

# python

## Introduce

贵州大学

*hnzhang1@gzu.edu.cn*

March 12, 2019

# Overview

Introduction

basic

数据类型  
输入和输出

list and tuple

homework

Q&A



python 是一门比较容易入门、流行、解释型、面向对象、跨平台的语言。

网上学习资源推荐：

<https://www.icourse163.org/course/BIT-268001>

# install

1. 版本：python 3.x
2. 系统平台：根据自己的实际情况选择
  - Linux
  - Windows
  - Mac OS
3. IDE: pyCharm

## python 代码示例

```
1 import time
2 # 今天有专业网站建设课程吗？
3 weekDayName = time.strftime("%a",time.gmtime())
4 if weekDayName=='Wed':
5     print(" 今天可能有专业网站建设课。")
6 else:
7     print(" 今天可能没有专业网站建设课。")
```

## python 标准库

标准库为我们提供了大量可重用的代码。比如上一页示例代码“import time”中的 time 就是 python 标准库中的一个模块。更多模块及其提供的功能请查看官方文档：

<https://docs.python.org/3/library/index.html>

1. 数据类型
2. 输入和输出

# 数据类型

- 数值
  - 整数
  - 浮点数
- 字符串
- 布尔
  - True
  - False



## 与数值有关的基本运算

+, -, \*, /, \*\*, %

```
1 >>> 1+2
2 3
3 >>> 1-2
4 -1
5 >>> 1*2
6 2
7 >>> 1/2
8 0.5
9 >>> 2**3
10 8
11 >>> 3%2
12 1
13 >>>
```

## 常用的数学函数<sup>1</sup>

```
1 >>> import math
2 >>> dir(math)
3 [ 'acos', 'ceil', 'cos', 'exp', 'gcd', 'log',
4   'log10', 'log2', 'pi', 'pow', 'sqrt',...]
5
6 >>> help(math.gcd)
7 Help on built-in function gcd in module math:
8 gcd(x, y, /)
9     greatest common divisor of x and y
10 (END)
```

<sup>1</sup>python 标准库中的 math 模块提供了许多实用的数学计算函数。如果需要使用，则需要在代码前使用“import math”导入 math 模块，如果想知道 math 提供了哪些函数方法，可以使用 dir(math) 来查看，如果想知道具体方法如何使用，可以使用 help(math.abs) 来获取帮助。

## 常用的数学函数<sup>2</sup>

```
1 >>> help(abs)
2 Help on built-in function abs in module builtins:
3 abs(x, /)
4     Return the absolute value of the argument.
5 >>> help(round)
6 Help on built-in function round in module builtins:
7 round(number, ndigits=None)
8     Round a number to a given precision
9     in decimal digits.
10    The return value is an integer if ndigits is
11    omitted or None. Otherwise the return value
12    has the same type as the number.
13    ndigits may be negative.
```

<sup>2</sup>除 math 模块外, python 标准库中内置了一些常用的函数。

# 字符串

字符串就是由零个或多个字符组成的有限字符序列。注意单、双引号的使用。

```
1 >>> 'abc'
2 'abc'
3 >>> "abc"
4 'abc'
5 >>> 'a"b"c'
6 'a"b"c'
7 >>> "a'b'c"
8 "a'b'c"
9 >>> 'a\'b\'c'
10 "a'b'c"
```

## 字符串的拼接、索引和切片

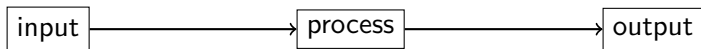
```
1 >>> a="guizhou"
2 >>> b="university"
3 >>> c=a+b
4 >>> c
5 'guizhouuniversity'
6 >>> c[0]
7 'g'
8 >>> c[-2]
9 't'
10 >>> c[3:]
11 'zhouuniversity'
12 >>> c[3:6]
13 'zho'
```

## 字符串的常用函数

```
1 >>> dir(str)
2 ['capitalize', 'endswith', 'isdigit', 'join',
3   'replace', 'split', 'splitlines', 'startswith']
4 >>> help(str.capitalize)
5
6 Help on method_descriptor:
7
8 capitalize(self, /)
9     Return a capitalized version of the string.
```

# 输入和输出

## 使用计算机求解问题的过程



## 输入输出函数

- input
- print

# input

```
1 >>> help(input)
2
3 Help on built-in function input in module builtins:
4
5 input(prompt=None, /)
6     Read a string from standard input.
7     The trailing newline is stripped.
8
9     The prompt string, if given,
10    is printed to standard output without a
11    trailing newline before reading input.
```



## input 典型用法

```
1 >>> name = input("What's your name?\n")
2 What's your name?
3 gzu
4 >>> print(name)
5 gzu
```

## print

```
1 >>> help(print)
2 Help on built-in function print in module builtins:
3 print(...)
4     print(value, ..., sep=' ', end='\n',
5           file=sys.stdout, flush=False)
6     Prints the values to a stream,
7     or to sys.stdout by default.
8
9     Optional keyword arguments:
10    sep:    string inserted between values,
11           default a space.
12    end:    string appended after the last value,
13           default a newline.
```

## print 典型用法

```
1 >>> print('the name is ',name)
2 the name is gzu
3 >>> print('the name is',name)
4 the name is gzu
5 >>> print('the name is',name,end='.')
6 the name is gzu.>>> print('the name is',name,end='.\n')
7 the name is gzu.
```

## input 补充 1

Q：要求用户输入两个数字，然后计算这两个数字的和。

```
1 >>> a = input("1st number:")  
2 1st number:1  
3 >>> b = input("2nd number:")  
4 2nd number:2  
5 >>> sum = a + b
```

## input 补充 2

### 字符串与数值类型间的转换

**int(x)** Convert a number or string to an integer, or return 0 if no arguments are given. If x is a number, return x.\_\_int\_\_(). For floating point numbers, this truncates towards zero.

