

Python基本语法

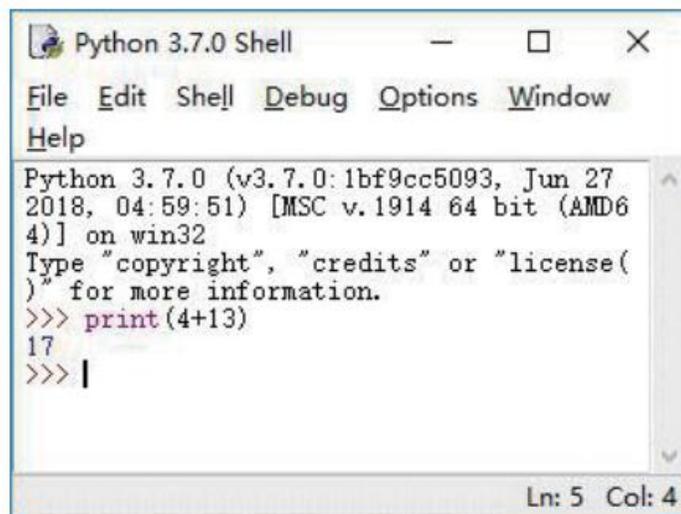
广州协和学校信息技术科

为什么学 Python

- 简单易学
- 有众多的库支持，功能强大
- 编程效率高
- 跨平台 (Windows, Linux, Mac OS...)

用IDLE 编辑程序并运行Python程序

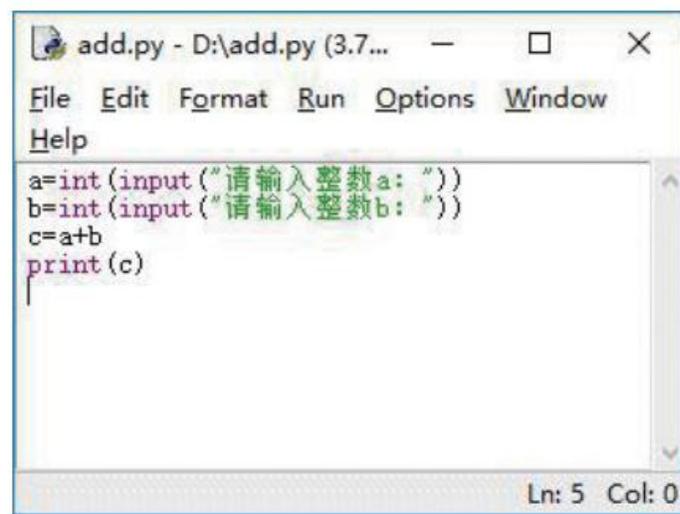
桌面图标找到“IDLE”，打开。



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print(4+13)
17
>>> |
```

Ln: 5 Col: 4

甲 解释器操作界面



