What is Angular and Why Angular?
Angular (2/4/5/6/7) is not a programming language like Javascript or Typescript. Angular is a front-end or client-side Framework, which needs a programming language like Typescript (developed by Microsoft). Now, what does it mean when one says, "Angular is a client-side framework"? It means that it runs on the client-side or user's browser and not on a Web Server (where java/python/ruby/vb.net run). Of course, there is the Angular Universal which allows Angular to run on a server; but, primarily, angular is a client-side framework. Angular is a product developed and maintained by the techie giant Google and has adopted the SPA (Single Page Application) principles.

What's new in Angular 5?
Certain tools are optimized in the new version of Angular, let us see what the tools are:
- Angular 5 supports Typescript version 2.4
- Angular 5 supports RxJS 5.5 which has new features like Pipeable Operators
- A build tool to make the js bundles (files) lighter
- Ahead of Time (AOT) is updated to be on by default
- Events like ActivationStart and ActivationEnd are introduced in Router

List new features comes with Angular6
- Angular Elements - Angular Elements is a project that lets you wrap your Angular components as Web Components and embed them in a non-Angular application.
- Rendering Engine improvements:- increases in speed and decreases in application size.
- Tree-shakeable providers - a new, recommended, way to register a provider, directly inside the @Injectable() decorator, using the new providedInattribute
- RxJS 6 - Angular 6 now uses RxJS 6 internally, and requires you to update your application also. RxJS released a library called rxjs-compat, that allows you to bump RxJS to version 6.0 even if you, or one of the libraries you're using, is still using one of the "old" syntaxes.
- ElementRef<T> - in Angular 5.0 or older, is that the said ElementRef had its nativeElement property typed as any. In Angular 6.0, you can now type ElementRef more strictly.
- Animations - The polyfill web-animations-js is not necessary anymore for animations in Angular 6.0, except if you are using the AnimationBuilder.
- i18n - possibility to have "runtime i18n", without having to build the application once per locale.

Name the building blocks of Angular.
The Angular application is made using the following:
- Modules
- Component
- Template
- Directives
- Data Binding
- Services
- Dependency Injection
- Routing

What is Transpiling in Angular?
Transpiling is the process of converting the typescript into javascript (using Traceur, a JS compiler). Though typescript is used to write code in the Angular applications, the code is internally transpiled into javascript.

Which of the Angular life cycle component execution happens when a data-bound input value updates?
ngOnChanges is the life cycle hook that gets executed whenever a change happens to the data that was bound to an input.

Differentiate between Components and Directives in Angular.
Components break up the application into smaller parts; whereas, Directives add behavior to an existing DOM element.

What is the use of @Input and @Output?

When it comes to the communication of Angular Components, which are in Parent-Child Relationship; we use @Input in Child Component when we are passing data from Parent to Child Component and @Output is used in Child Component to receive an event from Child to Parent Component.

What is ng-content Directive?
The HTML elements like p (paragraph) or h1 (heading) have some content between the tags. For example, <p>this is a paragraph</p> and <h1>this is a heading</h1>. Now, similar to this, what if we want to have some custom text or content between the angular tags like <app-tax>some tax-related content</app-tax> This will not work the way it worked for HTML elements. Now, in such cases, the <ng-content> tag directive is used.

What does a router.navigate do?
When we want to route to a component we use router.navigate. Syntax: this.router.navigate(['/component_name']);

What is ViewEncapsulation?
ViewEncapsulation decides whether the styles defined in a component can affect the entire application or not. There are three ways to do this in Angular:

- Emulated: styles from other HTML spread to the component.
- Native: styles from other HTML do not spread to the component. Native will be deprecated in future releases, so instead use ShadowDom
- None: styles defined in a component are visible to all components.

What are Services in Angular and what command is used to create a service?
Services help us in not repeating the code. With the creation of services, we can use the same code from different components.

What is Dependency Injection in Angular?
When a component is dependent on another component the dependency is injected/provided during runtime.

What is Routing in Angular?
Routing helps a user in navigating to different pages using links.

How to handle Events in Angular?
Any activity (button click, mouse click, mouse hover, mouse move, etc) of a user on a frontend/web screen is termed as an event. Such events are passed from the view (.HTML) page to a typescript component (.ts).

What is a RouterOutlet?
RouterOutlet is a substitution for templates rendering the components. In other words, it represents or renders the components on a template at a particular location.

Explain the usage of {{}}?
The set of brackets {{}} when used with an HTML tag, represent data from a component. For example, on a HTML page which has <h1>{{variableName}}</h1>, here the 'variableName' is actually typescript (component) data representing its value on the template; i.e., HTML. This entire concept is called String Interpolation.

