Angular 经典面试题

1. 说说 angular 生命周期钩子?

- ngOnChanges: 当组件数据绑定的输入属性发生变化是触发，该方法接收一个 SimpleChanges 对象，包括当前值和上一个属性值。首次调用一定发生在 ngOnInit 前，值得注意的是该方法仅限于对象的引用发生变化时才会触发。
- ngOninit: 初始化指令或组件，在 angular 第一次显示展示组件的绑定属性后调用，该方法只会调用一次
- ngDocheck: 检测
  - ngAfterContentInit 次当把内容投影进组件之后调用，第一次调用 ngDocheck()之后调用，只调用一次，只适用于组件
  - ngAfterContentChecked 每次完成被投影组件内容的变更检测之后调用，只适用于组件
  - ngAfterViewInit 在 angular 初始化组件及其子组件的视图之后调用，只调用一次,只适用于组件
  - ngAfterViewChecked 每次做完组件视图和子视图的变更检测之后调用，只适用于组件
- ngOnDestroy: 在 angular 每次销毁组件或指令之前调用，通常用于移除事件监听，退订可观察对象

2. 说说 angular 指令分类?

- 组件：用于构建 UI 组件，继承于 Directive 类。
- 属性指令：用于改变组件的外观或行为
  - ngClass
  - ngStyle
- 结构指令：用于动态添加或删除 DOM 元素来改变 DOM 布局
  - ngIf
  - ngFor
  - ngSwitch

3. angular 中父子组件之间的数据传递?

- 父到子组件之间的数据传递：父组件模板中引用子组件。

```
<child-item [name] = "fatherItemName" > </child-item>
```

  - 其中”fatherItemName” 为父组件中的属性，[name] 为子组件的输入，在子组件中使用 @Input() name 来接受父组件传递的值
  - @Input() 父组件向子组件传递数据和传递方法(子组件中使用)

- 子到父组件之间的数据传递
  - 事件传值
  - 父组件通过局部变量获取子组件的引用，`@output` 子组件传值给父组件 （事件传递的方式）（子组件中使用）
  - 使用`@ViewChild` 获取子组件的引用， 使用`@ViewChild` 父组件通过局部变量获取子组件的引用，主动获取子组件的数据和方法（父组件中使用）

4. 说说 angular 双向数据绑定的原理?

- 双向绑定机制维护了页面（View）与数据（Data）的一致性。如今，MVVM 已经是前段流行框架必不可少的一部分。
- 双向绑定原理:
  - `data=>view:` 数据绑定，模板语法是 []
  - `view=>data:` 事件绑定，模板语法是 ()
  - `Angular` 其实并没有一个双向绑定的实现，他的双向绑定就是数据绑定+事件绑定，模板语法是 [()]

5. 说说 AngularJS 与 Angular2 变化检测的区别?

- 在 angularJS1 中脏检查循环主要发生在:
  - 少的调用 `angular` 封装的 `DOM` 事件(click)
  - 调用 `http` 服务(`$http`)
  - 调用定时器(`$timeout,$interval`)
  - 手动调用`$scope.$apply` 或`$scope.digest`
  - 实现的过程则是，遍历监视器队列中的每一个监控元素，如果监控元素发生了变化，则调用其注册的监听函数，在监听函数中会改变相应视图元素
- angularJS2:
  - 变化监测的时机
    - 注用户行为操作，即 DOM 事件
    - 前后端的数据交互，使用 XHQ 对象
    - 各类的定时任务
  - 变动通知机制
    - 知 angularJS2 引入了 NgZone 服务，这个服务获取到了整个应用的上下文，能够对相关异步事件发生、完成、或异常等进行捕获。
    - 当有异步事件触发导致数据变化时，这些异步事件会被 NgZone 捕获并触发 onUnstable 自定义事件，在该自定义事件处理函数中通知 angular 去执行变化监测，如当鼠标经过的 mousemove 事件发生时，它将触发变化监测

- 变化监测的响应处理
  - 触发变化监测后，从根组件开始，依次触发各个子组件的内部的变化监测器去完成变化对比工作。因此任何一个组件中的数据变化都会导致整个组件树的变化监测

6. angular 双组件和指令的区别?

- omponent 使用注解@Component 修饰，directive 使用注解@Directive 修饰
- omponent 是组件化思想，基于组件创建应用，把应用划分成细小的可重复利用的组件，而 directive 用来在已经存在的 DOM 元素上实现一些行为
- omponent 是可重复使用的组件，directive 是可重复使用的行为
- omponent 可创建一个 即 template 或 templateUrl，而 directive 没有。

7. 说说 angular 有几种数据绑定方式?

- 属性绑定 [ ]
- 事件绑定（）
- 双向数据绑定 [()]

8. 单页面应用和传统的 web 技术有什么不同?

