### 1、TypeError: this constructor takes no arguments？

一般都是初始化出问题，\_\_init\_\_():这里是前后都是两个下划线

### 2、socket.error: [Errno 113] No route to host

这里我的解决方法是：关掉防火墙

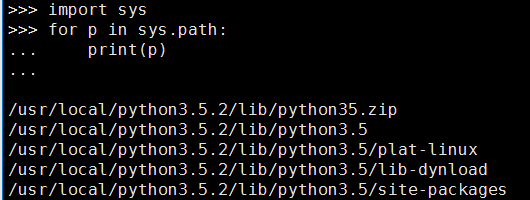
### 3、多行注释用三单引号或三双引号（’’’hello’’’,”””hello”””）

在注释单行时除了使用#,还可以用单引号。

### 4、反斜杠\连接

### 5、Python如何查找导入的模块？

首先是在当前目录下查找.py文件，如果没有则到sys.path所代表的的路径下去查找。有一个空行，代表是当前目录。



### 6、for index, item in enumerate(weekly.forcast(),1):表示：

1表示索引起始值;item 表示列表中的项

### 7、统计文件的行数：

count=0

for index, line in enumerate(open('server.c','r')):

count+=1

### 8、itertools.chain([1,2],[‘a’,’b’])输出的是

1 2 a b.忽略了列表

itertools.cycle([1,2])输出的是:

1 2 1 2 1 2 1 2 …(一直循环，这东西很好)

### 9、使用属性对特性进行访问。

class Duck():

... def \_\_init\_\_(self,input\_name):

... self.\_\_name=input\_name

... @property

... def name(self):

... print("inside the getter")

... return self.\_\_name

... @name.setter #这个属性只是在Python3中有用。

... def name(self,input\_name):

... print("inside the setter")

... self.\_\_name=input\_name

### 10、Python函数1

def score(self,value):

if not isinstance(value,int):

raise ValueError('score must be an integer!')

if value<0 or value>100:

raise ValueError('score must between 0 and 100')

self.\_score=value

### 11、Python函数2

def who\_says(obj):

print(obj.who(),’says’,obj.says())

这里的obj是任何一个具有who(),says()函数的对象。

### 12、self.text.lower() 将对象的属性text小写。

### 13、魔术方法：

\_\_eq\_\_() ne lt gt le ge add sub mul floordiv truediv mod pow

### 14、创建子类是一个is a 的

所谓组合，就是把自定义的类当做别的类的属性

### 15、命名元组

>>>from collections import namedtuple

>>> Duck=namedtuple('Duck','bill tail')

>>> duck=Duck('wide orange','long')

>>> duck

Duck(bill='wide orange', tail='long')

>>> duck.bill

'wide orange'

>>> duck.tail

'long' 15

### 16、关于Python3在Linux上运行安装pip3

需要有两个包pip-8.1.2.tar.gz setuptools-24.0.2.tgz.gz

### 17、一切皆对象，函数没有括号时也是对象。

### 18、位置参数和关键字参数

menu('chardonnay','chicken','cake')

menu(entree='beef',wine='wine',dessert='bagel')

### 19、字符

r u表示原生字符，排除转义字符带来的影响。

u表示utf-8编码

### 20、好好理理Django的逻辑

首先是三个文件 views.py urls.py settings.py,后面两个是创建项目时会自动生成一个同项目名同名的app,在该app中存在这两个文件，而views是每创建一个app都会有的，注意，每创建一个app都需要到settings.py中去注册。每个app中views.py的函数，都可以都可到urls.py中去注册成url地址，以便从网址访问。

Urlpatterns=[

url(r‘^index/$’,nmap\_views.index)

]

前端填写的网址是：$.get("/ajax\_nmap/",function(ret){ ret=JSON.parse(ret)}

两种返回网页的方式

return render(request,’namp/index.html’)

return HttpResponse(var) //这个var是字符串变量或者

### 21、管道符的原理

？？

### 22、type()查看数据对象的类型

### 23、说一下我对json的理解：

(1)首先json不是某种语言中的一种数据结构的名称，json可以Python中的字典，在构造一个json对象时，他就是一个Python字典（使用type(),返回的dict）。

