‘remove’ method is used.

**cookie.delete(‘nameOfCookie’);**

**Difference between “var” and “let” keyword**

**var:**

* Variables declared with “var” keyword has wide scope, accessible outside the block.
* var variables can be re-declared and updated.

var greeter = "hey hi";

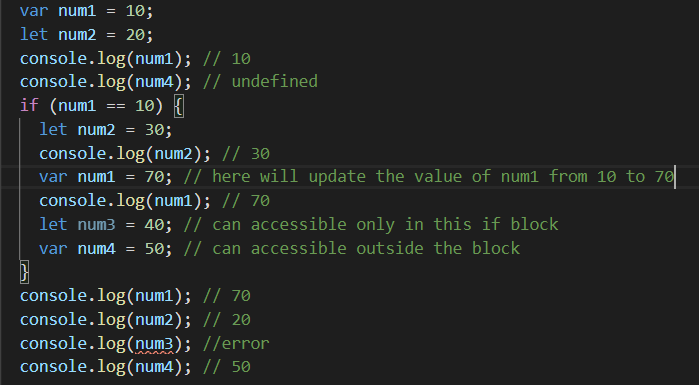
var greeter = "say Hello instead";

**Problem with var**

* It does not identify whether it is already declared or not.

**let:**

* It comes as an improvement to var declarations
* A variable declared in a block with **let** is only available for use within that block.
* If we declare a variable inside the block with **let** keyword and already have the same variable in parent block declared with **let** keyword then in that case both have different value in it.



**const:**

* Variable declare with const cannot be changed further.
* Must be assign value at the time of declaration.
* Object declare with const can be updated.
* We can make the object **read-only with the help of “Object.freeze()”** method.

//#region Const

    const cnt = 0;

    console.log(cnt);

    //cnt = 34; // cannot assign to 'cnt' because it is a constant.

    const person = {

      name: 'Joseph',

      age: 33

    }

    console.log(person);

    person.age = 34

    console.log(person);

    // Object.freeze will make person read-only

    const joseph = Object.freeze(person);

    //joseph.age = 35; // cannot assign to ‘age’ because it is a read-only property.

    console.log(joseph);

    person.age = 35; // will throw runtime error "Cannot assign to read only property 'age' of object '[object Object]'"

    console.log(person);

    //#endregion

**Decorators in Angular**

Here’s the list of decorators available in Angular:

1. @NgModule
2. @Component
3. @Injectable
4. @Directive
5. @Pipe
6. @Input
7. @Output
8. @HostBinding
9. @HostListener
10. [@ContentChild](https://medium.com/u/d220ff7175e9?source=post_page-----71bdf4ad6976--------------------------------)
11. @ContentChildren
12. @ViewChild
13. @ViewChildren