变量

# message变量

message = "Hello World"

# 多变量赋值

x, y, z = 0, 9, 30

常量，Python没有内置常量类型，通常通过全大写来指出将某个变量视为常量，其值应始终不变。

# 常量，常量名全大写

MAX\_CONNECTIONS = 5000

字符串，使用单引号或双引号括起来的一系列字符。

* title()方法：以首字母大写的方式显示每个单词。
* upper()方法：将字符串全部大写。
* lower()方法：将字符串全部小写。
* lstrip()/rstrip()/strip()方法：剔除字符串左侧/右侧/两侧空白。
* f字符串，format()方法的语法糖。

first\_name = "adA"

last\_name = "LoveLace"

name = f"{first\_name} {last\_name}"

print(name) # adA LoveLace

print("{} {}".format(first\_name, last\_name)) # adA LoveLace

print(name.title()) # Ada Lovelace

print(name.upper()) # ADA LOVELACE

print(name.lower()) # ada lovelace

pl = " python "

print(f"'{pl}'") # ' python '

print(f"'{pl.lstrip()}'") # 'python '

print(f"'{pl.rstrip()}'") # ' python'

print(f"'{pl.strip()}'") # 'python'

整数，浮点数

* 加(+)、减(-)、乘(\*)、除(/)、乘方(\*\*)。
* 带小数点的数称为浮点数。
* 任意两个数相除，结果总是浮点数。
* 无论哪种运算，只要操作数有浮点数，结果总是浮点数。
* 可以使用下划线将数字分组。

num1 = 4

num2 = 2

print(num1 + num2) # 6

print(num1 - num2) # 2

print(num1 \* num2) # 8

print(num1 / num2) # 2.0

print(num1 \*\* num2) # 16

print(4.0 + 2) # 6.0

big\_num = 1\_000\_000\_000

print(big\_num) # 1000000000

列表，由一系列按特定顺序排列的元素组成。使用方括号（[]）表示列表，逗号分割列表中的元素。

* len()函数：获取列表的长度，即列表中包含元素的数量。
* 通过索引($list[$index])可以访问和修改列表中的元素，索引从0开始，-1表示最后一个元素。

bicycles = ["trek", "cannondale", "redline", "specialized"]

print(bicycles) # ['trek', 'cannondale', 'redline', 'specialized']

# len()函数：列表的长度

print(len(bicycles)) # 4

# 通过索引访问列表元素, -1表示列表最后一个元素

print(bicycles[1]) # cannondale

print(bicycles[-1]) # specialized

print(f"My first bicycle was a {bicycles[0].title()}.") # My first bicycle was a Trek.

# 通过索引修改列表索引位置的元素

bicycles[0] = "ducati"

print(bicycles) # ['ducati', 'cannondale', 'redline', 'specialized']

* append()方法：在列表末尾添加元素。
* insert($index, $item)方法：在列表指定索引位置添加元素，索引右侧元素都右移一个位置。

# append()方法：在列表末尾添加元素

motorcycles = []

motorcycles.append("honda")

motorcycles.append("yamaha")

motorcycles.append("suzuki")

print(motorcycles) # ['honda', 'yamaha', 'suzuki']

# insert($index, $item)方法：在列表指定索引位置添加元素

motorcycles.insert(1, "ducati")

print(motorcycles) # ['honda', 'ducati', 'yamaha', 'suzuki']

* del $list\_name[$index]语句：删除元素，被删除元素的右侧元素都左移一个位置。
* pop()方法：删除列表末尾元素，并返回。列表就像一个栈，删除列表末尾的元素相当于弹出栈顶元素。
* pop($index)方法：删除列表指定索引的元素，并返回。索引右侧的元素都左移一个位置。
* remove($value)方法：删除列表中第一个指定的值。

motorcycles = ["honda", "ducati", "yamaha", "suzuki"]

# del语句：删除元素，被删除元素的右侧元素都左移一个位置

del motorcycles[1]

print(motorcycles) # ['honda', 'yamaha', 'suzuki']