```
add.py - D:\add.py (3.7...
File Edit Format Run Options Window Help
a=int(input("请输入整数a: "))
b=int(input("请输入整数b: "))
c=a+b
print(c)
```

Ln: 5 Col: 0

乙 编辑器操作界面

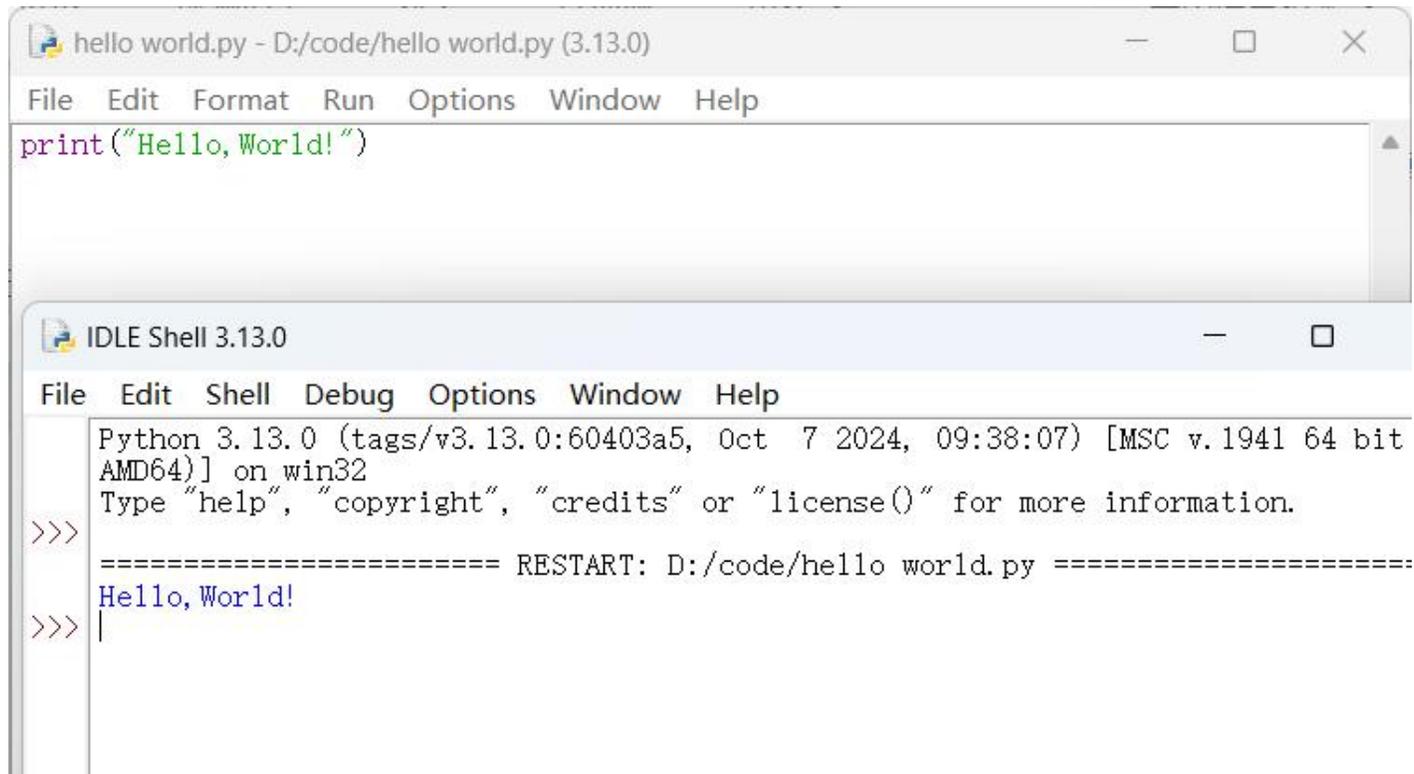
图3.2.1 IDLE的操作界面

最简单的Python程序

- `print("Hello,World")`

- 程序中的所有字符都必须是英文(半角)字符,不能是中文(全角)字符,注意.,(,"之类的,都必须是英文字符!

- 除非就是要输出中文文字,才会在“字符串”中使用中文



The screenshot shows two windows from a Python IDE. The top window, titled 'hello world.py - D:/code/hello world.py (3.13.0)', contains the code `print("Hello,World")`. The bottom window, titled 'IDLE Shell 3.13.0', shows the execution output: `Python 3.13.0 (tags/v3.13.0:60403a5, Oct 7 2024, 09:38:07) [MSC v.1941 64 bit AMD64] on win32`, followed by a prompt `>>>` and the output `Hello,World!`. A separator line indicates a restart of the program.

注释

- 并非需要执行的指令，只是为了编程者方便理解程序之用。
- 注释：“#” 开头。在IDLE里面显示为红色。



```
*test.py - D:\code\test.py (3.13.0)*  
File Edit Format Run Options Window Help  
a = b # 让a和b的值相同
```

变量

- 变量有名字，可以用来存储数据。其值可变。

```
a = 12
b = a      # 让b的值变得和a一样
print(a+b)
a = "hello"
print(a)
```

```
24
hello
```

变量的命名办法

- 变量由大小写字母、数字和下划线构成，中间不能有空格，长度不限，不能以数字开头。

name _doorNum x1 y z a2 A
number_of_students MYTYPE

- 变量名最好能够体现变量的含义。

例如：年龄（age）、学生姓名（studentName）、商品价格（productPrice）等

- 多单词的变量名，最好第一个单词小写，后面单词首字母大写。

dateOfBirth numOfDogs bookPrice

- 变量名是大小写相关的

a和 A, name 和 Name 是不同的变量。

- 哪些变量名是无效的？ fileNname01 \$result 2_result number_item

变量的命名办法

- 有些名字Python预留了，不可用作变量的名字。

and as assert break class continue def del
elif else except exec for finally from global if
import in is lambda not or pass print raise
return try while with yield

- 如果不小心用了，python会报错。

赋值语句

- 形式:
- 变量 = 表达式
- 将变量的值变得和“表达式”的值一样
- 变量、数、字符串.....都是“表达式”