(2)在网络中传输的必然是json字符串，因此在后台json对象要转化成json字符串，在前端要将json字符串转化为json对象

json.dumps(obj, skipkeys=False, ensure\_ascii=True, check\_circular=True, allow\_nan=True, cls=None, indent=None, separators=None, encoding=”utf-8”, default=None, sort\_keys=False, \*\*kw)

Python中：将python对象转化为json字符串：json.dumps()

将json字符串转换成python对象：json.loads()

### 24、Django是如何传传输数据到前端处理的？

前端到后端是request

后端到前端是response

### 25、xml文件和属性控件怎样一一对应？

这个问题问的没有水平，解决这个问题使用的是ztree控件

### 26、一直报错No module named 'xml2db'，但在交互式模式下可以导入

答：在交互式模式下可以导入,这种写法：import xml2db

但是在py文件中还是需要用：from . import xml2db

这个问题困扰我一天。有时候想起来这根本不是问题。

### 27、Xml的理解

根据 DOM，XML 文档中的每个成分都是一个节点。

DOM 是这样规定的：

整个文档是一个文档节点

每个 XML 标签是一个元素节点

包含在 XML 元素中的文本是文本节点

每一个 XML 属性是一个属性节点

注释属于注释节点

XML DOM 属性

一些典型的 DOM 属性：

x.nodeName - x 的名称

x.nodeValue - x 的值

x.parentNode - x 的父节点

x.childNodes - x 的子节点

x.attributes - x 的属性节点

x.firstChildren x的第一个子节点

XML DOM 方法

x.getElementsByTagName(name) - 获取带有指定标签名称的所有元素

x.appendChild(node) - 向 x 插入子节点

x.removeChild(node) - 从 x 删除子节点

注释：在上面的列表中，x 是一个节点对象。

Xml是一个纯文本

大小写敏感

XML 的属性值须加引号

<message>if salary < 1000 then</message>

为了避免这个错误，请用实体引用来代替 "<" 字符：

<message>if salary &lt; 1000 then</message>

<f:table xmlns:f="http://www.w3school.com.cn/furniture">

<f:name>African Coffee Table</f:name>

<f:width>80</f:width>

<f:length>120</f:length>

</f:table>

与仅仅使用前缀不同，我们为 <table> 标签添加了一个 xmlns 属性，这样就为前缀赋予了一个与某个命名空间相关联的限定名称。

xmlns:namespace-prefix="namespaceURI"

当命名空间被定义在元素的开始标签中时，所有带有相同前缀的子元素都会与同一个命名空间相关联。

关于xml中的空元素

空元素不会给我们带来任何好处，唯一的好处是少输入几个字。因此是否需要空元素由用户本人决定。但是需要记住一件事情：就XML而言，<middle></middle>与<middle/>是完全一样的。正是由于这个原因，有些解析器有时会把XML文档从一种形式变换为另一种形式，我们不要依靠文档里的空元素，不管是哪种格式里的空元素。但是从语法上讲，它们是完全一样的，因此这种转换不会产生任何影响(这正是IE随意地把前面的<Address2></Address2>转换为</Address2>的原因)。

### 28、url中的问号?

1）连接作用：比如

http://www.xxx.com/Show.asp?id=77&nameid=2905210001&page=1

2）清除缓存：比如

http://www.xxxxx.com/index.html

http://www.xxxxx.com/index.html?test123123

两个url打开的页面一样，但是后面这个有问号，说明不调用缓存的内容，而认为是一个新地址，重新读取。

### 29、Python类型

Python中nonetype 作为boolean 类型就是false ，而Python中的非是 not

[**http://blog.csdn.net/wbiblem/article/details/72885596**](http://blog.csdn.net/wbiblem/article/details/72885596)

30、ztree的使用，搞过很多方案，使用js解析**，**

以下这些网址都是在开发过程中借鉴的东西。

[**http://www.jb51.net/article/89659.htm**](http://www.jb51.net/article/89659.htm)

[**http://blog.csdn.net/yeluo414/article/details/73290738**](http://blog.csdn.net/yeluo414/article/details/73290738)

[**http://blog.csdn.net/yuanjianfei/article/details/21075807**](http://blog.csdn.net/yuanjianfei/article/details/21075807)

[**http://www.treejs.cn/v3/demo.php#\_102**](http://www.treejs.cn/v3/demo.php#_102)

