链接: https://www.cnblogs.com/wupeiqi/p/9078770.html

<https://mp.weixin.qq.com/s?__biz=MzA5ODUzOTA0OQ==&mid=2651689264&idx=1&sn=11fbc3076ad08dc22bf80eb975c8ee14&chksm=8b693276bc1ebb60cdf08d14a5856050edee89406b9f94679c0b78765921cd04b3702cae6381&mpshare=1&scene=1&srcid=0601iRcbOSz3LQTWArSeIWqS#rd>

# Python框架

## ****1、什么是pickling和unpickling？****

      Pickle模块接受任何Python对象，并将其转换为字符串，使用dump函数将其转储到文件中，这个过程称为pickling。从存储的字符串表示中检索原始Python对象的过程称为unpickling。

## ****2、作为解释型语言，Python如何运行？****

      Python是一种解释型语言。Python程序直接从源代码运行，将程序员编写的源代码转换成中间语言，再将中间语言翻译成必须执行的机器语言。

## ****3、有哪些工具可以帮助查找错误或执行静态分析？****

      PyChecker是一个静态分析工具，用于检测Python源代码中的错误，并给出错误的类型和复杂性。Pylint是验证模块是否符合编码标准的另一种工具。

## ****4、按下述要求编写代码，实现以下功能：****

（1）编写代码下载https://en.wikipedia.org/wiki/Machine\_translation 页面的内容并保存为mt.html

（2）统计mt.html中<p>标签下所有单词并存储到mt\_word.txt中，要求：

     a) 每个单词一行。单词在前，单词出现的次数在后，中间用Tab()分隔。

     b) 单词按照数目从多到少排列。比如说单词a出现了100次，单词b出现了10次，则单词a要在单词b前面。

## ****5、下面的代码会输出什么：****

def f(x,l=[]):

for i in range(x):

l.append(i\*i)

print l

f(2)f(3,[3,2,1])f(3)

答案：

[0, 1][3, 2, 1, 0, 1, 4][0, 1, 0, 1, 4]

## ****6、****stringify与parse方法

JSON.parse(): 用于将一个 JSON 字符串转换为 JavaScript 对象

eg:

console.log(JSON.parse('{"name":"Yuan"}'));

console.log(JSON.parse('{name:"Yuan"}')) ; // 错误

console.log(JSON.parse('[12,undefined]')) ; // 错误

JSON.stringify(): 用于将 JavaScript 值转换为 JSON 字符串。

eg: console.log(JSON.stringify({'name':"egon"})) ;

2、AJAX的优缺点

AJAX使用Javascript技术向服务器发送异步请求；

AJAX无须刷新整个页面；

因为服务器响应内容不再是整个页面，而是页面中的局部，所以AJAX性能高；

缺点：

AJAX并不适合所有场景，很多时候还是要使用同步交互；

AJAX虽然提高了用户体验，但无形中向服务器发送的请求次数增多了，导致服务器压力增大；

因为AJAX是在浏览器中使用Javascript技术完成的，所以还需要处理浏览器兼容性问题；

## **7、Python中lambda是什么意思？（1分）**

      它是一个经常用作内联函数的单个表达式匿名函数。

## **8、为什么python中的lambda表单没有语句？（1分）**

      python中的lambda表单没有语句，因为它用于创建新的函数对象，然后在运行时返回它们。

## **9、Python中的pass是什么意思？（1分）**

      pass意味着没有任何操作的Python语句，换句话说，它是复合语句中的一个占位符，如果一个地方没有什么必须写在那里，就需要用上pass了。

**10、阅读下面的代码，写出A0，A1至An的最终值。（5分）**

A0 = dict(zip(('a','b','c','d','e'),(1,2,3,4,5)))

A1 = range(10)

A2 = [i for i in A1 if i in A0]

A3 = [A0[s] for s in A0]

A4 = [i for i in A1 if i in A3]

A5 = {i:i\*i for i in A1}

A6 = [[i,i\*i] for i in A1]

{'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}

range(0, 10)

[]

[1, 2, 3, 4, 5]

[1, 2, 3, 4, 5]