```
a = "he"
print(a) # >>he
b = 3+2
a = b # a的值变得和b一样
print(b) # >>5
print(a) # >>5
b = b + a # b的值改为原来b的值加a
print(b) # >>10
```

```
a, b = "he", 12
print(a, b) # >>he 12
a, b = b, a # 交换 a, b的值
print(a, b) # >>12 he
c, a, b = a, b, a
print(a, b, c) # >>he 12 12
a = b = c = 10
print(a, b, c) #>>10 10 10
```

字符串

- 可以且必须用单引号、双引号或三单引号括起来。
- 字符串里面不会包含变量。

```
s = 1.75
print(s)                1.75
print("I am s m tall") I am s m tall
```

- 字符串必须用引号括起来，用引号括起来的就是字符串！

```
a = 4
b = 5
print("a+b") #>> a+b 不会打出9!!!!!! a+b
```

字符串

- 用"+"连接字符串

```
a = "ABCD"  
b = "1234"  
a = a + b  
print(a)
```

ABCD1234

Python数据类型

●int	整数	123456899899
●float	小数	3.2 1.5E6
●complex	复数	1+2j
●str	字符串	"hello"
●List	列表	[1,2,'ok',4.3]
●tuple	元组	(1,2,'ok',4.3)
●bool	布尔	True False
●dict	字典	{"tom":20,"jack":30}
●set	集合	{"tom",18,71}

输入输出初步

- 输出语句print

`print(x,y,z,...)` # 也可以只输出一项

连续输出多项，以空格分隔，然后换行

`print(x,y,z..., end="")`

连续输出多项，以空格分隔，不换行

```
print(3, 4, 5)
print(1, 2, 3, end="")
print("ok")
```

3 4 5
1 2 3ok

输入语句input

- 格式:

`x = input(y)`

x 是变量

y 是字符串, 或任何值为字符串的表达式

输出y, 并等待输入。敲回车后输入的字符串被赋值给 x

最简单的能处理输入的Python程序

- 输入一个字符串s，输出是该字符串复制三次的结果。

例如输入ABC，输出是ABCABCABC

```
s = input()
print(s + s + s)
```

```
ABC
ABCABCABC
```

- input() 每次输入一行，如果有多行输入，就用多次input()。

算术运算符

- Python 支持以下算术运算

+	加法
-	减法(双操作数) , 取相反数 (单操作数)
*	乘法
/	除法 , 结果是小数。即便能整除也是小数
%	取模(求余数)
//	求商, 结果取整数部分, 不要余数
**	求幂

算术表达式

```
a = (3+2)*(6-3)/2
print(a) #>>7.5
print(10/8) #>>1.25
print(10%8) #>>2
print(15/4) #>>3.75
print(15//4) #>>3
print(3.4/2.2) #>>1.5454545454545452
print(3.4//2.2) #>>1.0
print(2**3) #>>8
print(-9//4) #>>-3 往小里取整
```

```
7.5
1.25
2
3.75
3
1.5454545454545452
1.0
8
-3
```

算术运算符优先级

- 第一级: **
- 第二级: -(求相反数)
- 第三级: * / // %
- 第四级: + -(减法)
- 可以用() 指定计算顺序: $(3+4)*(1+2)$
- 记不得优先级就用 ()

算术运算的同时赋值

- += a += b 等价于 a = a + b
- -= a -= b 等价于 a = a - b
- *=
- /=
- %=
- //=
- **=

```
a = 6
a/=3
print(a) #>>2.0
a**=3
print(a) #>>8.0
```

2.0
8.0

常见问题或注意事项

- 数学上的运算符号或代数式写法，不是拿到程序里就能直接用的！

- 比如 $|x|$ 在程序里并不能求 x 的绝对值，而是没定义，会出错！

- $2(x+3)(4+x)$ 这样的表达式也不能在程序里写！

应该写：

$2*(x+3)*(4+x)$

- 乘法就一定要用 '*'！ $2*b$ 不能写成 $2b$ ！

常见问题或注意事项

- / 的结果一定是小数。
- 一个算术表达式，只要有一个操作数是小数，其结果就一定是小数。
- 如果x是小数print(x)就会打出小数形式，哪怕 $x = 4/2$ 。如果题目要求是整数，就要注意了，不要把整数当小数打出来。
- `print(4/2)` 打出 2.0

单选题

print(4+6/2)的结果是

A. 7

B. 7.0

C. 5.0

D. 5

单选题

print(4+6/2)的结果是

A. 7

B. 7.0

C. 5.0

D. 5