python就业班

01基础

# 第1节 linux操作系统基础

# 第2节 python语法基础

## 1.基础

### 1.1注释

# 单行注释

’’’

多行注释

’’’

在代码第一行写入执行时的python解释器路径，编辑完后需要对此python文件添加'x'权限。之后直接./ 运行

#!/usr/bin/python

#coding=utf-8

# -\*- coding:utf-8 -\*-

### 1.2变量以及类型

#### （1）定义

#### （2）类型

Numbers（数字）：int（有符号整型）

long（长整型[也可以代表八进制和十六进制]）

float（浮点型）

complex（复数）

布尔类型

String（字符串）

List（列表）

Tuple（元组）

Dictionary（字典）

#### （3）查看变量类型：type(变量名字)

### 1.3标识符和关键字

#### 标识符：

由字母、下划线和数字组成，且数字不能开头。且区分大小写。

#### （2）命名规则：

小驼峰式命名法（lower camel case）： 第一个单词以小写字母开始；第二个单词的首字母大写。

大驼峰式命名法（upper camel case）： 每一个单字的首字母都采用大写字母。

用下划线“\_”来连接所有的单词。

#### （3）关键字

and as assert break class continue def del

elif else except exec finally for from global

if ino import is lambda not or pass

print raise return try while with yield

#### （4）查看关键字

import keyword

keyword.kwlist()

### 1.4输出：print

#### （1）格式化输出

print("我今年%d岁"%age) #没有逗号

print("我的姓名是%s,年龄是%d"%(name,age))

#### （2）常用格式符

| 格式符号 | 转换 |
| --- | --- |
| %% | 输出% |
| %c | 字符 |
| %s | 通过str() 字符串转换来格式化 |
| %i | 有符号十进制整数 |
| %d | 有符号十进制整数 |
| %u | 无符号十进制整数 |
| %o | 八进制整数 |
| %x | 十六进制整数（小写字母） |
| %X | 十六进制整数（大写字母） |
| %e | 索引符号（小写'e'） |
| %E | 索引符号（大写“E”） |
| %f | 浮点实数 |
| %g | ％f和％e 的简写 |
| %G | ％f和％E的简写 |

（3）换行：\n

（4）不换行：print(“hello word”,end=” ”)

### 1.5输入：input(“提示”)

input()的小括号中放入的是，提示信息，用来在获取数据之前给用户的一个简单提

input()在从键盘获取了数据以后，会存放到等号右边的变量中

input()会把用户输入的任何值都作为字符串来对待

### 1.6运算符

#### （1） 算术运算符

| 运算符 | 描述 | 实例 |
| --- | --- | --- |
| + | 加 | 两个对象相加 a + b 输出结果 30 |
| - | 减 | 得到负数或是一个数减去另一个数 a - b 输出结果 -10 |
| \* | 乘 | 两个数相乘或是返回一个被重复若干次的字符串 a \* b 输出结果 200 |
| / | 除 | x除以y b / a 输出结果 2 |
| // | 取整除 | 返回商的整数部分 9//2 输出结果 4 , 9.0//2.0 输出结果 4.0 |
| % | 取余 | 返回除法的余数 b % a 输出结果 0 |
| \*\* | 幂 | 返回x的y次幂 a\*\*b 为10的20次方， 输出结果 100000000000000000000 |

#### （2）赋值运算符

| 运算符 | 描述 | 实例 |
| --- | --- | --- |
| = | 赋值运算符 | 把=号右边的结果给左边的变量 num=1+2\*3 结果num的值为7 |