In how many ways the Data Binding can be done?
Data Binding happens between the HTML (template) and typescript (component). Data binding can be done in 3 ways:

(i) Property Binding
(ii) Event Binding
(iii) Two-Way Data Binding.

What is the sequence of Angular Lifecycle Hooks?
OnChange() - OnInit() - DoCheck() - AfterContentInit() - AfterContentChecked() - AfterViewInit() - AfterViewChecked() - OnDestroy().

What is the purpose of using package.json in the angular project?

With the existence of package.json, it will be easy to manage the dependencies of the project. If we are using typescript in the angular project then we can mention the typescript package and version of typescript in package.json.

How is SPA (Single Page Application) technology different from the traditional web technology?
In traditional web technology, the client requests for a web page (HTML/JSP/asp) and the server sends the resource (or HTML page), and the client again requests for another page and the server responds with another resource. The problem here is a lot of time is consumed in the requesting/responding or due to a lot of reloading. Whereas, in the SPA technology, we maintain only one page (index.HTML) even though the URL keeps on changing.

What is Component in Angular Terminology?
A web page in Angular has many components involved in it. A Component is basically a block in which the data can be displayed on HTML using some logic usually written in typescript.

What are ngModel and how do we represent it?
ngModel is a directive which can be applied on a text field. This a two-way data binding. ngModel is represented by [()]

What does a Subscribe method do in Angular?
It is a method which is subscribed to an observable. Whenever the subscribe method is called, an independent execution of the observable happens.

Differentiate between Observables and Promises.
Observables are lazy, which means nothing happens until a subscription is made. Whereas Promises are eager; which means as soon as a promise is created, the execution takes place. Observable is a stream in which passing of zero or more events is possible and the callback is called for each event. Whereas, promise handles a single event.

What is an AsyncPipe in Angular?
When an observable or promise returns something, we use a temporary property to hold the content. Later, we bind the same content to the template. With the usage of AsyncPipe, the promise or observable can be directly used in a template and a temporary property is not required.

Explain Authentication and Authorization.

- Authentication: The user login credentials are passed to an authenticate API (on the server). On the server side validation of the credentials happens and a JSON Web Token (JWT) is returned. JWT is a JSON object that has some information or attributes about the current user. Once the JWT is given to the client, the client or the user will be identified with that JWT.
- Authorization: After logging in successfully, the authenticated or genuine user does not have access to everything. The user is not authorized to access someone else's data, he/she is authorized to access some data.

What is AOT Compilation?
Every angular application gets compiled internally. The angular compiler takes javascript code, compiles it and produces javascript code again. Ahead-of-Time Compilation does not happen every time or for every user, as is the case with Just-In-Time (JIT) Compilation.

What is Redux?
It is a library which helps us maintain the state of the application. Redux is not required in applications that are simple with the simple data flow, it is used in Single Page Applications that have complex data flow.

What are the Pipes?
This feature is used to change the output on the template; something like changing the string into uppercase and displaying it on the template. It can also change Date format accordingly.

Differentiate between ng-Class and ng-Style.
In ng-Class, loading of CSS class is possible; whereas, in ng-Style we can set the CSS style.

Why Typescript with Angular?

Typescript is a superset of Javascript. Earlier, Javascript was the only client-side language supported by all browsers. But, the problem with Javascript is, it is not a pure Object Oriented Programming Language. The code written in JS without following patterns like Prototype Pattern becomes messy and finally leading to difficulties in maintainability and reusability. Instead of learning concepts (like patterns) to maintain code, programmers prefer to maintain the code in an OOP approach and is made available with a programming language like Typescript was thus developed by Microsoft in a way that it can work as Javascript and also offer what javascript cannot ie;

- pure OOPS as Typescript offers concepts like Generics, Interfaces and Types (a Static Typed Language) which makes it is easier to catch incorrect data types passing to variables.
- TS provides flexibility to programmers experienced in java, .net as it offers encapsulation through classes and interfaces.
- JS version ES5 offers features like Constructor Function, Dynamic Types, Prototypes. The next version of Javascript ie ES6 introduced a new feature like Class keyword but not supported by many browsers.
- TS offers Arrow Functions (=>) which is an ES6 feature not supported by many browsers directly but when used in TS, gets compiled into JS ES5 and runs in any browser.
- TS is not the only alternative to JS, we have CoffeeScript, Dart(Google).
- Finally, it is like, TS makes life easier when compared to JS.