**float(x)** Convert a string or number to a floating point number, if possible.

**eval(x)**<sup>3</sup> Evaluate the given source.

**str(x)** Create a new string object from the given object.

---

<sup>3</sup>eval() is a dangerous function, eg: eval(os.system('rm -R \*'))



# list(列表) and tuple(元组)

## list

list 是 python 中的有序序列，可以存储任何类型的元素，每个元素以逗号分隔。使用中括号将元素包括起来即可创建一个 list。

```
1 >>> mList = ['cs','is','dm',798,3.14]
2 >>> numList = [2,8,8,9,1,2,3,9,7,6,5,4]
```

函数/方法	说明	函数/方法	说明
len(mList)	返回元素个数	sum(numList)	求和 <sup>4</sup>
max(numList)	返回最大值	min(numList)	返回最小值 <sup>5</sup>
mList.insert(1,'m')	指定位置插入	del mList[1]	删除指定元素
numList.sort()	排序	numList.reverse()	逆序

<sup>4</sup>仅针对数字元素组成的列表

<sup>5</sup>仅针对相同元素组成的列表



## split 和 join

split 和 join 是作用相反的一对非常有用的方法。

**split** 将一个字符串切分成其子串组成的列表

**join** 将一个字符串列表变成一个字符串

```
1 >>> s="gzu,zzu,cqu,tju"
2 >>> sList = s.split(",")
3 >>> sList
4 ['gzu', 'zzu', 'cqu', 'tju']
5 >>> ns=''.join(sList)
6 >>> ns
7 'gzuzzucqutju'
8 >>> ns=','.join(sList)
9 >>> ns
10 'gzu,zzu,cqu,tju'
```

## split

```
1 >>> help(str.split)
2 Help on method_descriptor:
3
4 split(self, /, sep=None, maxsplit=-1)
5     Return a list of the words in the string,
6     using sep as the delimiter string.
7
8     sep
9         The delimiter according which to split the string.
10        None (the default value) means split
11        according to any whitespace,
12        and discard empty strings from the result.
13    maxsplit
14        Maximum number of splits to do.
15        -1 (the default value) means no limit.
```

## join

```
1 >>> help(str.join)
2 Help on method_descriptor:
3
4 join(self, iterable, /)
5     Concatenate any number of strings.
6
7     The string whose method is called
8     is inserted in between each given string.
9     The result is returned as a new string.
10
11 Example: '.'.join(['ab', 'pq', 'rs']) -> 'ab.pq.rs'
```

## tuple

tuple 与 list 类似，不同之处在于不能直接修改 tuple 中的元素，即不能使用 del、insert 方法。使用括号将元素包括起来即可创建一个 tuple，不使用括号也可以。

```
1 >>> mTuple=('cs','mis',1,3.14)
2 >>> nTuple=1,2,3,4,6
3 >>> (x,y,z)=(3,4,5)
4 >>>x,y,z=3,4,5
5 >>> x
6 3
7 >>> x,y = y,x
8 >>> x
9 4
```

## 嵌套列表

列表中的元素也可以是另外一个列表，或者是一个元组。

```
1 >>> comList=[("China",86),("USA",1)]
2 >>> comList[0]
3 ('China', 86)
4 >>> comList[1][1]
5 1
```

## 不可变和可变对象

对象是一个包含方法和属性的实体。数值、字符串、列表和元组都是对象。

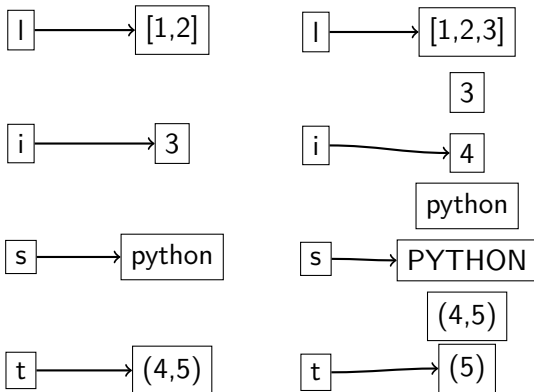
当一条赋值语句执行完之后，“=”右边的值变成了内存中的一个对象，“=”左边的变量引用（或者说指向）内存中的那个对象。

当要改变一个**列表**中的元素值时，在列表**原来的位置**进行更改。

而要改变**数值、字符串和元组**时，python 会分配一个**新的内存位置**存储新值，并将变量引用（或指向）到新的对象上。

## 不可变和可变对象内存示意

```
1 >>> l=[1,2]
2 >>> i=3
3 >>> s="python"
4 >>> t=(4,5)
5 >>> l.append(3)
6 >>> i+=1
7 >>> s=s.upper()
8 >>> t=t[1:]
```



# Homework

1. 计算  $x^y$ ,  $x, y$  的值使用 `input` 获取。
2. 从控制台输入被除数和除数, 求余数。



# Q&A