https://github.com/yinminghao19/angular4helpChm （明昊老师）

ES 5 6 ：

1.let与const

(1)let: 如下图 let所声明的变量，只在let命令所在的代码块内有效，

也就是局部变量，如果把下图注释部分放开就会报错

(2)const: const声明的变量是常量，不能被修改

2.浅拷贝 深拷贝

如果你拷贝的对象中引用了某个外部的内容（比如分配在堆上的数据），

那么在拷贝这个对象的时候，让新旧两个对象指向同一个外部的内容，

就是浅拷贝；

如果在拷贝这个对象的时候为新对象制作了外部对象的独立拷贝，就是深拷贝

浅复制：仅仅是指向被复制的内存地址，如果原地址发生改变，那么浅复制出来的对象也会相应的改变。

深复制：在计算机中开辟一块新的内存地址用于存放复制的对象。

深：

function deepClone(obj): any

{

const newObj = obj instanceof Array ? [] : {};

if (typeof obj !== 'object')

{

return obj;

}else{

// tslint:disable-next-line:forin

for (const i in obj)

{

newObj[i] = typeof obj[i] === 'object' ? deepClone(obj[i]) : obj[i];

}

}

return newObj;

}

const a = [1, 2, 4, 6, 'a', '12', [1, 2]];

const b = deepClone(a);

a[3] = 7;

a[6] = [1, 1];

console.log(a);

console.log(b);

angular.foreach遍历数组，for.in遍历对象(虽然也可以遍历数组但不建议)

模板内的局部变量 <input type="text" formControlNmae = username #username >#username==>模板内局部变量

数据绑定：



下面的例子可以解释：

<div>

<li>{{hero.name}}</li>

<app-hero-detail [hero]="selectedHero">

</app-hero-detail>

<li (click)="selectHero(hero)"></li>

</div>

第一个<li>中，{{hero.name}}，这是一个插值绑定，就是我们在控制器定义的变量，数据能直接显示到前台。

第二个[hero]属性绑定把父组件HeroListComponent的selectedHero的值传到子组件HeroDetailComponent的hero属性中。

第三个事件绑定给click绑定事件。

双向绑定：这个 <input [(ngModel)]="myData"> 注意这里就需要引入FormsModule模块了

安全取值 {{currentRace?.name}} ?. ==》安全操作符，在数据为null 或者不存在时不会报错

| **名称** | **类型** | **作用** |
| --- | --- | --- |
| selector | string | 自定义组件的标签，用于匹配元素。 |
| inputs | string[] | 指定组件的输入属性。 |
| outputs | string[] | 指定组件的输出属性 |
| host | {[key:string],string} | 指定指令或组件的事件，动作和属性等 |
| providers | any[] | 指定该组件及其所有子组件（含ContentChildren）可用的服务依赖注入 |
| exportAs | string | 给指令分配一个变量，使其可以在模板中使用 |
| moduleId | string | 包含该组件模块的id，他被用于解析模板和样式的相对路径 |
| queries | {[key:string],any} | 设置需要被注入到组件的查询 |
| viewProviders | any[] | 指定该组件及其所有子组件可用的服务 |
| changeDetection | ChangeDetectionStrategy | 指定使用的变化监测策略 |
| templateUrl | string | 模板路径 |
| styleUrls | string[] | 样式路径 |
| template | string | 内联模板 |
| styles | string[] | 内联样式 |
| animations | AnimationEntryMetadata[] | angular动画 |
| encapsulation | ViewEncapsulation | 设置组件的试图包装选项 |
| interpolation | [string,string] | 设置自定义插值标记，默认是{{}} |