What are the built-in validators in Angular?

- min
- max
- required
- requiredTrue
- email
- minLength
- maxLength
- pattern

List some Inbuilt Pipes available in Angular

Below is the list of few Pipes available in Angular Js

- DatePipe
- CurrencyPipe
- AsyncPipe
- DecimalPipe
- TitleCasePipe
- JsonPipe
- SlicePipe
- PercentPipe
- UpperCasePipe
- LowerCasePipe

What is difference between "declarations", "providers" and "import" in NgModule?

- imports make the exported declarations of other modules available in the current module
- declarations are to make directives (including components and pipes) from the current module available to other directives in the current module. Selectors of directives, components or pipes are only matched against the HTML if they are declared or imported.
- providers are to make services and values known to DI. They are added to the root scope and they are injected to other services or directives that have them as dependency.

A special case for providers is lazy loaded modules that get their own child injector. providers of a lazy loaded module are only provided to this lazy loaded module by default (not the whole application as it is with other modules).

What is AOT?

The Angular Ahead-of-Time compiler pre-compiles application components and their templates during the build process. Apps compiled with AOT launch faster for several reasons.

- Application components execute immediately, without client-side compilation.
- Templates are embedded as code within their components so there is no client-side request for template files.
- You don't download the Angular compiler, which is pretty big on its own.

- The compiler discards unused Angular directives that a tree-shaking tool can then exclude.

Explain the difference between "Constructor" and "ngOnInit"

- The Constructor is a default method of the class that is executed when the class is instantiated and ensures proper initialization of fields in the class and its subclasses.
- ngOnInit is a life cycle hook called by Angular to indicate that Angular is done creating the component. We have to import OnInit in order to use like this (actually implementing OnInit is not mandatory but considered good practice).

Why would you use renderer methods instead of using native element methods?

Angular is a platform, and the browser is just one option for where we can render our app. When we access the native element directly we are giving up on Angular's DOM abstraction and miss out on the opportunity to be able to execute also in none-DOM environments such as:

- native mobile,
- native desktop,
- web worker
- server side rendering.

The Renderer2 class is an abstraction provided by Angular in the form of a service that allows to manipulate elements of your app without having to touch the DOM directly. This is the recommended approach because it then makes it easier to develop apps that can be rendered in environments that don't have DOM access, like on the server, in a web worker or on native mobile.

What is Zone in Angular?

NgZone is a wrapper around Zone.js which is a library that creates a context around asynchronous functions in order to to make them trackable. Angular's change detection is heavily dependent on Zones.

Why would you use lazy loading modules in Angular app?

To load a feature module lazily we need to load it using loadChildren property in route configuration and that feature module must not be imported in application module. Lazy loading is useful when the application size is growing. In lazy loading, feature module will be loaded on demand and hence application start will be faster.

What are the lifecycle hooks for components and directives?

A component in Angular has a life-cycle, a number of different phases it goes through from birth to death. We can hook into those different phases to get some pretty fine grained control of our application.

- **constructor** This is invoked when Angular creates a component or directive by calling new on the class.
- **ngOnChanges** Invoked every time there is a change in one of th input properties of the component.
- **ngOnInit** Invoked when given component has been initialized. This hook is only called once after the first ngOnChanges
- **ngDoCheck** Invoked when the change detector of the given component is invoked. It allows us to implement our own change detection algorithm for the given component.
- **ngOnDestroy** This method will be invoked just before Angular destroys the component. Use this hook to unsubscribe observables and detach event handlers to avoid memory leaks.

These hooks are only called for components and not directives.

- **ngAfterContentInit** Invoked after Angular performs any content projection into the components view (see the previous lecture on Content Projection for more info).
- **ngAfterContentChecked** Invoked each time the content of the given component has been checked by the change detection mechanism of Angular.
- **ngAfterViewInit** Invoked when the component's view has been fully initialized.
- **ngAfterViewChecked** Invoked each time the view of the given component has been checked by the change detection mechanism of Angular.

What is ngUpgrage?

NgUpgrade is a library put together by the Angular team, which we can use in our applications to mix and match AngularJS and Angular components and bridge the AngularJS and Angular dependency injection systems.

What does a just-in-time (JIT) compiler do (in general)?
A JIT compiler runs after the program has started and compiles the code (usually bytecode or some kind of VM instructions) on the fly (or just-in-time, as it's called) into a form that's usually faster, typically the host CPU's native instruction set. A JIT has access to dynamic runtime information whereas a standard compiler doesn't and can make better optimizations like in lining functions that are used frequently.