[**https://www.cnblogs.com/shinhwazt/p/5828031.html**](https://www.cnblogs.com/shinhwazt/p/5828031.html)

[**http://www.jb51.net/Special/925.htm**](http://www.jb51.net/Special/925.htm) **比较全面**

### 31、XMLHttpRequest的一些知识：

ReadyState取值 描述   
0  描述一种"未初始化"状态；此时，已经创建一个XMLHttpRequest对象，但是还没有初始化。   
1  描述一种"发送"状态；此时，代码已经调用了XMLHttpRequest open()方法并且XMLHttpRequest已经准备好把一个请求发送到服务器。   
2  描述一种"发送"状态；此时，已经通过send()方法把一个请求发送到服务器端，但是还没有收到一个响应。   
3  描述一种"正在接收"状态；此时，已经接收到HTTP响应头部信息，但是消息体部分还没有完全接收结束。   
4  描述一种"已加载"状态；此时，响应已经被完全接收。

### 32、出现documentElement of null 是因为xml 对象没有加载成功

//建立XMLdom对象，并载入xml，xmlFilePath为xml的文本路径

function loadXMLDoc(file)

{

var xmlDoc;

try //Internet Explorer

{

xmlDoc=new ActiveXObject("Microsoft.XMLDOM");

xmlDoc.async=false;

xmlDoc.load(file);

}

catch(e)

{

try //Firefox, Mozilla, Opera, etc.

{

xmlDoc=document.implementation.createDocument("","",null);

xmlDoc.async=false;

xmlDoc.load(file);

}

catch(e)

{

try //Google Chrome

{

var xmlhttp = new window.XMLHttpRequest();

xmlhttp.open("GET",file,false);

xmlhttp.send(null);

xmlDoc = xmlhttp.responseXML; //.documentElement

}

catch(e)

{

error=e.message;

}

}

}

return xmlDoc;

}

### 33、getAttribute() 没有该函数？

是因为childNodes属性得到的节点包括文本节点和属性节点、元素节点。而只有元素节点才有getAttribute()函数。故解决办法是使用children属性。

### 34、Cannot set property 'innerHTML' of null

因为JavaScript运行时，id="TreeMenu"的那个div元素可能还没解析和加载。、

### 35、关于js中this的理解

<https://www.cnblogs.com/pssp/p/5216085.html>

### 36、关于js的自定义对象

有三种定义方法：

**方法一:直接定义并创建对象实例**

var obj = new Object(); //创建对象实例

//添加属性obj.num = 5; //添加属性

obj.fn = function( cin ){ return cin;} //添加方法

访问对象的方法:objectName.methodName()

访问对象的属性:objectName.propertyName

**方法二:用函数来定义对象然后创建对象实例**

function Class(){

//添加属性

this.val = 1;

//添加方法

this.fn = function(){

return 'fn';

}

}

var obj = new Class(); //创建对象实例

访问对象的方法:objectName.methodName()

访问对象的属性:objectName.propertyName

方法三:literals 直接创建对象实例（literals是字面量，常量意思）

literals创建对象实例是一种简化方法.

var obj = {}; //创建空对象

var obj = {

num : '2', //这是设置对象属性

add: function(){

....//这是设置对象方法

}

}

访问对象的方法:objectName.methodName()

访问对象的属性:objectName.propertyName

### 37、JavaScript的函数都是有默认返回值的，

如果函数结尾不写return，会默认返回undefined，这就是为什么在chrome的console控制台里，写代码经常下面会出现一行undefined的原因。

<https://www.cnblogs.com/raoyunxiao/p/5644032.html>

### 38、xml转json

<http://blog.csdn.net/wuhawang/article/details/52756516>

### 39、$.each()的理解

$.each(obj,function(i,value) 这里的obj表示要遍历的对象，i表示当前元素的下标

$.each(obj,function(key,value) key表示相当于字典里的key

$.each(obj,function(i) i表示下标

### 40、对xml,json 这一类格式化数据的理解

对象，标签，属性，属性值，文本。

前面用Python处理过xml，js处理过xml

现在用jQuery处理xml，json

### 41、下面这段代码没能遍历xml树的原因：

function BuildZnodes(rootNode,myId,pID){

var zNodes=[];

