**Why we need Angular 2 when Angular 1.3 was already their**

1. **Performance**

* **When angular was created it was not for developers but for a designers who needed to build persistent HTML forms.**
* **Developers had picked up angular to create more and more complex applications.Angular 2 tried worked very hard to make incremental changes to the design allowing it continue to be relevant as the need of modern applications.**
* **But there are hard limits on the improvements that can be made which leads to performance problems . In order to fix those problems , new strategies are needed.**

1. **Mobile**

* **Phones and tablets are everywhere While angular can be used to build mobile apps . it was not designed with them in mind which leads to fundamental performance issues also issues like inability to cache precompile view and even touch support. Some can be fixed but other needs fundamental changes to fix.**

1. **Ease of Use**

* **Angular js is not easiest things to learn as you start building complex applications its very difficult to understand whats happening.**

1. **Changing Web**

* **Web has changed significantly , in coming next years the ES6 will be finalized . so we need support for modules ,classes, lambdas generators etc. this will transform the JS programming experience**

**Features and Benefits**

**Cross Platform**

1. **Progressive Web Apps**

* **Use modern web platform capabilities to deliver app like experience .**

1. **Desktop**

* **Create desktop installed apps across mac, windows and linux .**

**Speed and Performance**

1. **Code generation**

* **Angular turns your templates into a code that’s highly optimized for today JS virtual machines.**

1. **Universal**

* **Serve the first view of your application on nodeJS,.NET and PHP and other servers for near instant rendering in just HTML and CSS**

1. **Code Splitting**

* **Apps load quickly with new component router , it provides automatic code splitting so users only load code required to render the view they request.**

**Productivity**

1. **Templates**

* **Quickly create UI views with simple and powerful template syntax**

1. **Angular CLI**

* **Command line tools start building fast , add components and test and then instantly deploy**

1. **IDEs**

* **Get intelligent code completion, instant errors and other feedback in popular editors and IDEs**

**Full Development Story**

1. **Testing**

* **With Karma for unit tests you can know if you have broken things every time you save.**
* **Protractor makes your scenario tests run faster and in a stable manner**

1. **Animation**

* **Create high performance , animation timelines with very little code**

**Angular JS Architecture**

* **It’s a Framework for building client application is HTML and either JS or language like TS that compiles to JS.**
* **Write angular applications**

1. **Composing HTML templates with Angularized markup**
2. **Writing Component classes to manage those templates**
3. **Adding application logic in services**
4. **Boxing Components and Services in modules**
5. **Launch the app by bootstrapping the root module.**

Building Blocks of your Angular 2 application

**Modules**

* **Modules are great way to organize the application and extend it with capabilities from external libraries.**
* **An Angular module is a class decorated with @NgModule metadata**

1. **Metadata declares the component , directives and pipes belong to the module**
2. **Make some of those classes public so that other components templates can use them.**
3. **Import other modules with the components, directives and pipes needed by the components in this module.**
4. **Provide services at the application level that any application component can use.**

N**gModule Object Properties**

1. **Declarations : View classes that belong to this module. It has 3 types**

* **Components**
* **Directives**
* **Pipes**

1. **Exports : Subset of declaration that should be visible and usable in the component templates of other modules**
2. **Imports : Other modules whose exported classes are needed by component templates declared in this module.**
3. **Providers : Creators of services that this module contributes to the global collection of services. They become accessible to all parts of app**
4. **Bootstrap : the main application view called the root component that hosts all other app views. Only root module should set this bootstrap property**

**Components**

* **Component controls a patch of screen called view**
* **Components application logic needs to be define inside class. The class interacts with the view through an API of properties and methods.**

**Templates**

* **You define a components view with its companion template.**
* **Template tells angular how to render the component.**
* **It contains HTML elements with some angular related code like (click), \*ngFor,{{titile}}**

**Metadata**

* **Tells Angular how to process a class**
* **To do this you need to attach metadata using decorator**
* **We have to use @Component decorator which identifies the class immediately below it as a component class.**

**@Component Consists of**

1. **Selector : CSS selector that tells the angular to create and insert an instance of this instance of this component where it finds this selector tag.**

**For example**

**My-apps is a tag then angular inserts an instance of MyClass view between those tags.**

1. **templateUrl : module relative address of this components HTML template**
2. **providers :array of dependency injection providers for services that component requires.**

**Template, Metadata and Component together describe a view.**

**Data Binding**

* **Angular JS supports data binding , a mechanism for coordinating parts of a template with parts of a component.**
* **Binding mark up tells angular how to connect both sides.**
* **We have for forms of data binding**
* **It plays an imp role in communication between template and component or parent and child component.**

1. **Interpolation : displays the components property value.{{}}**
2. **Property binding : passes the value from property of parent to another property of child component []**
3. **Event Binding : Its use to bind the events (click)**
4. **Two Way : it combines the property and event binding**

**Directives**

* **Angular templates are dynamic , when angular renders them , it transforms the DOM according to the instructions given by directives.**
* **A directive is class with a @Directive decorator**
* **A component is a directive with a template**

**There are 2 types of directive**

1. **Structural**

* **It alters the layout by adding , removing and replacing elements in the DOM**

**For example .. ngIf, ngSwitch and ngFor**

1. **Attribute**

* **It alters the appearance of an existing element**

**Services**

* **It gives any value , function or feature that your application needs**
* **It’s a class with narrow well defined purpose**

**Example**

* **Logging Service**
* **Data service**
* **Tax calculator**

**Dependency Injection**

* **Its a way to supply a new instance of a class with fully formed dependencies it requires.**
* **DI provide new components with services the application needs or component needs.**
* **An injector maintains container of service instances that it has previously created . if requested services is not in the container then it makes one. When all requested services has been resolved and returned , angular can call the components constructor with those services as arguments . this is dependency injection.**

