

nsd1909-py01-day02

数据类型分类

按存储模型分类

- 标量：数字、字符串
- 容器：列表、元组、字典

按访问模型分类

- 直接：数字
- 序列：字符串、列表、元组
- 映射：字典

按更新模型分类

- 不可变：数字、字符串、元组
- 可变：列表、字典

```
>>> s1 = 'python'
>>> s1[0] = 'P'      # 报错，因为字符串不可变
>>> s1 = 'Python'    # 可以重新赋值

>>> l1 = [10, 20]
>>> l2 = l1          # l1和l2都指向内存相同地址，并且内存是可变的
>>> l2.append(30)     # 修改l2，也会影响到l1
>>> l2
[10, 20, 30]
>>> l1
[10, 20, 30]
```

判断语句

除了表达式以外，各种各样的数据类型，也可以作为判断条件。对于数字来说，任何值为0的数字都表示False，非0表示True；其他数据类型，非空为True，空为False

```
n1 = 10
s1 = 'Python'

if n1 > 5:  # 判断条件为真才会执行相应的代码块
    print('yes')
    print('ok')

if 'to' not in s1:
    print('True')

if -0.0:
    print('任何值为0的数字都是假')

if 10:
    print('非0数字为真')

if ' ':
    print('空格字符也是一个字符，为真')

if '':
    print('字符串长度为0，为假')

if [10, 20]:
    print('非空列表，为真')

if (10, 20):
    print('非空元组，为真')

if {}:
    print('空字典，为假')

if None:  # None是关键字，相当于其他语言中的 null，表示真空
    print('None也为假')

if not 0:
    print('0为假，取反为真')
```

三元运算符

也叫条件表达式

```
>>> a = 10
>>> b = 20
>>> if a < b:
...     s1 = a
... else:
...     s1 = b
...
>>> s1
10
# 以上写法，可以简化为
>>> s2 = a if a < b else b
>>> s2
10
```

random模块

```
# 导入模块
>>> import random
>>> random.choice('abc') # 从字符串中随机选择一个字符
'c'
>>> random.choice('abc')
'b'
>>> random.choice(['aaa', 'bbb', 'ccc']) # 从列表中随机选取一项
'ccc'
>>> random.choice(('aaa', 'bbb', 'ccc')) # 从元组中随机选取一项
'aaa'
>>> random.randint(1, 100) # 生成100以内的随机整数，包括 1和100
86
```

shell颜色

```
[root@localhost nsd2019]# echo -e "\033[31;43;1mOK\033[0m"
# 30以上是前景色，40以上是背景色；1m表示加亮
# \033[0m关闭颜色。
```

循环

- 如果循环次数未知，推荐使用while循环
- 如果循环次数已知，推荐使用for循环

while循环

语法：

```
while 条件:
    如果条件为真，执行的语句块
```

while后面的条件，与if后面的条件用法一致。

break语句

循环一旦遇到break，就结束了。break后面的，在循环体内的代码也不再执行。

continue语句

- break语句，结束循环，程序继续执行循环后面的代码
- continue语句，结束本次循环，但是不终结循环
- 循环一旦遇到continue语句，将不再执行continue后面的代码，而回到判断条件处，如果条件为真，则继续循环

else语句

- 在python中，循环也有else，当循环正常结束（循环条件不满足，导致循环结束），else语句执行；如果循环被break中断，else语句将不再执行。

range函数

range函数潜在可以产生很多数字，但是不会立即生成，当向range取值时，它会生成一个。

range经常与for循环结合使用。

```
>>> range(10)
range(0, 10)
>>> for i in range(10):
...     print(i)
>>> list(range(10)) # 10是结束数字，产生的数字不包含结束数字。起始数字默认从 0开始
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> list(range(6, 10)) # 起始数字是6
[6, 7, 8, 9]
>>> list(range(1, 10, 2)) # 2是步长值
[1, 3, 5, 7, 9]
>>> list(range(10, 0, -1))
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

列表解析

也叫列表推导，用于生成列表

```
>>> [10] # 直接将数字放到列表中
[10]
>>> [5 + 5] # 将表达式的计算结果放到列表中
[10]
>>> [5 + 5 for i in range(10)] # 表达式计算10遍
[10, 10, 10, 10, 10, 10, 10, 10, 10, 10]
>>> [5 + i for i in range(1, 11)] # 表达式可以使用for的变量]
[6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
>>> [5 + i for i in range(10) if i % 2 == 1] # 满足判断条件时，才计算表达式
[6, 8, 10, 12, 14]
# 生成192.168.1.0网段中的所有地址
>>> ['192.168.1.%s' % i for i in range(1, 255)]
```