# pop()方法：删除列表末尾元素，并返回

popped\_motorcycle = motorcycles.pop()

print(motorcycles) # ['honda', 'yamaha']

print(popped\_motorcycle) # suzuki

# pop($index)方法：删除列表指定索引的元素，并返回。

motorcycles = ["honda", "ducati", "yamaha", "suzuki"]

print(motorcycles) # ['honda', 'ducati', 'yamaha', 'suzuki']

popped\_motorcycle = motorcycles.pop(1)

print(motorcycles) # ['honda', 'yamaha', 'suzuki']

print(popped\_motorcycle) # ducati

# remove($value)方法：删除列表中第一个指定的值。

names = ["remilia", "flandre", "cirno", "remilia"]

print(names) # ['remilia', 'flandre', 'cirno', 'remilia']

names.remove("remilia")

print(names) # ['flandre', 'cirno', 'remilia']

* sort(reverse=True|False)方法：按字母顺序对列表元素永久排序。reverse: False(默认值)正序，True倒序。
* sorted(reverse=True|False)函数：按字母顺序对列表元素临时排序。reverse: False(默认值)正序，True倒序。
* reverse()方法：永久反转列表中元素的排列顺序。

# sort(reverse=True|False)方法：对列表中元素按字母顺序永久排序

names = ["remilia", "flandre", "cirno", "sakura"]

print(names) # ['remilia', 'flandre', 'cirno', 'sakura']

names.sort()

print(names) # ['cirno', 'flandre', 'remilia', 'sakura']

names.sort(reverse=True)

print(names) # ['sakura', 'remilia', 'flandre', 'cirno']

# sorted(reverse=True|False)函数：对列表中元素按字母顺序临时排序

names = ["remilia", "flandre", "cirno", "sakura"]

print(names) # ['remilia', 'flandre', 'cirno', 'sakura']

sortedNames = sorted(names)

print(names) # ['remilia', 'flandre', 'cirno', 'sakura']

print(sortedNames) # ['cirno', 'flandre', 'remilia', 'sakura']

# reverse()方法：永久反转列表中元素的排列顺序

names = ["remilia", "flandre", "cirno", "sakura"]

print(names) # ['remilia', 'flandre', 'cirno', 'sakura']

names.reverse()

print(names) # ['sakura', 'cirno', 'flandre', 'remilia']

* 使用for循环可以遍历列表中的元素：`for $item in $list\_of\_items:`。

magicians = ["alice", "david", "carolina"]

# for循环：`for $item in $list\_of\_items:`

for magician in magicians: # Alice, David, Carolina,

print(f"{magician.title()}", end=", ")

* range($start\_include, $end\_exclude, $step)函数：创建一系列数字。$start\_include默认值0，$step默认值1。
* list()函数，将range()函数的结果转换为列表。
* min()、max()、sum()函数：数字列表最小值、最大值和总和。

print(range(0, 6)) # range(0, 6)

print(list(range(0, 6))) # [0, 1, 2, 3, 4, 5]

print(list(range(0, 6, 2))) # [0, 2, 4]

digits = list(range(6))

# [0, 1, 2, 3, 4, 5] ==> min=0, max=5, sum=15

print(f"{digits} ==> min={min(digits)}, max={max(digits)}, sum={sum(digits)}")

* 列表解析

squares\_1 = []

for v in range(11):

squares\_1.append(v \*\* 2)

# 列表解析

squares\_2 = [v \*\* 2 for v in range(11)]

print(squares\_1) # [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

print(squares\_2) # [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

切片，列表的部分元素。

* $list\_name[${from\_include}:${to\_exclude}:${step}]，$from\_include默认值0，$to\_exclude默认值len($list\_name)，$step默认值1。
* 切片的遍历通过for循环。
* 通过$list\_name[:]可以复制整个列表，复制后的列表是原列表的一个副本，对原列表或副本的修改互不影响。

names = ["Remilia", "Flandre", "Cirno", "Sakura"]

print(names) # ['Remilia', 'Flandre', 'Cirno', 'Sakura']

print(names[0:3:2]) # ['Remilia', 'Cirno']

print(names[:3]) # ['Remilia', 'Flandre', 'Cirno']

print(names[3:]) # ['Sakura']

for name in names[:]: # Remilia,Flandre,Cirno,Sakura,

print(name, end=",")

print()

