1, 通过ES6方式给 <ul> <li> </li> <li> </li> <li> </li> <li> </li> </ul> 中li绑定click事件，要求点击li输出该li的序号（0,1,2,3……）；

<**script**>  
 **window**.onload = **function** () {  
 **var** root = **document**.getElementById(**"test\_ul\_id"**);  
 **var** childrens = root.**children**;  
 **if** (childrens && childrens.**length** > 0){  
 **for**(**let** index = 0; index < childrens.**length**; index++){  
 childrens[index].**onclick** = **function** () {  
 **console**.log(index)  
 }  
 }  
 }  
 }  
</**script**>  
<**body**>  
<**div**>  
 <**ul id="test\_ul\_id"**>  
 <**li**>Java</**li**>  
 <**li**>Spring</**li**>  
 <**li**>Spring MVC</**li**>  
 <**li**>Mysql</**li**>  
 <**li**>Mybatis</**li**>  
 </**ul**>  
</**div**>

2, 理解let实现原理；

Let 弥补了ES6 之前js 没有块级作用域的遗憾。Let 关键字避免了变量提示，同时拥有暂时性死区（TDZ）

3，简单描述let和const实现不同；

用const声明的变量必须初始化，且const 变量指向的地址无法改变，这一点和java的final 一样，只是指针无法改变，对象还是可以改变的

4，简单说明一下ES6模块化

通过export 、import 两个关键字实现。支持一个模块（js文件）中的部分内容导入导出,以js对象的形式。

1. 熟悉ES6语法糖（构造函数、属性、静态方法等等），通过自定义一个类实现一个Demo；
2. <**script**>  
    **class** Student{  
    **constructor**(age,name){  
    **this**.age = age  
    **this**.**name** =name *// name 属性时关键字* }  
    sayHello(){  
    **console**.log(**'hello, I am '** + **this**.**name**)  
    }  
    **get** age(){  
    **console**.log(**"get age"**)  
   *// return this.age //死循环* }  
    **set** age(val){  
   *// this.age = val // 死循环* **console**.log(**"set age"** + val)  
    }  
     
    **static** *running*(){  
    **console**.log(**'running'**)  
    }  
    }  
   *// Student() 只能通过new 关键字构造* **var *stu*** = **new** Student(10,**"tom"**)  
    ***stu***.sayHello()  
    ***stu***.age = 10  
    **console**.log(***stu***.age)  
   *// stu.running() // 报错，实例无法调用静态方法* **debugger** Student.*running*() *//静态方法不在原型链上*</**script**>