This is in contrast to a traditional compiler that compiles all the code to machine language before the program is first run.

How to detect a route change in Angular?
In Angular you can subscribe (Rx event) to a Router instance. So you can do things like:

```
class MyClass {
  constructor(private router: Router) {
    router.subscribe((val) => /*whatever*/ )
  }
}
```

What is Reactive programming and how does it relate to Angular?
Reactive programming is programming with asynchronous data streams. RxJs stands for Reactive Extensions for Javascript, and it's an implementation of Observables for Javascript. An Observable is like a Stream (in many languages) and allows to pass zero or more events where the callback is called for each event. Angular currently uses RxJs Observables in two different ways:

- as an internal implementation mechanism, to implement some of its core logic like EventEmitter
- as part of its public API, namely in Forms and the HTTP module

You do not need to know Reactive Programming or RxJS in order to build even the most complex applications with Angular. It can however make some types of applications easier to architect.

Name some security best practices in Angular

- To systematically block XSS bugs, Angular treats all values as untrusted by default (sanitation)
- Angular templates are the same as executable code: HTML, attributes, and binding expressions (but not the values bound) in templates are trusted to be safe. To prevent these vulnerabilities, use the offline template compiler, also known as template injection.
- Avoid interacting with the DOM directly and instead use Angular templates where possible.
- Injecting template code into an Angular application is the same as injecting executable code into the application. So, validate all data on server-side code and escape appropriately to prevent XSS vulnerabilities on the server.
- Angular HttpClient provides built-in support to prevent XSRF attacks on the client side.
- Servers can prevent the XSSI attack by prefixing all JSON responses to make them non-executable, by convention, using the well-known string ")]}',\n". Angular's HttpClient library recognizes this convention and automatically strips the string ")]}',\n" from all responses before further parsing.

Just-in-Time (JiT) vs Ahead-of-Time (AoT) compilation. Explain the difference.

JIT - Compile TypeScript just in time for executing it:

- Compiled in the browser.
- Each file compiled separately.
- No need to build after changing your code and before reloading the browser page.
- Suitable for local development.

AOT - Compile TypeScript during build phase:

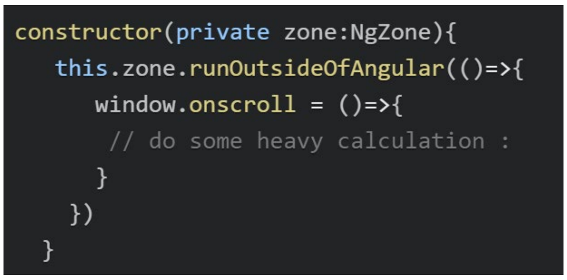
- Compiled by the machine itself, via the command line (Faster).
- All code compiled together, inlining HTML/CSS in the scripts.
- No need to deploy the compiler (Half of Angular size).
- More secure, original source not disclosed.
- Suitable for production builds.

Do you know how you can run angularJS and angular side by side?
In order to run both frameworks side-by-side and make components interoperable, the Angular projects comes with a module ngUpgrade. The module basically acts as an adapter facade, so we don’t really feel that there are two frameworks running side-by-side.

For this to work, four things need to interoperate:
- Dependency Injection - Exposing Angular services into Angular 1.x components and vice-versa.
- Component Nesting - Angular 1 directives can be used in Angular 2.x components and Angular 2.x components can used Angular 1 directives
- Content Projection / Transclusion - Angular 1 components transclude Angular 2.x components and Angular 2.x component project Angular 1 directives
- Change Detection - Angular 1 scope digest and change detectors in Angular >= 2.x are interleaved

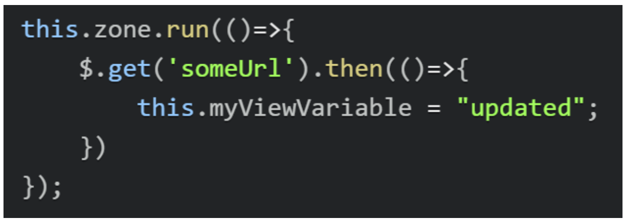
Here’s what a typical upgrade process would look like:
- Include Angular and upgrade module
- Pick component to upgrade and change its controller and template Angular 2.x syntax (this is now an Angular 2.x component)
- Downgrade Angular 2.x component to make it run in Angular 1.x app
- Pick a service to upgrade, this usually requires little amount of change (especially if we’re on ES2015)
- Repeat step 2 and 3 (and 4)
- Replace Angular 1 bootstrap with Angular 2.x bootstrap

Could you provide some particular examples of using ngZone?
There would be a lot of cases when you want to use NgZone, I can name two :
- When you want something to run outside of Angular's change detection. Lets say we want to do some calculation when user is scrolling and don't want you to run change detection, in this case, you'd use NgZone:



```
constructor(private zone:NgZone){
   this.zone.runOutsideOfAngular(()=>{
      window.onscroll = ()=>{
        // do some heavy calculation :
      }
    })
  }
```

- The exact opposite of above, where you have a function that is somehow outside of Angular's zone and you want it to be inside, like when a third party library is doing some stuff for you and you want it to be bound to your Angular cycle.