# 复制列表副本

fork\_names = names[:]

del names[-1]

del fork\_names[0]

print(names) # ['Remilia', 'Flandre', 'Cirno']

print(fork\_names) # ['Flandre', 'Cirno', 'Sakura']

元组，不可变的列表，使用圆括号标识。

* 可以通过索引访问元组中的元素。
* 通过for循环遍历元组的所有元素。

tuple = (200, 50, 100)

print(tuple[0]) # 200

print(tuple[1]) # 50

print(tuple[2]) # 100

for num in tuple: # 200,50,100,

print(num, end=",")

if语句

* 布尔表达式：结果为布尔值（True/False）。
* 比较运算符：。
* 逻辑运算符：。
* 检查特定值是否在/不在列表中：in / not in。
* 判断列表是否为空：`if $list: `等价与`if len($list) > 0: `。
* if, if-else, if-elif-else。

cars = ["audi", "bmw", "subaru", "toyota"]

for car in cars: # Audi, BMW, Subaru, toyota,

if car.lower() == "bmw":

print(car.upper(), end=", ")

elif car.lower() == "toyota":

print(car.lower(), end=", ")

else:

print(car.title(), end=", ")

print()

age = 18

print(age < 16 and age >= 18) # False and True => False

print(age < 16 or age >= 18) # False or True => True

names = []

if names: # names list is EMPTY.

print(names)

else:

print("names list is EMPTY.")

字典，一系列键-值对，使用花括号（{}）标识。每个键都与一个值相关联，键和值之间用冒号（:）分隔，键-值对之间用逗号（,）分隔。

* $dict[$key]：获取与键相关联的值。
* $dict[$key] = $value：向字典中添加一个键-值对，如果键已经存在，则替换成新的值。
* del $dict [$key]语句：将键对应的键-值对从字典中删除。

alien = {"color": "green", "points": 5}

print(alien) # {'color': 'green', 'points': 5}

print(alien["color"]) # green

print(alien["points"]) # 5

alien["x\_position"] = 5

alien["y\_position"] = 10

print(alien) # {'color': 'green', 'points': 5, 'x\_position': 5, 'y\_position': 10}

alien["color"] = "red"

print(alien) # {'color': 'red', 'points': 5, 'x\_position': 5, 'y\_position': 10}

del alien["color"]

print(alien) # {'points': 5, 'x\_position': 5, 'y\_position': 10}

del alien["x\_position"]

print(alien) # {'points': 5, 'y\_position': 10}

* `for $key, $value in {$dict}.items(): ` 遍历字典中的所有键-值对，字典的items()方法返回一个键-值对列表。
* `for $key in {$dict}.keys(): ` 遍历字典的所有键，字典的keys()方法返回一个列表，包含字典中的所有键。
* `for $value in {$dict}.values(): ` 遍历字典的所有值，字典的values()方法返回一个列表，包含字典中的所有值。
* set()函数，返回一个不包含重复元素的集合。

favorite\_languages = {

"Sakura": "Java",

"Remilia": "Python",

"Cirno": "Go",

"Flandre": "Python"

}

""" items()

Sakura favorite language is Java.

Remilia favorite language is Python.

Cirno favorite language is Go.

Flandre favorite language is Python.

"""

for name, language in favorite\_languages.items():

print(f"{name.title()} favorite language is {language}.")

""" keys()

Sakura Remilia Cirno Flandre

"""

for name in favorite\_languages.keys():

print(name.title(), end=" ")

print()

""" 隐式调用了keys()方法

Sakura Remilia Cirno Flandre

"""

for name in favorite\_languages:

print(name.title(), end=" ")

print()

""" values()

dict\_values(['Java', 'Python', 'Go', 'Python'])

"""

print(favorite\_languages.values())

print(list(favorite\_languages.values())) # ['Java', 'Python', 'Go', 'Python']

print(set(favorite\_languages.values())) # {'Java', 'Go', 'Python'}