{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}

## 11、什么是Python的单元测试？（1分）

      Python中的单元测试框架被称为unittest。它支持共享设置，自动化测试，测试关机代码，测试集合等。

## ****12、在Python中unittest是什么？（1分）****

      从列表，元组，字符串等序列类型中选择一系列项目的机制被称为unittest。

## **13、什么是Python中的生成器？**

     实现迭代器的方式被称为生成器。除了在函数中产生表达式之外，它是一个正常的函数。

**14、\_\_new\_\_和\_\_init\_\_的区别？**

解释一：首先要知道在面向对象编程中，实例化基本遵循创建实例对象、初始化实例对象、最后返回实例对象这么一个过程。Python 中的 ***\_new\_*** 方法负责创建一个实例对象，***\_init\_*** 方法负责将该实例对象进行初始化；​

解释二：有人说如果将类比喻为工厂，那么***\_init\_***方法则是该工厂的生产工人，***\_init\_***方法接受的初始化参数则是生产所需原料，***\_init\_***方法会按照方法中的语句负责将原料加工成实例以供工厂出货。而***\_new\_***则是生产部经理，***\_new\_***方法可以决定是否将原料提供给该生产部工人，同时它还决定着出货产品是否为该生产部的产品，因为这名经理可以借该工厂的名义向客户出售完全不是该工厂的产品。

## ****15、如何复制Python中的对象？****

      要在Python中复制对象，一般情况下可以尝试copy.copy（）或copy.deepcopy（）。不能复制所有的对象，但大多数还是可以的。

## ****16、如何将数字转换为字符串？****

      为了将数字转换为字符串，使用内置函数str（）。如果想要一个八进制或十六进制表示，使用内置函数oct（）或 hex()。

## ****17、Xrange和range有什么区别？****

      Xrange返回一个xrange对象，而range返回一个数组。不管范围多大，使用同样的内存。

## ****18、什么是Python中的模块和包？****

      在Python中，模块是构造程序的方式。每个Python程序文件都是一个模块，它导入其他模块，如对象和属性。

      Python程序的文件夹是一个模块包，包可以有模块或子文件夹。

## **19、提到Python中的局部和全局变量的规则是什么？**

      局部变量：如果一个变量在函数体内的任何地方被分配了一个新的值，它被认为是本地的。

      全局变量：使用global定义的变量就是全局变量

当局部变量名字和全局变量名字重复时，局部变量会覆盖掉全局变量。

## ****20、怎样才能跨模块共享全局变量？****

      要在单个程序的模块之间共享全局变量，请创建一个配置模块。在应用程序的所有模块中导入配置模块，该模块将作为跨模块的全局变量提供。

## **21、解释如何在Unix上创建一个Python脚本可执行文件？**

      要在Unix上创建Python脚本可执行文件需要做两件事情：

**·**Script文件的模式必须是可执行的

**·**第一行必须以＃（＃！/ usr / local / bin / python）开头

## **22、Python垃圾回收机制**

      Python GC主要使用引用计数（reference counting）来跟踪和回收垃圾。在引用计数的基础上，通过“标记-清除”（mark and sweep）解决容器对象可能产生的循环引用问题，通过“分代回收”（generation collection）以空间换时间的方法提高垃圾回收效率。

## **23、解释如何在Python中生成随机数字？**

      要在Python中生成随机数需要将命令导入

随机导入：random.random（）

这将返回范围[0,1）中的随机浮点数

## **24、解释如何访问用C语言编写的Python模块？**

      你可以通过下面的方法访问一个用C写成的模块，

Module = = PyImport\_ImportModule（“<modulename>”）;

## **25、在Python中如何使用//运算符？**

      它是一个Floor Division Operator，用于将两个操作数相除，结果为小数点前面的数字。例如，10 // 5 = 2和10.0 // 5.0 = 2.0。

