* **What Are The New Features Of Angular2?**

**Angular 2 is written entirely in Typescript and meets the ECMAScript 6 specification :**

* + **Component-Based-** Angular 2 is entirely component based. Controllers and $scope are no longer used. They have been replaced by components and directives.
  + **Directives-** The specification for directives is considerably simplified, although they are still subject to change. With the @Directive annotation, a directive can be declared.
  + **Dependency Injection**- Because of the improved dependency injection model in Angular2 there are more opportunities for component / object-based work.
  + **Use of TypeScript**-TypeScript is a typed super set of JavaScript which has been built and maintained by Microsoft and chosen by the AngularJS team for development. The presence of types makes the code written in TypeScript less prone to run-time errors. In recent times, the support for ES6 has been greatly improved and a few features from ES7 have been added as well.
  + **Generics-** TypeScript has generics which can be used in the frontend.
  + **Lambdas with TypeScript-** In TypeScript, lambdas are available.
  + **Forms and Validations-** Forms and validations are an important aspect of frontend development. Within Angular 2 the Form Builder and Control Group are defined.
* **What Is The Need Of Angular2?**

Angular 2 is not just a typical upgrade but a totally new development. The whole framework is rewritten from the ground. Angular 2 got rid of many things like $scope, controllers, DDO, jqLite, angular.module etc.

It uses components for almost everything. Imagine that even the whole app is now a component. Also it takes advantage of ES6 / TypeScript syntax. Developing Angular 2 apps in TypeScript has made it even more powerful.

Apart from that, many things have evolved and re-designed like the template engine and many more.

* **What Is Typescript ?**

TypeScript is a typed super set of JavaScript which has been built and maintained by Microsoft and chosen by the AngularJS team for development.

* **What Is The Need For Typescript In Angular2?**

**Understanding the need for TypeScript file in Angular2 applications :**JavaScript rules in Web development. Its the most popular language for developing web application UI. For may application developers having exposure in languages like Java and C#, creating the front end of a Web application in JavaScript is a very cumbersome process. For example if the user wants to create a class Employee in JavaScript. There is no class keyword in JavaScript so the code will be as follows-

<html>  
<head>  
</head>  
<body>  
<script>  
function Employee()  
{  
this.name="";  
this.id="";  
this.Validate=function()  
{  
alert("Validate");  
}  
}  
</script>  
</body>  
</html>  
**Same can be written using TypeScript as follows-**

class Employee  
{  
public name : string = "";  
public id : string = "";  
Validate()  
{  
alert("validate");  
}  
}

This Customer.ts will compile to the above JavaScript code.

So TypeScript provides the following advantages over JavaScript-

* + Structure the code- There were many different coding styles for JavaScript. This leads to unstructured code. With TypeScript we create structured code.
  + Use object-oriented programming paradigms and techniques-  There is lack of object-oriented design paradigms and techniques in JavaScript. This is not the case in TypeScript. It makes use of Objected Oriented features like Polymorphism, Inheritance etc.
  + Standard Coding guidelines- There is no Type checking in JavaScript. The code style needs to be defined. Hard to enforce style guide. TypeScript overcomes this issue with features like Code Analysis and Navigation, Documentation, Intellisense etc.
* **What Is Ecmascript ?**

ECMAScript is a subset of JavaScript. JavaScript is basically ECMAScript at its core but builds upon it. Languages such as ActionScript, JavaScript, JScript all use ECMAScript as its core. As a comparison, AS/JS/JScript are 3 different cars, but they all use the same engine... each of their exteriors is different though, and there have been several modifications done to each to make it unique. Angular2 supports ES6 and higher versions.

* **What Is @ngmodule?**

@NgModule is a decorator function. A decorator function allows users to mark something as Angular 2 thing (could be a module or component or something else) and it enables you to provide additional data that determines how this Angular 2 thing will be processed, instantiated and used at the runtime. So, whenever user writes @NgModule, it tells the Angular 2 module, what’s going to be included and used in and using this module.

