# 第一部分：Python基础

## 1.0 Linux命令、Python基础

**【知识点】命令的使用**

【问题描述1】什么是相对路径和绝对路径

【问题描述2】权限的命令是怎么用的？

【问题描述3】能一次性创建多个文件夹吗

## python简介 表达式、运算符

**【知识点1】变量**

【问题描述1】a1 = a2 =3 ， a1 和a2的ID 一样吗？

【问题描述2】is 和 == 有什么区别？

**【知识点2】数据类型的强转转换**

【问题描述】十六进制和八进制之间能转换吗？

## 1.2 Python变量 基本输入输出 关系运算 布尔运算 条件语句

**【知识点】运算符**

【问题描述1】运算符优先级是怎样的？

【问题描述2】print基本输出语法里的sep 和 end的用法

**【知识点】 if 语句**

【问题描述】 if 语句的pass 语句后面的语句还执行吗？

## 1.3 字符串 字符串访问 字符串格式化 RAW字符串

**【知识点1】布尔运算**

【问题描述】bool（）后能跟表达式吗

**【知识点2】 字符串**

【问题描述1】单引号双引号 和三引号的区别

【问题描述2】字符串的索引和切片，后面跟得数据能一样的嘛？

L = [1,2,3,4]

L[1:2] = 6 #为什么会报错

而 L[1] = 6 #不报错，为什么l

## 1.4 while 循环 break continue for 循环

【问题描述1】 死循环为什么不执行else 语句

【问题描述2】while 循环语句的嵌套关系，每一步是怎么执行的？

【问题描述3】while 循环和for循环执行continue的下一步是返回到哪里？

## 1.5 列表 列表访问 列表常用方法 列表推导式式

**【知识点 range函数】**

【问题描述1】range（4,0） #空

【问题描述2】range（4,0，-1） #4,3,2,1

**【知识点 for循环】**

【问题描述】for 循环嵌套的执行步骤是怎么样？、

**【知识点 in运算符】**

x = [1, 'two', 3, '四']

3 in x # True

'3' in x # False

## 1.6 元组 元组常用方法

**【知识点 列表的切片】**

【问题描述1】 列表的切片后的变量是同一个列表吗？

>>> L = [2, 3, 4]

>>> id(L)

139697672849160

>>> L[0:1] = [1.1, 2.2]

>>> L

[1.1, 2.2, 3, 4]

>>> id(L)

139697672849160

**【知识点 元组的构造函数】**

【问题描述1】

t = (20) # t 绑定20这个对象，不是元组

>>> type(t)

<class 'int'>

t = (20,) #构造一个元祖

>>> type(t)

<class 'tuple'>

【问题描述2】元祖的索引和切片赋值是怎么索引和赋值的？

元组不支持索引赋值和切片赋值(因为元组不可被改变)

## 1.7 字典 字典常用方法

【问题描述1】 怎么用可迭代对象构造字典？

d = dict([('name', 'tarena'), ['age', '15']])

【问题描述2】怎么取字典的键值对？

for k，v in d.items():

print(k,v)

【问题描述3】字典的键为什么是不可变的类型？

## 1.8 集合 集合推导式

**【知识点】**

【问题描述1】集合和字典的关联是什么？

【问题描述2】集合和固定集合的具体的区别是什么？

【问题描述3】怎么用集合去除列表中的重复元素

>>> l = [1,4,9,12,9,4,1]

>>> l = list(set(l))

>>> l

[1, 4, 12, 9]

## 1.9 函数 函数参数传递

**【知识点】**

【问题描述1】为什么要用函数？

【问题描述2】说明\*args 和 \*\*kwargs 的区别？

【问题描述3】函数传参的方式有哪几种，有没有先后顺序?

## 1.10 递归函数 作用域

**【知识点】**

【问题描述1】函数的嵌套的执行步骤是怎么样的？

【问题描述2】Python作用域内变量的查找顺序？

【问题描述3】什么是匿名函数？他有什么好处？

【问题描述4】什么是递归，分别说明递归函数的优点和缺点

## 1.11 高阶函数 map filter sorted eval exec

【问题描述1】高阶函数sorted中，key值的用法

【问题描述2】eval 和 exec 的区别是什么？

【问题描述3】递归的思想是什么？应该怎么修改递归的深度？

## 1.12 闭包 装饰器

**【知识点】**

【问题描述1】是什么是闭包？闭包作用是什么？

【问题描述2】装饰器的具体功能是什么？

1.引入日志；2.函数执行时间统计；3.执行函数前预备处理；

4.执行函数后清理功能；5.权限校验；6.缓存

【问题描述3】函数的 \_\_doc\_\_ 是什么作用？

## 1.13 模块 包开发

【问题描述1】模块的导入有哪几种方法？

【问题描述2】Python是解释执行语言，是怎么解释执行的？

【问题描述3】为什么要用\_\_init\_\_？

## 1.14 异常 断言

【问题描述1】介绍一下except 的用法和作用？

【问题描述2】在异常处理语句中，raise和assert的用法和作用

【问题描述3】为什么要异常处理机制:

在程序调用层数较深时,向主调函数传递错误信息需要层层return 返回比较麻烦,用异常处理机制可以较简单的传送错误信息

## 1.15 迭代器 生成器

【问题描述1】为什么要使用生成器？

使用生成器表达式取代列表推导式可以同时节省 cpu 和 内存(RAM)。

【问题描述2】yield 函数的用法

【问题描述3】bytes 和 str 的区别:

bytes 存储字节( 通常值在 range(0, 256))

str 存储unicode字符( 通常值在0~65535)

## 1.16 文件 标准输入输出文件

【问题描述1】read readline 和readlines 的用法

read 读取整个文件

readline 读取下一行，使用生成器方法

readlines 读取整个文件到一个迭代器一共我们遍历

【问题描述2】标准文件需要打开和关闭吗？

【问题描述3】-\*- coding:gbk -\*-的作用是什么？

## 1.17 对象和类

【问题描述1】类的实例方法和类的方法的区别是什么？

【问题描述2】函数里为什么要写\_\_init\_\_？

【问题描述3】什么是类的析构方法？

## 1.18 函数重写 str/repr 函数重写

**【知识点1】重载与重写的区别**

【问题描述1】重载与重写的具体区别如何？分别在什么情景下使用？

【问题描述2】示例方法和实例变量有什么区别？

【问题描述3】面向对象的思想是什么？

## 1.19 继承 派生 封装 多态

【问题描述1】类的继承的方法有哪些，什么是继承

【问题描述2】覆盖在类的继承里有有什么作用？

【问题描述3】super函数的作用是什么？

## 1.20 异常高级 运算符重载

【问题描述1】对象转字符串函数的重写方法有哪些？

【问题描述2】什么是面向对象的mro

【问题描述3】 异常处理写法以及如何抛出异常