## **26、提到使用Python的五个好处？**

**·**Python包含了大多数互联网平台（如电子邮件，HTML等）的巨大标准库。

**·**Python不需要显式的内存管理，因为解释器本身将内存分配给新变量并自动释放它们

**·**由于使用方括号而提供易读性

**·**易于初学者学习

**·**具有内置的数据类型，可以节省编程时间和工作量，从而声明变量。

## **27、简单说明在Python中如何使用split函数？**

      在Python中使用split函数是使用定义的分隔符将字符串分解成更短的字符串。它给出了字符串中所有单词的列表。

## **28、解释什么是Flask及其好处？**

      Flask是一个基于“Werkzeug，Jinja 2和良好意图”BSD许可的web微型框架，Werkzeug和jingja是它的两个依赖项。

      Flask是微观框架的一部分。这意味着它将很少或不依赖于外部库，它使框架轻而易举，更新和安全漏洞更少。

## **29、Django，Pyramid和Flask有什么区别？**

      Flask是一个“微框架”，主要用于需求更简单的小型应用程序。在Flask中，你必须使用外部库。

      Pyramid是为更大的应用程序建立的。它提供了灵活性，并让开发人员为他们的项目使用正确的工具。开发人员可以选择数据库，URL结构，模板样式等等。

      Pyramid可重新配置。像Pyramid一样，Django也可以用于更大的应用程序。 它包括一个ORM。

## ****30、Flask-WTF是什么，有什么特点？****

     Flask-WTF提供了与WTForms的简单集成，功能包括：

**·**与wtforms集成

**·**使用csrf令牌安全形式

**·**全球csrf保护

**·**Reptcha支持

**·**与Flask Uploads一起使用的文件上传

## **31、Flask脚本的常用方式是什么？**

      应该是应用程序的导入路径或Python文件的路径

## **32、如何在Flask中访问会话？**

      一个会话基本上允许记住从一个请求到另一个请求的信息。在Flask中，它使用签名的cookie，以便用户可以查看会话内容并进行修改。用户可以修改会话，只要它有密钥Flask.secret\_key。

## **33、Flask是一个MVC模型吗？如果是，可以示例一下吗？**

基本上，Flask是一个简单的框架，其行为与MVC框架相同。所以MVC是Flask的完美选择，示例参考如下：

## **34、解释Python Flask中的数据库连接？**

      Flask支持数据库驱动的应用程序（RDBS）。这样的系统需要创建一个模式，将shema.sql文件传送到sqlite3命令。所以需要sqlite3命令才能在Flask中创建或启动数据库。

      Flask允许以三种方式请求数据库

**·**before\_request（）：它们在请求前被调用并且不传递任何参数

**·**after\_request（）：它们在请求之后被调用并且传递将被发送到客户端响应

**·**teardown\_request（）：在引发异常的情况下调用，并且不保证响应。他们在响应结束后被调用。他们不允许修改请求，他们的值被忽略。

## ****35、你有多个运行Python的Memcache服务器，其中一个memcacher服务器失败，它有你的数据，它会试图从那个失败的服务器获取关键数据吗？****

       发生故障的服务器中的数据不会被删除，但是可以为多个节点配置自动故障规定。可以在任何类型的套接字或Memcached服务器级错误期间触发故障切换，而不会在正常的客户端错误（如添加现有密钥等）期间触发。

## **36、解释如何最大限度地减少Python开发中的Memcached服务器中断？**

       当一个实例失败，这将在客户端发出请求时重新加载丢失的数据，在数据库服务器上承受更大的负载。为了避免这种情况，如果代码已经写入，尽量减少缓存的冲击，那么它将产生最小的影响

       另一种方法是使用丢失的机器IP地址在新机器上启动Memcached实例

       代码是最大限度减少服务器停机的另一种方法，因为它可以自由地以最少的工作更改Memcached服务器列表

       设置超时值是一些Memcached客户端为Memcached服务器中断实现的另一个选项。当Memcached服务器关闭时，客户端将不断尝试发送请求，直到达到超时限制

## **37、解释Python项目中应不应该使用Memcached？**

      Memcached常见的误用是将其用作数据存储，而不是用作缓存

      切勿使用Memcached作为运行应用程序所需信息的唯一来源，数据应该始终可以通过其他来源获得

      Memcached只是一个键或值存储，不能对数据执行查询或遍历内容以提取信息

      Memcached在加密或认证时不提供任何形式的安全性