

初识 Python

作者: [lionlinglong](#)

原文链接: <https://ld246.com/article/1632885618815>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>index count replace split startswith endswith upper lower isalpha isdigit title isalnum</p>
>
<p>列表中最大值:

max(list)</p>
<p>python 中的内置函数:

max()

min()

del()

len()

sum()

</p>
<p>用于任何数据类型</p>
<p>原样输出: print

单行注释:

#注释内容 快捷键: ctrl+

多行注释:

"""注释内容""" 也可以用单引号 '''注释内容'''

print(type(a))输出 a 的类型

</p>
<p>输入函数

a=input("提示内容") 赋给 a 值 是字符串 用强制转换将 a 变成其他类型 如: int(a) 变成整型

</p>
<p>%d 整型

</p>
<p>%f 浮点型

</p>
<p>%s 字符串型

</p>
<p>%c 字符型

</p>
<p>a=[3,1,2,5,4]

</p>
<p>a.sort(reverse=True)

</p>
<p>print(a) 倒序</p>
<p>字符串: str

</p>
<p>数字类型: 整型 int 浮点型 float

</p>
<p>布尔类型: 1=true 0=false

</p>
<p>列表: list = [1,2,3]

</p>
<p>元祖: tuple = (1,2)

</p>
<p>字典: dict = {"name": "123", "age": 18}

</p>
<p>集合: set = {1,2,3,4}

</p>
<p>if 条件一:

</p>
<p>print()

</p>
<p>elif 条件二:

</p>
<p>print()

</p>
<p>else :

</p>
<p>print()</p>
<p>name="abcdef"

</p>
<p>输出下标为 0 print(name[0])</p>
<p>切片: 范围是左闭右开!!!

</p>
<p>print(name[0:3]) 结果: abc

</p>
<p>print(name[0:5:2]) 结果: ace

</p>
<p>print(name[0:-1:2]) 结果: ace

</p>
<p>print(name[5:0:-2]) 结果: fdb

</p>
<p>print(name[5:0:-1]) 结果: fedcb

</p>
<p>print(name[:3]) 结果: abc</p>
<p>字符串的常见操作:

</p>
<p>my="hello world hello python"

</p>
<p>1、find():检测字符串里是否包含你要查询的字符串

</p>
<p>例如:

</p>
<p>print(my.find("python"))

</p>
<p>如果存在返回开始的索引值 (也就是下标)

</p>

不存在返回值为-1

2、index():检测你设定的字符串长度里是否存在你要查询的字符串

例如:

```
print(my.index("py",0,25))
```

如果存在返回开始的索引值, 否则报错, index 里的第一个数字参数是起始, 后一个是终止的索引值-

3、count:查询出现的次数

```
a=len("123") a 的值为该字符串的长度 a=3
```

```
print(my.count("l"))
```

结果为 5

```
my_str.count(str, 0, len(mystr))
```

4、replace() 替换

```
my_str.replace("l","r",2) 将 l 替换为 r 两次
```

5、split() 切片

以 str 为分隔符切片 mystr,如果 maxsplit 有指定值, 则仅分割 maxsplit 个字符串

```
my_str.split(str=" ",2)
```

```
my_str.split(" ") 以空格为切片条件
```

结果为: ['hello','world','hello','python']为列表结构

```
str="qaqwrwrwee"
```

```
print(str.split("w",2))
```

结果为: ['qaq','r','rwee'] 2 表示切两次 以 w 为切点 结果没有切点

6、startswith()

检查字符串是否以()开头 是返回 True 不是返回 False

```
my_str.startswith("hello") 结果为: True
```

7、endswith() 查询是否以什么什么结尾

4、upper(): 转换 my 中的小写字母为大写

```
print(my.upper())
```

5、lower():转换 my 中的大写字母为小写

```
print(my.lower())
```

6、title():把字符串的每个单词首字母大写

```
print(my.title())
```

7、capitalize():把字符串的第一个字符大写

```
print(my.capitalize())
```

8、splitlines():按照行分隔, 返回一个包含各行作为元素的列表

例如:

```
my_name="hello\nworld"
```

```
print(my_name)
```

结果为:

```
hello
```

```
world
```

```
print(my_name.splitlines())
```

结果为:

```
["hello","world"]
```

9、isalpha():如果 my 所有字符都是字母 则返回 True,否则返回 False

```
print(my.isalpha())
```

结果为: True

10、isdigit(): 如果 my 只包含数字则返回 True 否则返回 False

```
print(my.isdigit())
```

结果为: False

11、isalnum():如果 my 所有字符都是字母或数字则返回 True,否则返回 False

```
print(my.isalnum())
```

结果为 True

12、isspace():如果 my 中只包含空格, 则返回 True,否则返回 False

```
print(my.isspace())
```

结果为: False

<p>13、rfind()
从右往左找你要查找的字符串 找到后返回下标值
a=rfind('.') 用来取文件的后缀名</p>

<p>14、join
用来将列表里的元素连接变为字符串
一行代码实现九九乘法表
print('.')join([1,2])
结果为'1.2'</p>

<p>列表:
增加 (append,extend,insert)
append:
A=[1,2,3]
A.append("4")
结果为: A=[1,2,3,"4"]</p>