- 在传统的 web 技术中，客户端请求一个 web 页面(HTML/JSP/asp)，服务器返回资源(或 HTML 页面)，客户端再次请求另一个页面，服务器用另一个资源响应。问题就在于请求/响应中消耗了大量时间，或者是重新加载使用了大量时间。而在 SPA 技术中，即使 URL 不断变化，我们也只维护一个页面(index.HTML)。

9. 说说 Observables 和 Promises 的区别?

- Observables 是惰性的，意思是在 subsciption 之前什么都不会发生
- Promise 是 eager 的，意思是一旦创建，就会执行
- Observable 是一个 stream，可以传递 0,1，或者多个事件，并且为每个事件回
- Promise 只处理一个事件。
- Observable 可取消
- Promise 不可取消。

10. Authentication and Authorization 的区别?

- Authentication (认证) ：用户登录凭据传递给(服务器上的)认证 API。在服务器端验证凭据并返回 JSON Web Token(JWT)。JWT 是一个 JSON 对象，它有关于当前用户的一些信息或属性。一旦 JWT 返回给给客户端，客户端或用户将被该 JWT 所标记

- Authorization（授权）：登录成功后，经过身份验证或真正的用户不能访问所有内容。用户未被授权访问其他人的数据，他/她被授权访问某些数据

11. 什么是 Redux?

- 它是一个帮助我们维护应用程序状态的库。简单的数据流应用程序不需要 Redux，它用于具有复杂数据流的单页应用程序。

12. ng-Class 和 ng-Style 的区别?

- ng-Class: 加载 css 类
- ng-Style: 设置 css 样式

13. component 和 module 的区别?

- component:控制视图（html）.组件之间以及组件和 service 之间互相交互给 app 提供功能
- module: 是包括一个或多个组件，module 不会控制视图（HTML）。module 声明了哪些模块可以被其他模块使用，依赖注入了哪些类，以及启动的 component，模块来管理组件，使 app 实现模块化。

14. angular 中怎样在组件中选择一个元素?

- 在组件的 constructor 中引入 ElementRef 来操作 DOM 元素

```
constructor(myElement: ElementRef) { ... }
```

15. Constructor 和 ngOnInit 的本质区别?

- Constructor 在 ES6 中 constructor 表示构造函数，使用在 class 中。来初始化操作。当类被初始化之后，构造函数会被调用
- ngOnInit 是 angular 中 OnInit 钩子的实现，用来初始化组件，在 angular 第一次显示数据绑定和设置指令、组件的输入属性之后，初始化指令、组件
- 所以从 angular 的生命周期看，constructor 是执行在先的
- 既然 ngOnchanges 是输入属性发生变化的时候调用，并且 ngOnInit 是在 ngOnchanges 执行之后才调用，而 constructor 是在组件实例化的时候就调用了，也就是说，在 constructor 中是取不到输入属性的值的
- Constructor 中不适合进行任何和组件通信类似的复杂操作，一般在 constructor 中进行一些简单的初始化操作，比如依赖注入、变量初始化等。
- ngOnInit 可以用来初始化组件之间通信的，如异步请求等

16. 在 angular 中你对服务有什么了解？？

- 在 `angularjs` 中的服务用于组织和共享应用程序中的代码。 这些是在依赖注入的帮助下连接在一起的合适对象。 angularjs 服务被懒惰地实例化。 仅当应用程序组件依赖于它时，该服务才由 angularjs 实例化。

17. 在 angular 中指令和组件之间的区别?

- 指令用于向现有元素添加行为。 而组件用于创建具有附加行为的组件。
- `@directive` 用于创建指令。 而`@component` 用于创建组件。
- 指令用于将不同的行为附加到现有 `DOM` 元素。在组件的帮助下，可以将应用程序分解为更小的组件。
- 指令用于创建可重用行为。 然而，组件用于创建可重用组件。
- 指令不需要视图。然而，组件需要通过@view 查看。

18. 使用一个例子解释说明 ngif 指令?

- `ngIf` 是一个内置的模板指令，用于添加或删除 DOM 的某些部分。 此添加或删除取决于表达式是 `true` 是 `false`。

```
<ul *ngFor="let person of people" *ngIf="person.age < 30">
    <li>{{person.name}}</li>
</ul>
```

19. 请使用一个例子解释说明 ngFor 指令?

- `ngFor` 指令为给定迭代器的每个元素实例化一个模板。 `ngFor` 指令的不同局部变量可用于迭代。 `ngFor` 指令甚至可以与 HTML 元素一起使用。 它还在 DOM 中执行各种更改。 在 `ngFor` 指令的帮助下，可以将多个导出值别名化为局部变量。 它允许在 HTML 模板中构建数据表示列表。

```
<tr *ngFor="hero of heroes">
    <td>({hero.name})</td>
</tr>
```

20. angular 双向数据绑定的原理?

- 要在 Angular 4 中包含外部 CSS，请打开`.angular-cli.json` 并在 `styles` 数组中添加 css 文件路径。