#### （3）复合赋值运算符

| 运算符 | 描述 | 实例 |
| --- | --- | --- |
| += | 加法赋值运算符 | c += a 等效于 c = c + a |
| -= | 减法赋值运算符 | c -= a 等效于 c = c - a |
| \*= | 乘法赋值运算符 | c \*= a 等效于 c = c \* a |
| /= | 除法赋值运算符 | c /= a 等效于 c = c / a |
| %= | 取模赋值运算符 | c %= a 等效于 c = c % a |
| \*\*= | 幂赋值运算符 | c \*\*= a 等效于 c = c \*\* a |
| //= | 取整除赋值运算符 | c //= a 等效于 c = c // a |

a=10

a\*=10-4

a->60 不是96（复合运算符会把后面当成一个整体）

#### （4）比较(即关系)运算符

python中的比较运算符如下表

| 运算符 | 描述 | 示例 |
| --- | --- | --- |
| == | 检查两个操作数的值是否相等，如果是则条件变为真。 | 如a=3,b=3则（a == b) 为 true. |
| != | 检查两个操作数的值是否相等，如果值不相等，则条件变为真。 | 如a=1,b=3则(a != b) 为 true. |
| > | 检查左操作数的值是否大于右操作数的值，如果是，则条件成立。 | 如a=7,b=3则(a > b) 为 true. |
| < | 检查左操作数的值是否小于右操作数的值，如果是，则条件成立。 | 如a=7,b=3则(a < b) 为 false. |
| >= | 检查左操作数的值是否大于或等于右操作数的值，如果是，则条件成立。 | 如a=3,b=3则(a >= b) 为 true. |
| <= | 检查左操作数的值是否小于或等于右操作数的值，如果是，则条件成立。 | 如a=3,b=3则(a <= b) 为 true. |

#### （5）逻辑运算符

| 运算符 | 逻辑表达式 | 描述 | 实例 |
| --- | --- | --- | --- |
| and | x and y | 布尔"与" - 如果 x 为 False，x and y 返回 False，否则它返回 y 的计算值。 | (a and b) 返回 20。 |
| or | x or y | 布尔"或" - 如果 x 是 True，它返回 True，否则它返回 y 的计算值。 | (a or b) 返回 10。 |
| not | not x | 布尔"非" - 如果 x 为 True，返回 False 。如果 x 为 False，它返回 True。 | not(a and b) 返回 False |

### 1.7数据类型转换

| 函数 | 说明 |
| --- | --- |
| int(x [,base ]) | 将x转换为一个整数 |
| long(x [,base ]) | 将x转换为一个长整数（在python3中实测无） |
| float(x ) | 将x转换到一个浮点数 |
| complex(real [,imag ]) | 创建一个复数 |
| str(x ) | 将对象 x 转换为字符串 |
| repr(x ) | 将对象 x 转换为表达式字符串 |
| eval(str ) | 用来计算在字符串中的有效Python表达式,并返回一个对象 |
| tuple(s ) | 将序列 s 转换为一个元组 |
| list(s ) | 将序列 s 转换为一个列表 |
| chr(x ) | 将一个整数转换为一个字符（ASCII） |
| unichr(x ) | 将一个整数转换为Unicode字符（在python3中实测无） |
| ord(x ) | 将一个字符转换为它的整数值（ASCII） |
| hex(x ) | 将一个整数转换为一个十六进制字符串 |
| oct(x ) | 将一个整数转换为一个八进制字符串 |

## 2.判断与循环

### 2.1 if

if 条件1:

事件1

…

elif 条件2:

事件2

…

else:

事件3

...

### 2.2 while

while 条件1:

条件1满足时，做的事情1

条件1满足时，做的事情2

条件1满足时，做的事情3

...(省略)...

