1. 在vs code中按Ctrl+/，出现的不是注释，而是{% comment %}？

答: 禁用插件中的django template

2. 在vs code中用boot没有提示？

答: 安装bootstrap3 snippets 或bootstrap4 snippets插件

3. 如果调用cook(。。。)函数，但是没有用变量wan接，是不是就没有执行return？

答：函数只要调用，都会完整执行。return也是语句，当然也会被执行。这和外部用不用变量接无关。

就好像现实中，家人煮好面条了，问你吃还是不吃。你可以选择不吃，也可以选择拿碗吃。但是，不会因为你不吃面，家里人煮面的步骤就会少一步。这是不可能的。

必须把晦涩的程序和现实结合起来。因为程序也是人根据现实中的经验创造的出来的。都离不开现实。

4. vscode中，按tab键不会自动补全？

答：可以点菜单"编辑"->Emmet展开缩写，效果是一样的

也可能是右下角不是html的问题。

5. var fun=100; 是将地址值改为100，所以找不到函数才报错吗？

答？无论是普通的number数值，还是一个对象的地址值，对程序来说，都是无差别的一个数据。只要是一个数据就可以保存在变量中

fun=100只是用一个整数100数据，替换了原来保存在变量中的地址数据。

报错，是因为，语法规定，如果fun中保存的是一个数字，则数字后是不能跟()，像函数一样调用的。

fun=100替换后，函数对象的地址是不会改变的，一个对象的地址，一旦创建，就不会改变了。就像现实中的手机号。

6. 为什么声明两个同名函数，函数定义不同，也会覆盖？

答：js中，函数名其实就是一个普通的变量。函数体其实是一个对象。函数名中保存着函数对象的地址值。函数名变量通过函数对象的地址值，找到函数对象。就像现实中打电话找人一样。

声明两次同名函数，意味着，先创建一个函数名变量，然后将第一个函数对象的地址保存到函数名变量中。此时，函数名变量暂时通过第一个函数对象的地址值指向第一个函数对象。

第二次声明同名函数时，不会重复创建函数名变量，而是将新函数对象的地址，覆盖到函数名变量中现有旧函数对象的地址。就像手机通讯录里，将一条通讯录中的地址换成另一个人的手机号。从此，函数名变量，就和旧函数再无关系。而是用新函数对象的地址，引用着新函数对象。

7. var a=10

function foo(){

console.log(a); //10

a=20;

}

foo();  
 console.log(a) //20

答：10和20, 因为程序中，自始至终只有一个变量a，无论函数内，还是函数外，用的都是一个全局变量a。对a做任何修改，都是修改的一处。都会影响后续操作a的值。

如果换成:

var a=10

function foo(){

console.log(a); //undefined

var a=20;

}

foo();  
 console.log(a) //10

结果变为undefined和 10

首先：其实程序中不止一台hoist起重机。其实每个函数内的顶部，都各自带一个起重机。

在调用函数时，都会先再函数内声明提前，再执行。上面程序会变为:

var a=10

function foo(){

//hoist

// |

// |

// J

var a; //undefined

console.log(a); //undefined

a=20; //这里因为函数中有局部变量a了，优先使用局部变量，所以，不会影响全局。且局部变量，在函数调用后，就释放了。

}

foo();  
 console.log(a) //10 //所以，除了函数，全局a还是10

8. console.log(fun()); //2 undefined，为啥输出带一个undefined

答：因为函数fun(){}内部没有return，就没有返回值，返回到函数外部。所以，凡是没有return的函数，都不能直接打印执行结果，打印输出的都是undefined。只有函数内部有return的函数，才能直接输出函数的调用结果。

9. var fun=100;  
 console.log(fun()) 结果为什么不是100？

答：程序有严格的语法规定。如果想使用变量中的值，就不能加()。如果想调用变量引用的函数，执行一段任务逻辑，就必须加()。

所以写程序前必须先要知道变量中存的是什么，然后再想想打算怎么用这个变量里的东西。

如果变量中引用的是一个函数，想调用函数，就必须fun()

如果变量中保存的是一个值，想输出值，就不能加()。

这里，如果想输出变量的值，就应该console.log(fun)

如果想输出函数调用后执行的结果，就要console.log(fun())

执行时，先调用fun()，如果有返回值，会放到console.log()中输出。

如果没有返回值，console.log()没有接到结果，就输出undefined。

10. 函数和方法的差别?

答: 函数和方法，本质都是function，也就是都是函数

严格来说:

不属于任何对象的独立的function称为函数

比如: parseInt() parseFloat() isNaN()

保存在对象内的函数，称为方法，也称为method

比如: arr.sort() reg.test() str.toUpperCase()

数组的下标和length，不是方法，也不是函数，而是属性。属性不做事儿，只存值。

11. ajax四步能不能包装在匿名函数中？

答：凡是可能反复使用的函数，就不能用匿名函数！

比如ajax四步，将来一定会封装在一个叫ajax的有名字的函数中被反复调用。

12. (function(){

function add(){

...

}

})()  
 </script>  
 <button onclick="add()">

为什么绑定不上事件处理函数？

答：function add(){}已经不是全局函数了，而是局部函数。

不能直接用onclick=""方式找到。

必须在匿名函数内，通过DOM查找到元素，动态绑定。

今后，不会用onclick="js语句"这种方式。可以忘掉了。

12. do{

var reg=/明明|静静/g;

var arr=reg.exec(str);

console.log(arr)

}while(...);

永远只能找第一个。

但是:

var reg=/明明|静静/g;

do{

var arr=reg.exec(str);

console.log(arr)

}while(...);

就可以自动向后找所有

答: /正则/是创建一个正则表达式对象的意思。

/正则/放在do外部，只创建一个，反复使用

/正则/放在do内部，每循环一次，都创建新的reg对象

又因为exec之所以能向后找，是因为reg对象上有一个lastIndex属性指定累加下次开始的位置。

如果只创建一个reg对象，则reg.lastIndex可反复累加。而如果每次创建新的reg对象，都会导致新reg.lastIndex重新回到0.

所以，放在外边，可自动向后找。放在内部，只能不断找第一个关键词。