//这里又得到疑问，如果属性不添加在下面的数据里，将来如何得到属性

zNodes.push({id:myId,pid:pID,name:rootNode.nodeName})

console.log("打印tag:",rootNode.nodeName)

//先得到rootNode的子节点列表

childrenNodeList = rootNode.children;

//debug

console.log("打印childrenNodeList：",childrenNodeList);

//？？？需要得到属性列表吗

//

for(var i=0;i<childrenNodeList.length;i++){

//最后要将列表合并。

console.log("打印i",i);

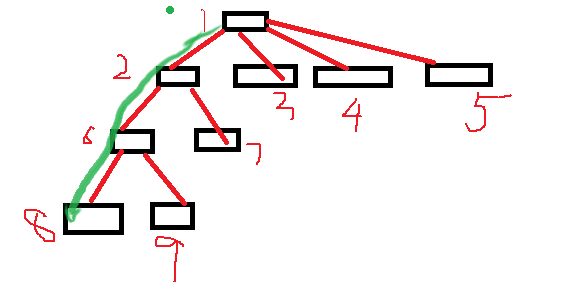
zNodes=zNodes.concat(BuildZnodes(childrenNodeList[i],myId\*10+i+1,myId));

}

return zNodes;

}

上面的代码一直只能访问1268,：



仔细调试了一下代码，发现访问到8的时候，就跳出来了，原因在于：js的变态语法，js中加“var”和不加“var”的区别，看完觉得这么多年js白学了,加var的变量在当前作用域，不加var 在全局。

Javascript声明

<https://www.cnblogs.com/liuna/p/6140901.html>

### 42、js闭包的使用：

1、从函数外部访问函数内部的变量。

2、**如何将变量的值始终保存在内存当中？**

<http://www.jb51.net/article/87567.htm>

function f1(){

　　var n=999;

　　function f2(){

　　　　alert(n);

　　}

　　return f2;

}

var result=f1();

result(); // 999

### 43、这次本来以为能够成功的，结果因为Maximum call stack size exceeded

（超过最大的栈内存调用，这一点导致的原因有死循环，无限递归，可我只是加了个数据）

zNodes.push({id:myId,pid:pID,name:rootNode.nodeName,data: attrNode});

attrNode是一个NamedNodeMap类型的数据，听说是动态的。

<http://blog.csdn.net/u012207345/article/details/78165931>

NamedNodeMap集合的元素是attr元素拥有name和value属性，分别表示元素特性名称和特性值

先试一下解决方案一，将这个类型转换成json或者列表，在js中map对象可以储存键值对，非常好用，<https://www.cnblogs.com/xcxcxcxc/p/6841528.html>，这个网址里是map和json字符串互相转换的。很好很强大。

### 44、上面的这个问题终于解决掉了

### 45、iframe

<iframe id="nodeIframe" name="nodeIframe" width="100%" height="600" scrolling="auto" ></iframe>这个东西可以载入另外一个网页，只需要给定网址就行。

### 46、碰到一个新的问题，前端如何请求一个xml文件

这个问法其实是有错误的，（暂时认为的），我的解决方式是前端请求的xml字符串，这个xml字符串是直接从xml文件中读取的。使用的是Python 的open函数。在HttpResponse中设置content-type为application/xml。前端接收方也设置为xml 类型，这样前端为自动将xml类型的字符串解析为xml对象，这一点非常好。

### 47、chrome的取色器问题

<http://blog.csdn.net/ardo_pass/article/details/78956172>

### 48、xml的文本节点

刚刚发现出现一个小问题，加载了xml的元素节点和属性节点却忘记了文本节点。这是一件非常恼火的事情。

//之所以大于1，取巧，是因为有些节点的文本存在特殊字符，只占一个长度;

if(rootNode.firstChild && (rootNode.firstChild.nodeType==3) && (rootNode.firstChild.nodeValue).length>1){

// console.log(rootNode.nodeName,"文本节点：",rootNode.firstChild,rootNode.firstChild.nodeValue);

data["text\_str"]=rootNode.firstChild.nodeValue;

}

### 49、关于前端直接访问后端一个xml文件的问题

url会报错，之所以会这样，因为url的访问都会经过urls.py的处理，所以前端的访问都必须经过urls.py这样一个转发站。