<p>extend:
a=[1,2] b=[3,4]
a.extend(b)
结果为 a=[1,2,3,4]
a.append(b)
结果为 a=[1,2,[3,4]]</p>

<p>insert:
a=[0,1,2]
a.insert(1,3)
结果为 a=[0,3,1,2]</p>

<p>修改元素:
a=[1,2,3]
a[0]=3
结果为: a=[3,2,3]</p>

<p>查找元素:
1、in 判断存在
name="123456789"
find="123"
if find in name:
print("存在")
else:
print("不存在")
2、not in 判断不存在</p>

<p>3、index 和 count 和在字符串里的应用一样</p>

<p>删除元素:
list=[1,2,3]
1、del list[1]
2、list.pop()删除最后一个元素
3、list.remove(1)</p>

<p>排序
list.sort() 从小到大排序
list.sort(reverse=True)从大到小排序
list.reverse()反转</p>

<p>列表的嵌套
list=[1,[2,3],4]
list[1].insert(1,5)结果为 list[1,[2,5,3],4]</p>

<p>字典:
字典的每个元素由 2 部分组成, 键:值。例如 'name':'班长', 'name'为键, '班长'为值
info={"name":'师玲珑',"age":18}


```
print(info["name"]) 结果为: 师玲珑</p>
<p>修改数据: <br>
info["age"]=20</p>
<p>删除数据<br>
del info["age"]</p>
<p>info.clear()删除所有数据</p>
<p>如果 info 中不存在'age'这个键, 我就让他返回默认值 18<br>
age=info.get('age',18)<br>
print(age)</p>
<p>len()测量字典中, 键值对的个数<br>
print(info.keys()) 返回一个包含字典所有的 key 列表<br>
结果为: info_keys(['name','age','sex'])</p>
<p>print(info.values()) 返回一个包含字典所有的 value 列表<br>
结果为: info_values([18,20,'女'])</p>
<p>print(info.items()) 返回一个包含字典所有的键值对列表<br>
结果为: info_values([('name','age','sex'),(18,20,'女')])</p>
<p>遍历字典的 key 210 480 690<br>
for key in info.keys():<br>
print(key)</p>
<p>遍历字典的 values<br>
for value in info.values():<br>
print(value)</p>
<p>遍历字典的 items<br>
for item in info.items():<br>
print(item)<br>
或者<br>
for key,value in info.items()<br>
print(key,value)</p>
<p>如何实现带下标的遍历<br>
chars=[1,2,3,4]<br>
for i,chr in enumerate(chars):<br>
print(i,chr)<br>
结果为: <br>
0 1<br>
1 2<br>
2 3<br>
3 4</p>
<p>
```

集合:

集合是无序的, 集合中的元素是唯一的, 集合一般用于元组或者列表中的元素去重。</p>

<p>添加元素

add:


```
set1={1,2,3}<br>
```

```
set1.add(8)<br>
```

```
print(set1)<br>
```

结果为 1,2,3,8</p>

<p>update:


```
set1={1,2,3}<br>
```

```
set1.update("abcd")<br>
```

```
print(set1)<br>
```

结果为: 1,2,3,a,b,c,d</p>

<p>删除元素


```
set1.remove(1)如果没有报错</p>
```

set1.discard(1) 如果没有不执行
set1.pop()是随机删除集合中的元素
交集:
set1={1,2,3}
set2={3,4}
new_set=set1&set2
print(new_set) 结果为{3}
并集:
new_set=set1|set2
print(new_set) 结果为{1,2,3,4}

公共方法:

运算符	python 表达式	结果	描述	支持的数据类型
-----	------------	----	----	---------

[1,2]+[3,4]	[1,2,3,4]	合并	字符串.列表.元祖
-------------	-----------	----	-----------

[1]*4	[1,1,1,1]	复制	字符串.列表.元祖
-------	-----------	----	-----------

</code></pre>

in	3 in (1,2,3)	True	元素是否存在	同上 + 字典
----	--------------	------	--------	---------

not in	4 not in(1,2,3)	True	元素是否不存在	同上
--------	-----------------	------	---------	----

