

## 文章目录

- [1 标识符](#)
- [2 保留字](#)
- [3 注释](#)
- [4 多行语句](#)
- [5 print 输出](#)
- [6 import 与 from...import](#)

## 1 标识符

- 第一个字符必须是字母表中 **字母** 或下划线 `_`
- 标识符的其他部分由 **字母、数字和下划线** 组成。
- 标识符 **区分大小写**。

## 2 保留字

保留字即关键字，我们不能把它们用作任何标识符名称

```
import keyword
print(keyword.kwlist)
'''
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def',
'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import',
'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try',
'while', 'with', 'yield']
'''
```

## 3 注释

Python 中单行注释使用 `#`，多行注释使用三个单引号 (`'''`) 或三个双引号 (`"""`)

**注释的作用：**使用自己熟悉的语言，在程序中对某些代码进行标注说明，增强程序的可读性

**如下所示：**

```
# 单行注释 （为了保证代码的可读性，`#` 后面建议先添加一个空格，然后再编写相应的说明文字）
```

```
'''  
多行注释  
多行注释  
'''
```

```
"""  
多行注释  
多行注释  
"""
```

## 4 多行语句

Python 通常是一行写完一条语句，但如果语句很长，我们可以使用反斜杠 `\` 来实现多行语句，例如：

```
a = b + \  
    c + \  
    d  
# 等价于  
a = b + c + d
```

在 `[]`, `{}`, `()` 中的多行语句，不需要使用反斜杠 `\`，例如：

```
temp = ['a', 'b', 'c',  
        'd', 'e']
```

## 5 print 输出

`print` 默认输出是换行的，如果要实现不换行需要在变量末尾加上 `end=""`：

```
a = 1  
b = 2  
print(a)  
print(b)  
  
print('-----')  
# 不换行输出  
print(a, end=" ")  
print(b, end=" ")
```

输出结果如下：

```
1
2
-----
1 2
```

## 6 import 与 from...import

---

在 python 用 `import` 或者 `from...import` 来导入相应的模块。

将整个模块(somemodule)导入，格式为： `import somemodule`

从某个模块中导入某个函数,格式为： `from somemodule import somefunction`

从某个模块中导入多个函数,格式为： `from somemodule import firstfunc, secondfunc, thirdfunc`

将某个模块中的全部函数导入，格式为： `from somemodule import *`