**Root Module**

* **Angular module class describes how the application parts fits together.**
* **Every application has atleast one module.**
* **Root module is use to bootstrap to launch your application**
* **Conventionally it is named as AppModule**

**It starts with imports after that @NgModule decorator is used . it tells angular how to compile and launch the application**

1. **Imports :**

* **Angular modules are way to consolidate features that belong together into discrete units**
* **Whenever application requires some module and its features , add a module to the imports array.**
* **Angular JS application needs to execute on the browser so we need browsermodule . So every such application includes Browser Module in its root.**

1. **Declarations :**

* **You must declare event component in NgModule class**
* **You tell angular which components belong to AppModule by using modules declarations array.**

1. **Bootstrap**

* **You launch the application by boot straping the root.**
* **Bootstrapping is a process create the component listed and inserts each one into the browser DOM.**
* **It sets up execution environment , digs root AppComponent . creates an instance of the component and inserts it withing the element tag identified by components selector**

**Component**

* **Angular applications are made up of components.**
* **Angular application is a tree of Angular components**
* **Component must belong to an Ngmodule in order to be usable by another component or application.**
* **To specify that a component is a member of an NgModule you should list it in the declarations field of that Ngmodule**
* **Component is the combination of and HTML component and component class that controls the portion of the screen.**
* **Every Component begins with an @Component**
* **@Component is decorator function that takes metadata object . metadata describes how the HTML template and Component class work together**

**Component Decorator**

* **It allows you to mark a class as an angular component and provide additional metadata that determines how the component should be processed , instantiated and used at runtime.**

**Template and TemplateURL**

* **Our Components often refer to external template .**
* **We identify those files with a URL in the templateURL.**
* **We must specify absolute path to load external templates.**
* **While template is used to specify the HTML CODE**

**Component Style**

* **For every Angular Component we can write HTML template as well as CSS styles that go with that template .**
* **We can set this style for component by using styles property of component metadata.**

**There are 2 ways of Setting style for component**

1. **Style**

* **It takes array of string that contain CSS code.**

1. **StyleUrl**

* **It takes the ref where style is stored.**

**Style and StyleUrl**

* **Its use to ref external stylesheet.**

**What is difference between Component Style and global style**

1. **Component Styles only apply within the template of that component**

**Binding**

* **You can display data by binding controls in an HTML template to properties of an Angular component.**

1. **Component Properties with interpolation**
2. **Component property with interpolation**

* **In this you can put the property name in the view template enclosed in double curly braces.**

1. **Binding to user input events**

* **To bind DOM event and surround the DOM event name in parenthesis and assign eventhandler to it.**
* **(click) identifies button click event and event handler which responds to the click event by calling onClickMe method.**

**Get user input from the $event object**

* **DOM events carry payload of information that may be useful for component.**
* **$ event object can be use to get the information about the event inside component.**
* **Passing an event object is like passing entire DOM event into method and the component has too much awareness about template details.**
* **So we can use template reference variable .**

**Get user input from a template ref variable**

* **This is one more way of getting the user data .**
* **Template ref variables provides direct access to an element from within the template.**
* **To declare variable you need # sign**
* **Here the template is self contained and doesnot bind to the component and component does nothing.**
* **In the prev example we had to write a code to bind hence it impacts the separation of concern .**
* **Its easier to get the input box with temp ref variable then to go thorough $event object.**

**Key Event Filtering**

* **The (keyup) event handler hears every keystroke. Some time only the enter key matters because it signals that the use has finished typing.**
* **You can accomplished it using Angulars keyup.enter pseudo event.**
* **The angular will call the event handler only when the user presses enter.**

1. **Property Binding**

* **Property binding and interpolation is one an the same . interpolation is a convenient alternative for property binding in many cases.**
* **Angular automatically translate interpolations into property binding**

1. **Class and Style Binding**
2. **Class binding**

* **We can add and remove CSS class names from an elements class attribute with class binding**
* **Class binding syntax resembles property binding**
* **Applying class binding helps us to add and remove class dynamically. So that depending on some condition you can enable or disable the class .**

1. **Style Binding**

* **We can set inline styles using Style Binding**
* **Its syntax is same as property binding**

1. **Two Way Data binding**

**-When developing entry forms we often want to display a data property and update data property when the user makes changes.**

**ngModel directives makes this easy.**

* **Forms Module is required if you want to work with ngModel.**

1. **Input and Output**

* **It is used to have communication between components where 2 or more components share information.**
* **Purpose is to pass the data from one component to another.**

**We can do same thing using @input and @output but adv is here**  we can **define the type and whether it is private or public:**

**Services**

* **Suppose we have 3 components which some code which is duplicated**

**Hence**

1. **Code duplication**
2. **Hard to maintain and Update**

**So duplicated code be kept in centralized manner and called as service**

**Uses of Service**

* **Provide , store and Interact with data**
* **Provide communication between components and classes**
* **Other business logic access from various places in your application.**

Angular Routing

* Its about Changing the state of your application , Loading different components depending on the URL user enters.
* Angular2 Job is to look at the URL and parsing it by identifying the different segments in the URL and then he try to find user component and userdata component.

Url Segments

Identify Routes

Parse Routes

http://my.app/user/12/data

Setting up Routes

* Our whole application should be aware of this route.
* We can register pur routes in AppModules.
* Angular has built in Router which is very powerful and this router ships in its own module.
* We have to create a separate file where we configure our routes and export it to application as a module.