

一、基本数据类型

（一）关键词（保留字）

1、基础词

is, None, global, not, with, as, or, yield, assert, import, in

2、流程词: True, False, Try, finally, except, continue, fo, if, elif, else, break, raise

3、函数词: def, return, lambda, pass

4、对象词: class

具体可以看官网的说明:

https://docs.python.org/3/reference/lexical_analysis.html#keywords

（二）变量（name=value）

1、注意用why/what/how来理解变量。

2、## 命名规则（不能用中文!、大小写、有意义的单词、不与关键词重复、连接符）全大写一般作为全局变量（常量）的命名，另外PYTHON大小写敏感。

```
In [13]: a=1
```

```
In [15]: a
```

```
Out[15]: 1
```

```
In [18]: type(a)
```

```
Out[18]: int
```

```
In [19]: 3/5
```

```
Out[19]: 0.6
```

```
In [20]: 3-5
```

```
Out[20]: -2
```

```
In [21]: 3*5
```

```
Out[21]: 15
```

Python是按顺序执行代码（括号、乘除、加减），如下：

```
(paul) C:\Users\翁子城\Documents\python>python
Python 3.7.1 (default, Oct 28 2018, 08:39:03) [MSC v.1912 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 3+5
8
>>> 3*5
15
>>> a=2
>>> a
2
>>> b=3
>>> b
3
>>> b=a
>>> b
2
>>>
```

```
In [35]: 3 // 5
```

```
Out[35]: 0
```

```
In [36]: 5 // 3
```

```
Out[36]: 1
```

```
In [37]: 5 % 3
```

```
Out[37]: 2
```

"/"表示整除。"%"求余

3、数字比较长的时候，为了更加清晰，赋值还能这样表示：

```
number=111_222_333_444
```

```
>>> number=111_222_333_444
```

```
>>> number
111222333444
```

数字还能这样表示：

```
>>> |
```

```
python.exe*[64]:952 « 180626[64] 3/3 [+1] NUM PRI: 78x25 (5,32) 25V 7288 86%
```

4、知乎上关于浮点数的讨论。

<https://www.zhihu.com/question/25457573>

<https://www.zhihu.com/question/65960911>

```
>>> 6.6%5
1.5999999999999996
>>>
python.exe*[64]:952      « 180626[64] 3/3  [+] NUM PRI  78x25   (5,34) 25V   7288  86%
```

(三) 内置类型，内置函数和标准库

1、随机函数

```
import random
```

```
random.randint(1,5) #1-5之间随机抽数，包括端点
```

```
# 示例, 21点, show help
```

```
import random
```

```
ratio = 0.5 # 这是一个比例，可以起到作弊效果
```

```
num1 = random.randint(3,10)
```

```
num2 = random.randint(3,10) * ratio
```

```
num3 = random.randint(3,10)
```

```
print(num1, int(num2), num3)
```

2、数学计算库

```
## round? (这个是内置函数，不用声明)
```

```
Round a number to a given precision in decimal digits. 四舍五入
```

```
round(number, ndigits=None)
```

```
In [31]: round?
```

```
In [33]: round(10.56, ndigits=1)
```

```
Out[33]: 10.6
```

```
In [35]: round(10.56)
```

```
Out[35]: 11
```

Signature: round(number, ndigits=None)

Docstring:

Round a number to a given precision in decimal digits.

The return value is an integer if ndigits is omitted or None. Otherwise the return value has the same type as the number. ndigits may be negative.

Type: builtin_function_or_method

```
## math (数学库)
```

```
import math 将数学库导入到我们这个大的编程环境中
```

```
math.floor(3.6) 地板~向下取值
```

```
math.ceil(3.6) 天花板~向上取值
```

```
In [41]: import math
```

```
In [42]: math.floor(3.6)
```

```
Out[42]: 3
```

```
In [43]: math.ceil(3.6)
```

```
Out[43]: 4
```

输入math,然后按Tab键可以下拉看到很多方法，忘记了用法就在后面加个问号执行，比如

math.pow?

```
In [46]: math.
```

```
In [38]: math.acosh
```

math.acos

math.acosh

math.asin

math.asinh

math.atan

math.atan2

math.atanh

math.ceil

math.copysign

math.cos

Traceback (most recent call last)

38-1a2d75e487de> in <module>

tin)

'builtin' is not defined

```
In [47]: math.pow?
```

```
In [38]: dir(builtin)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-38-1a2d75e487de> in <module>
----> 1 dir(builtin)

NameError: name 'builtin' is not defined
```

```
Signature: math.pow(x, y, /)
Docstring: Return x**y (x to the power of y).
Type:      builtin_function_or_method
```

这个方法计算平方，例如 $3**2$ ，得到的是9。

(三) 字符串，str/string

1、新建

name=input('name:') 就能输入名字。

name:paul

```
In [51]: name=input(' name:')
```

name:paul

```
In [52]: name
```

```
Out[52]: 'paul'
```

```
In [ ]:
```

2、字符串

用单引号或者双引号都行，也有用三双引号表示的"""

```
In [53]: """
sadasd
asdasdassad
sadsa
"""
```

```
Out[53]: '\nsadasd\nasdasdassad\nsadsa\n'
```

输出的\n表示换行，三双引号可以支持换行。

也可以第一行输入不下的时候补一个续第二行的内容

```
In [55]: """第一行\
第二行"""
```

```
Out[55]: '第一行第二行'
```

3、基本的数据结构list(列表)

my_list=[2,4,6,8] 每个列表内建议用相同一类型

type([]) 可以看类型

str_list[]

str_list.append('a') 往这个字符串列表最后增加a

```
In [60]: str_list=[]
```

```
In [61]: str_list
```

```
Out[61]: []
```

```
In [65]: str_list.append('b')
```

```
In [66]: str_list
```

```
Out[66]: ['a', 'b']
```

str_list.insert(1,11) 在第一个位置增加11。

my_list.count(11) 列表中共有几个11

```
Out[148]: [2, 11, 11, 4, 33, 6, 8, 9, 11]
```

```
In [149]: my_list[:3]
```

```
Out[149]: [2, 11, 11]
```

```
In [150]: my_list[3:]
```

```
Out[150]: [4, 33, 6, 8, 9, 11]
```

```
In [151]: my_list[1::2]
```

```
Out[151]: [11, 4, 6, 9]
```

```
In [81]: my_list
```

```
Out[81]: [2, 4, 6, 8, 9]
```

```
In [32]: arr[0], arr[1], arr[2]
```

```
Out[32]: ([1, 2, 3], [4, 5, 6], [7, 8, 9])
```

my_list[1::2] 隔两个取一个。