

基本输入输出

等号赋值

赋值类型	描述	示例
基本赋值	使用等号 (=) 进行赋值。	<code>x = 10</code>
同一个值给多个变量	可以使用一个值来赋给多个变量。	<code>x = y = z = 10</code>
多重赋值	可以同时给多个变量赋多个值。	<code>x, y, z = 1, 2, 3</code>
使用下划线的赋值	当你不关心某个值时，可以使用下划线 (_) 作为“丢弃”变量。	<code>x, _ = 1, 2</code>

命名规则

◆ 以下名字有特殊意义，不能用作变量名

不能作为变量名的单词					
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	

命名规则

◆ 除此之外，变量名不能以数字和特殊字符开头

- 1 # 以下内容为非法标识符
- 2 4word # 以数字开头
- 3 try # Python保留字
- 4 \$lady # 特殊字符

命名规则

◆ 建议命名规则1_下划线法

```
1 user_name = "Alice"  
2 account_balance = 1000.50
```

◆ 建议命名规则2_驼峰法

```
1 userName = "Alice"  
2 accountBalance = 1000.50
```

注释

◆ 使用 # 进行单行注释

◆ 使用三引号进行多行注释

```
1 a = 1  
2 # b = 2  
3 c = 3  
4 '''  
5 c = 4  
6 e = 5  
7 '''  
8 f = 6  
9 """"  
10 g = 7  
11 h = 8  
12 """"  
13 i = 9
```

屏幕输出

参数	描述	示例	结果
obj	可以打印一个对象	<code>print("AI")</code>	AI
obj	需要打印的对象，可以是多个	<code>print("Hello", "World")</code>	Hello World
sep	用于分隔多个对象的字符，默认是空格	<code>print("Hello", "World", sep="-")</code>	Hello-World
end	打印结束后的字符，默认是换行符\n	<code>print("Hello", end="!")</code>	Hello!

键盘输入

- ◆ 等待用户键盘输入，用户输入完毕后敲回车进行确认，结果保存在result中

```
1 | result = input()
```

- ◆ 在提示用户输入之前，先在屏幕显示出“请输入结果”

```
1 | result = input("请输入结果")
```

综合案例

◆ 看懂下面代码

```
1  '''
2      下面的代码用来欢迎用户
3      代码版本: 0.1
4  '''
5  name = input("你是谁? ")    # 提示用户输入名字
6  print("欢迎学习Python, ", name) # 显示欢迎用户
```