

Python 概念

Python 是一种解释型，交互式，面向对象的高级编程语言。和别的一些使用标点符号的语言不同，Python使用了大量的英语单词作为关键字，因而具有很好的可读性。而且跟其他编程语言相比，它有更少的语法结构。

Python 是一种解释性语言：这意味着你的程序在执行之前不需要编译，而是由解释器在运行时处理。这个特点跟PERL，PHP很像。

Python是可交互的：这意味着你可以使用一个Python终端在写程序时和解释器直接交互。

Python是面向对象的：它支持面向对象风格和将代码封装成类的各种技术。

Python非常适合编程初级人员：Python非常容易入门而且应用领域也非常广泛，从简单的文字处理到网络浏览和游戏开发。

1) 什么是Python? 使用Python有什么好处?

Python是一种编程语言，它有对象，模块，线程，异常处理和自动内存管理。它简洁，简单，方便，容易扩展，有许多自带的数据结构，而且它开源。

2) 什么是PEP8?

PEP8 是一个编程规范，一些关于如何让你的程序更具有可读性的建议。

3) 什么是pickling和unpickling?

Pickle模块读入任何Python对象，将它们转换成字符串，然后使用dump函数将其转储到一个文件中——这个过程叫做pickling。反之从存储的字符串文件中提取原始Python对象的过程叫做unpickling。

4) Python 是如何被解释的？

Python是一种解释性语言。Python的源代码可以直接运行。Python解释器会将源代码转换成中间语言，之后再翻译成机器码再执行。

5) Python是怎样管理内存的？

Python的内存管理是由私有heap空间管理的。所有的Python对象和数据结构都在一个私有heap中。程序员没有访问该heap的权限，只有解释器才能对它进行操作。

为Python的heap空间分配内存是由Python的内存管理模块进行的。其核心API提供一些访问该模块的方法供程序员使用。

Python有自带的垃圾回收系统，它回收并释放没有被使用的内存让它们能够被其他程序使用。

6) 有哪些工具可以帮助debug或做静态分析？

PyChecker 是一个静态分析工具，它报告源代码中的错误并且会报告错误类型和复杂度。Pylint是检验模块是否达到代码标准的另一个工具。

7) 什么是Python装饰器？

Python装饰器是Python中的特有变动，可以使修改函数变得更容易。

8) 数组和元组之间的区别是什么？

数组和元组之间的区别是数组内容是可以被修改的而元组内容是只读的。元组可以被哈希比如作为字典的关键字。

9) 参数按值传递和引用传递是怎样实现的？

Python中的一切都是类，所有的变量都是一个对象的引用。引用的值是由函数确定的，因此无法被改变。但是如果一个对象是可以被修改的你可以改动对象。

10) 字典推导式和列表推导式是什么？

他们是可以轻松创建字典和列表的语法结构。

11) Python都有那些自带的数据结构？

Python自带的数据结构分为可变的和不可变的。

可变的有：

- 数组
- 集合
- 字典

不可变的有：

- 字符串

- 元组
- 数

12) 什么是Python的命名空间?

在 Python 中，所有的名字都存在于一个空间中，它们在该空间中存在和被操作——这就是命名空间。它就好像一个盒子，每一个变量名字都对应装着一个对象。当查询变量的时候，会从该盒子里面寻找相应的对象。

13) Python 中的lambda 是什么?

这是一个常被用于代码中的单个表达式的匿名函数。

14) 为什么 lambda 没有语句?

匿名函数lambda没有语句的原因是它被用于在代码被执行的时候构建新的函数对象并且返回。

15) Python中的pass是什么?

Pass是一个在Python中不会被执行的语句。在复杂语句中，如果一个地方需要暂时被留白，它常常被用于占位符。

16) Python中什么是遍历器?

遍历器用于遍历一组元素，比如列表这样的容器。

17) Python中的unittest是什么?

在Python中，unittest是Python中的单元测试框架。它支持共享搭建，自动测试，在测试中暂停代码，将不同测试迭代成一组等等功能。

18) 在Python中什么是slicing?

Slicing是一种在有序的对象类型中（数组，元组，字符串）节选某一段的语法。

19) 在Python中什么是生成器?

生成器是实现迭代器的一种机制。它功能的实现依赖于yield表达式，除此之外它跟普通的函数没有两样。

20) Python 中docstring什么?

在 Python 中文档字符串被称为docstring，它被用于在Python中为函数，模块和类注释生成文档。

21) 在Python中如何拷贝一个对象?

如果要在Python中拷贝一个对象，大多时候你可以用copy.copy () 或者copy.deepcopy()。并不是所有的对象都可以被拷贝。

22) Python中的负索引是什么?

Python中的序列索引可以是正也可以是负。如果是正索引，0是序列中的第一个索引，1是第二个索引。如果是负索引，（-1）是最后一个索引而（-2）是倒数第二个索引。

23) 如何将一个数字转换成一个字符串？

你可以使用自带函数`str()`将一个数字转换为字符串。如果你想要八进制或者十六进制数，可以用`oct()` 或 `hex()`。

24) Xrange和range的区别是什么？

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

4. range()和xrange() 函数的差异

range()生成的是list，而**xrange ()** 生成的是迭代器 (iterator) 。例如：

```
range(5)
# [0, 1, 2, 3, 4]
```

```
xrange(5)
```

```
list(xrange(1, 5))
# [1, 2, 3, 4]
```

25) Python中的模块和包是什么？

在Python中，模块是搭建程序的一种方式。每一个Python代码文件都是一个模块，并可以引用其他的模块比如对象和属性。

一个包含许多Python代码的文件夹是一个包。一个包可以包含模块和子文件夹。

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。

14、__new__和__init__的区别（4分）

1. __new__是一个静态方法,而__init__是一个实例方法.
2. __new__方法会返回一个创建的实例,而__init__什么都不返回.
3. 只有在__new__返回一个cls的实例时后面的__init__才能被调用.
4. 当创建一个新实例时调用__new__,初始化一个实例时用__init__

19、提到Python中的局部和全局变量的规则是什么？（3分）

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？（3分）

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？（4分）

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（3分）

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？（3分）

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？（3分）

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？（1分）

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？（2分）

- 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
- 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？（1分）

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。

3. *init.py*是什么

一个空文件，用于将导入目录中的模块。比如有一个模块是maindir/subdir/module.py, *init.py* 可以使之可以通过以下形式导入。 `import maindir.subdir.module`

5. 如何对list中的item进行随机重排

使用`shuffle()`函数

7. 装饰器的作用

装饰器可用于修饰函数或类。通过装饰器可以包裹函数或类使之执行之前或之后调用装饰器函数，从而达到抽离出大量函数中与函数功能本身无关的雷同代码并继续重用的目的。比较经典

的厨力场景有检验权限，追踪参数，日志登陆等。

```
def makebold(fn):
    def wrapped():
        return "<b>" + fn() + "</b>"
    return wrapped
def makeitalic(fn):
    def wrapped():
        return "<i>" + fn() + "</i>"
    return wrapped
@makebold@makeitalicdef hello():
    return "hello world"
```