* **What Is Traceur Compiler ?**

Traceur is a JavaScript.next-to-JavaScript-of-today compiler that allows you to use features from the future today. Traceur supports ES6 as well as some experimental ES.next features. Traceur's goal is to inform the design of new JavaScript features which are only valuable if they allow you to write better code.

* **What Is Component In Angularjs 2 ?**

In Angular, a Component is a special kind of directive that uses a simpler configuration which is suitable for a component-based application structure.

* **What Is @inputs In Angular 2?**

@Input allows you to pass data into your controller and templates through html and defining custom properties. This allows you to easily reuse components and have them display different values for each instance of the renderer.

* **What Is @outputs In Angular?**

Components push out events using a combination of an @Output and an EventEmitter. This allows a clean separation between reusable Components and application logic.

#### WHAT’S NEW IN ANGULAR 4?

Writing code is lots of cleaner and lesser.

It Improve the execution performance for Data binding and so on.

It has included Animations features.

In Angular 4, no need to apply observable methods because Angular analyses every page’s DOM and it is automatically modifies to page’s DOM.

It is also supported by Visual Studio, IntelliJ, And NET IDES and so on.

Migration is really very soft and cleaner.

#### WHAT IS THE FORROOT METHOD IN ANGULAR 4?

The forRoot is a static method and it’s very easy for developers to configure the modules and the best example is – **RouterModule.forRoot**.

The RouterModule also offers a **forChild**. It’s also a static method and use to configure the routes of lazy-loaded modules. The forRoot and forChild are the traditional names for methods that configure services in root.

Explain the life cycle hooks of Angular 2 application

Angular 2 component/directive has lifecycle events, managed by @angular/core. It creates the component, renders it, creates and renders its children, processes changes when its data-bound properties change, and then destroys it before removing its template from the DOM. Angular provides a set of lifecycle hooks(special events) which can be tapped into this lifecycle and perform operations when required. The constructor executes prior to all lifecycle events. Each interface has a single hook method prefixed with ng. For example, *ngOnint*interface has *Oninit*method that must be implemented in the component.

Some of the events are applicable for both component/directives while few are specific to components.

* **ngOnChanges**: Responds when angular sets its data-bound property which receives the current and previous object values.
* **ngOnInit**: Initializes the component/directive after first ngOnChange triggers. This is most frequently used method to retrieve the data for the template from a back-end service.
* **ngDoCheck**: Detect and act upon changes occuring outside Angular context. It is called when every change detection run.
* **ngOnDestroy**: Cleanup just before Angular destroys the directive/component. Unsubscribe observables and detach event handlers to avoid memory leaks.

**Component-specific hooks:**

* **ngAfterContentInit**: Component content has been initialized
* **ngAfterContentChecked**: After Angular checks the bindings of the external content that it projected     into its view.
* **ngAfterViewInit**: After Angular creates the component’s view.
* **ngAfterViewChecked**: After Angular checks the bindings of the component’s view.

What are the advantages of using Angular 2 over Angular 1?

1. Angular 2 is a platform not only a language:
2. Better Speed and Performance: *No $Scope in Angular 2, AOT*
3. Simpler Dependency Injection
4. Modular, cross platform
5. Benefits of ES6 and Typescript.
6. Flexible Routing with Lazy Loading Features
7. Easier to Learn

How routing works in Angular 2.

Routing is a mechanism which enables user to navigate between views/components. Angular 2 simplifies the routing and provide flexibility to configure and define at module level (Lazy loading).

The angular application has single instance of the Router service and whenever URL changes, corresponding Route is matched from the routing configuration array. On successful match, it applies redirects and the router builds a tree of ActivatedRoute objects and contains the current state of the router. Before redirection, the router will check whether new state is permitted by running guards ([CanActivate](https://blog.thoughtram.io/angular/2016/07/18/guards-in-angular-2.html)). Route Guards is simply an interface method that router runs to check the route authorization. After guard runs, it will resolve the route data and activate the router state by instantiation the required components into <router-outlet> </router-outlet>.