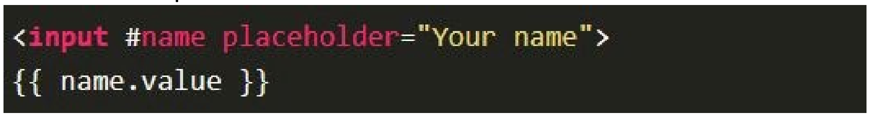
```
this.zone.run(()=>{
    $.get('someUrl').then(()=>{
        this.myViewVariable = "updated";
    })
});
```

When to use query parameters versus matrix parameters?
The differences between Matrix parameters and Query Parameters are much more than just convention.
The main differences are:
- urls with query params won't have their response cached by intermediaries/proxies (at present)
- matrix parameters may appear anywhere in path
- calculating the relative uri is different
- query params are generally abused to add new verbs instead of using existing methods on resources
- matrix parameters are not resources, they are aspects that help reference a resource in an information space that is difficult to represent within a hierarchy

Explain different types of subjects in Rxjs.

- **Subject** - a subscriber will only get published values that were emitted after the subscription. For example, with component-to-component communication. Say you have a component that publishes events for other components on a button click. You can use a service with a subject to communicate.
- **BehaviorSubject** - the last values is cached. An subscriber will get the latest value upon initial subscription. The semantics for this subject is to represent a value that changes over time. For example a logged in user. The initial user might be an anonymous user. But once a user logs in, then the new value is the authenticated user state.
- **ReplaySubject** - it can cache up to a specified number of emissions. Any subscribers will get all the cached values upon subscription.

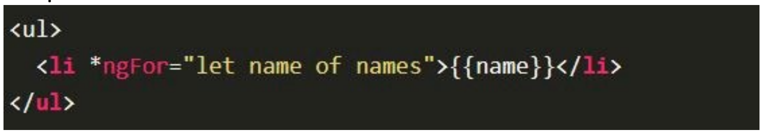
What is Template reference variables?
A template reference variable (#var) is a reference to a DOM element within a template. We use hash symbol (#) to declare a reference variable in a template.



```
<input #name placeholder="Your name">
{{ name.value }}
```

In the above code the #name declares a variable on the input element. Here the name refers to the input element. Now we can access any property of the inputDOM, using this reference variable. For example, we can get the value of the inputelement as name.value and the value of the placeholder property by name.placeholder anywhere in the template.
Finally, a Template reference variable refers to its attached element, component or directive. It can be accessed anywhere in the entire template. We can also use ref- instead of #. Thus, we can also write the above code as ref-name.

What are structural directives?
Structural directives are responsible for HTML layout. They shape or reshape the DOM's structure, typically by adding, removing, or manipulating elements. Structural directives are easy to recognize. An asterisk (*) precedes the directive attribute name as in this example.



```
<ul>
  <li *ngFor="let name of names">{{name}}</li>
</ul>
```

The ngFor directive iterates over the component's names array and renders an instance of this template for each name in that array.
Some of the other structural directives in Angular are ngIf and ngSwitch.

What are all the types of Directives?
There are three types of directives in Angular.

- **Structural directives** change the DOM layout by adding and removing DOM elements. For example, *ngIf and *ngFor
- **Attribute directives** change the appearance or behavior of an element. For example, *ngStyle and *ngClass
- **Components** are basically directives with a template.

What is Pure and Impure Pipes?

- **Pure pipes** are stateless that flow input date without remembering anything or causing detectable side-effects. Pipes are pure by default; hence most pipes are pure. We can make a pipe impure by setting its pure flag to false. Angular executes a pure pipe only when it detects a pure change to the input value. A pure change is either a change to a primitive input value or a changed object reference.
- **Impure pipes** are those which can manage the state of the data they transform. A pipe that creates an HTTP request, stores the response and displays the output, is an impure or stateful pipe. Stateful Pipes should be used cautiously. Angular provides AsyncPipe, which is stateful.