

Python 基础

1. Python语法介绍

第一个程序 hello world

In [1]:

```
print("hello world")
```

hello world

Python语言中的注释

注释是编程语言中重要的一环，在Python中单行注释使用#实现，多行注释使用三引号实现

In [2]:

```
print("hello world") # 使用 “#” 实现单行注释
```

hello world

In [4]:

```
"""  
使用三引号  
实现多行注释  
"""  
print("hello world")
```

hello world

工具包导入

Python之所以简单，一部分取决于其丰富的工具包（标准库和第三方库），导入工具包可以使用import或者from ... import ...，所导入的第三方模块是需要安装的

In [6]:

```
import os # 导入os模块  
os.getcwd() # 查看当前路径
```

Out[6]:

'D:\\python project'

In [9]:

```
from os import getcwd  
getcwd()
```

Out[9]:

'D:\\python project'

如果某个工具包名较为复杂可以通过as起别名

In [1]:

```
# 导入画图模块
import matplotlib.pyplot as plt
```

变量的使用

变量使用时需要一个名字-变量名（标识符），这个名字由字母数字下划线组成，并且数字不能开头，不能和关键字重名（已经被定义好拥有特定功能的标识符）。

In [6]:

```
a = 1 # "="的作用为赋值
a1_ = 1
_1 = 1
1a = 1 # 错误的变量名
```

File "<ipython-input-6-73223ba4b3f9>", line 4

```
1a = 1 # 错误的变量名
```

^

SyntaxError: invalid syntax

查看Python中的关键字有哪些

In [12]:

```
import keyword
keyword.kwlist
```

Out[12]:

```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 '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']
```

In [15]:

```
help("keywords")
```

Here is a list of the Python keywords. Enter any keyword to get more help.

False	class	from	or
None	continue	global	pass
True	def	if	raise
and	del	import	return
as	elif	in	try
assert	else	is	while
async	except	lambda	with
await	finally	nonlocal	yield
break	for	not	

In [16]:

```
def = 1 # 使用关键字作为变量名
```

```
File "<ipython-input-16-09f57f03a914>", line 1
```

```
def = 1
    ^
```

SyntaxError: invalid syntax

Python中的变量根据使用范围分为全局变量:定义在模块内都可以使用；和局部变量：定义在方法和函数中，只能在其内部使用

In [3]:

```
a = 1 # 全局变量
```

```
def func(): # 定义一个函数
```

```
    b = 3 # 在函数内部定义局部变量
```

```
c = 2
```

```
print(a+c)
```

```
print(a+b)
```

3

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-3-ad3ad6c8508d> in <module>
      7
      8 print(a+c)
---->  9 print(a+b)
```

NameError: name 'b' is not defined

b定义在函数内部是局部变量，外部使用会使程序异常

Python程序的执行顺序

Python语句自上向下执行

In [5]:

```
a = 0
print(a+b)
b = 1
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-5-3e1a61996624> in <module>
      1 a = 0
----> 2 print(a+b)
      3 b = 1

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

Python中的缩进

Python区分代码块使用的是缩进，缩进可以为n (>=1),全局要保持一致

In [11]:

```
def func1():
    a = 1
def func2():
    b = 2
```

Python基础函数

In [17]:

```
help(print) # help函数查看帮助信息
```

Help on built-in function print in module builtins:

```
print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file: a file-like object (stream); defaults to the current sys.stdout.
    sep:   string inserted between values, default a space.
    end:   string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.
```

In [19]:

```
dir(print) # 查看函数的属性和具有的方法
```

Out[19]:

```
['_call__',
 '_class__',
 '_delattr__',
 '_dir__',
 '_doc__',
 '_eq__',
 '_format__',
 '_ge__',
 '_getattribute__',
 '_gt__',
 '_hash__',
 '_init__',
 '_init_subclass__',
 '_le__',
 '_lt__',
 '_module__',
 '_name__',
 '_ne__',
 '_new__',
 '_qualname__',
 '_reduce__']
```

```
'__reduce_ex__',
'__repr__',
'__self__',
'__setattr__',
'__sizeof__',
'__str__',
'__subclasshook__',
'__text_signature__']
```

In [20]:

```
a = 1
print(id(a)) # id 查看内存地址
```

140734273581328

In [21]:

```
type(a) # 查看数据类型
```

Out[21]:

int

In [22]:

```
del(a) # 删除对象
print(a)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-22-e54ced672327> in <module>
      1 del(a) # 删除对象
----> 2 print(a)
```

NameError: name 'a' is not defined

In [23]:

```
s = input("input: ") # 接受输入, 输入类型为字符串
type(s)
```

Out[23]:

str

In [24]:

```
len("hello") # 查看对象长度
```

Out[24]:

5

In []: