

nsd1912-py01-day02

python官方手册：<https://docs.python.org/zh-cn/3/library/index.html>

python数据类型分类

按存储模型分类

- 标量：不能包含其他数据。数字、字符串 'a123b'
- 容器：可以包含多种数据。列表、元组、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'
>>> l1 = [100, 200, 50]
>>> l1[0] = 10
>>> l1
[10, 200, 50]
>>> s1[0] = 'P'      # 报错，字符串不可变，不能原位修改
>>> s1 = 'Python'    # 可以重新赋值

>>> l1
[10, 200, 50]
>>> l2 = l1          # l1和l2指向相同的一段可变内存
>>> l2
[10, 200, 50]
>>> l2.append(30)
>>> l2
[10, 200, 50, 30]
>>> l1
[10, 200, 50, 30]
```

按访问模型分类

- 直接访问：数字
- 顺序访问：字符串、列表、元组
- 映射访问：字典

判断语句

- 语法结构

```
if 条件1:
    条件1为真时执行的语句块
elif 条件2:
    条件2为真时执行的语句块
elif 条件3:
    条件3为真时执行的语句块
... ..
else:
    以上条件都为假时，执行的语句块
```

- 判断是一个多分支结构，某一个分支条件为真执行相应的语块，其他分支不再判断
- 各种各样的数据类型也可以作为判断条件，任何值为0的数字都是假，非0为真；各种非空对象为真，空为假。关键字None为假。
- 三元运算符、条件表达式

```
a = 10
b = 20
if a < b:
    s1 = a
else:
    s1 = b
# 以上代码可以简写为
s2 = a if a < b else b
print(s1)
print(s2)
```

random模块

```
>>> import random
>>> random.randint(1, 100) # 生成1到100间的随机整数
31
>>> random.choice('abc')   # 随机选一项
'a'
>>> random.choice('abc')
'c'
>>> random.choice(['石头', '剪刀', '布'])
'石头'
>>> random.choice(['石头', '剪刀', '布'])
'布'
```

循环语句

- 如果循环次数未知，使用while循环
- 如果循环次数已知，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
```

- 循环是否可以结束，依赖于条件。条件为假后，循环语句就不再执行了
- while循环的条件和if判断的条件，使用方法相同
- break语句结束循环
- continue语句跳过本次循环剩余语句，回到while条件处执行
- else语句：当循环条件为假时，else语句才执行。如果循环是被break掉的，else语句也就不再执行了。

for循环

- for循环的使用的方法与shell一样
- range函数

```
# range只有一个参数时，表示结束数字，但是不包含，默认从 0开始  
>>> range(10)    # 得到一个range对象，潜在可以生成多个数字  
range(0, 10)  
>>> list(range(10)) # 取出range中所有的值，生成列表  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
>>> for i in range(10): # 使用时，不需要转列表  
...     print(i)  
>>> list(range(6, 11))  
[6, 7, 8, 9, 10]  
>>> list(range(1, 11, 2)) # 第三个数字是步长值  
[1, 3, 5, 7, 9]  
>>> list(range(10, 0, -1))  
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

- 列表解析：用于生成列表

```
>>> [10]
[10]
>>> [5 + 5]    # 将表达式计算结果存入列表
[10]
# 通过循环控制表达式计算的次数，每个表达式结果都存入列表
>>> [5 + 5 for i in range(1, 11)]
[10, 10, 10, 10, 10, 10, 10, 10, 10, 10]
# 表达式还可以用 for 循环中的变量
>>> [5 + i for i in range(1, 11)]
[6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
# 可以通过判断条件来过滤，哪些情况需要计算表达式
>>> [5 + i for i in range(1, 11) if i % 2 == 1]
[6, 8, 10, 12, 14]

# 生成192.168.1.1 - 192.168.1.254
>>> ['192.168.1.%s' % i for i in range(1, 255)]
>>> ['192.168.1.' + str(i) for i in range(1, 255)]

# 随机产生10个1到100的整数放到列表
>>> import random
>>> random.randint(1, 100)
26
>>> [random.randint(1, 100)]
[72]
>>> [random.randint(1, 100) for i in range(10)]
[79, 45, 46, 29, 47, 56, 92, 11, 7, 84]
```