### 2.3 for

for 临时变量 in 列表或者字符串等:

循环满足条件时执行的代码

else:

循环不满足条件时执行的代码

### 2.4 break/continue

（1）break的作用：用来结束整个循环

（2）continue的作用：用来结束本次循环，紧接着执行下一次的循环

（3）break/continue只能用在循环中，除此以外不能单独使用

（4）break/continue在嵌套循环中，只对最近的一层循环起作用

## 3.字符串、列表、元组、字典

### 3.1字符串

#### 字符串介绍：

双引号或者单引号中的数据

#### （2）字符串组合

##### <1>直接加或者乘

a = “ha”

b = “world”

c = a+b

##### <2>**格式**控制符

a = “ha”

b = “%sworld”%(a)

#### （3）下标：string[0]

#### （4）切片：

切片是指对操作的对象截取其中一部分的操作。字符串、列表、元组都支持切片操作。

切片的语法：[起始:结束:步长]

注意：选取的区间属于左闭右开型，即从"起始"位开始，到"结束"位的前一位结束（不包含结束位本身)。

#### （5）字符串常见操作：

##### <1>find：检测 str 是否包含在 mystr中，如果是返回开始的索引值，否则返回-1

mystr.find(str, start=0, end=len(mystr))

##### <2>index：检测 str 是否包含在 mystr中，如果是返回开始的索引值，否则报一个异常.

mystr.index(str, start=0, end=len(mystr))

##### <3>**count**：返回 str在start和end之间 在 mystr里面出现的次数。

mystr.count(str, start=0, end=len(mystr))

##### <4>**replace**：把 mystr 中的 str1 替换成 str2,如果 count 指定，则替换不超过 count 次.

mystr.replace(str1, str2, mystr.count(str1))

##### <5>split：以 str 为分隔符切片 mystr，如果 maxsplit有指定值，则仅分隔 maxsplit 个子字符串。

mystr.split(str=" ", 2)

##### <6>**capitalize**：把字符串的第一个字符大写

mystr.capitalize()

##### <7>**title**：把字符串的每个单词首字母大写

##### <8>**startswith**：检查字符串是否是以 obj 开头, 是则返回 True，否则返回 False

mystr.startswith(obj)

##### <9>**endswith**：检查字符串是否以obj结束，如果是返回True,否则返回 False.

mystr.endswith(obj)

##### <10>**lower**：转换 mystr 中所有大写字符为小写

mystr.lower()

##### <11>**upper**：转换 mystr 中的小写字母为大写

mystr.upper()

##### <12>**ljust**：返回一个原字符串左对齐,并使用空格填充至长度 width 的新字符串

mystr.ljust(width)

##### <13>**rjust**：返回一个原字符串右对齐,并使用空格填充至长度 width 的新字符串

mystr.rjust(width)

##### <14>**center**：返回一个原字符串居中,并使用空格填充至长度 width 的新字符串

mystr.center(width)

##### <15>**lstrip**：删除 mystr 左边的空白字符

mystr.lstrip()

##### <16>**rstrip**：删除 mystr 字符串末尾的空白字符

mystr.rstrip()

##### <17>**strip**：删除mystr字符串两端的空白字符

##### <18>**rfind**：类似于 find()函数，不过是从右边开始查找.

mystr.rfind(str, start=0,end=len(mystr) )

##### <19>**rindex**：类似于 index()，不过是从右边开始.

mystr.rindex( str, start=0,end=len(mystr))

##### <20>**partition**：把mystr以str分割成三部分,str前，str和str后

mystr.partition(str)

##### <21>**rpartition**：类似于 partition()函数,不过是从右边开始.

mystr.rpartition(str)

##### <22>**splitlines**：按照行分隔，返回一个包含各行作为元素的列表

mystr.splitlines()

##### <23>**isalpha**：如果 mystr 所有字符都是字母 则返回 True,否则返回 False

mystr.isalpha()

##### <24>**isdigit**：如果 mystr 只包含数字则返回 True 否则返回 False.

mystr.isdigit()

##### <25>**isalnum**：如果 mystr 所有字符都是字母或数字则返回 True,否则返回 False

mystr.isalnum()

##### <26>**isspace**：如果 mystr 中只包含空格，则返回 True，否则返回 False.

mystr.isspace()

##### <27>**join**：mystr 中每个字符后面插入str,构造出一个新的字符串

mystr.join(str)