

Overview

- 计算机基础
- Python语言简介
- Python开发环境
- 标识符和变量
- 输入及输出函数

Python程序设计

2

1.1 计算机基础

- 1.1.1 计算机特点
 - 运算速度快
 - 计算精确度高
 - 具有记忆和逻辑判断能力
 - 人工智能

Python程序设计

3

1.1.2 计算机常用的数制

- 二进制，十进制，八进制，十六进制

二进制	十进制	八进制	十六进制	二进制	十进制	八进制	十六进制
0000	0	0	0	1000	8	10	8
0001	1	1	1	1001	9	11	9
0010	2	2	2	1010	10	12	A
0011	3	3	3	1011	11	13	B
0100	4	4	4	1100	12	14	C
0101	5	5	5	1101	13	15	D
0110	6	6	6	1110	14	16	E
0111	7	7	7	1111	15	17	F

Python程序设计

4

文本编码--ASCII码

- 字符编码(Character Code)是用二进制编码来表示字母、数字以及专门符号
- 普遍采用是ASCII(American Standard Code for Information Interchange)码，例如：

ASCII值	字符	ASCII值	字符	ASCII值	字符
32	(space)	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b

Python程序设计

5

编码的要素

- 码字：a
- 编码：97
- 位和字节表示：01100001

Python程序设计

6

Unicode编码

- Unicode码是计算机科学领域里的一项业界标准,包括字符集、编码方案等。
- Unicode是为了解决传统的字符编码方案的局限而产生的，它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换、处理的要求。
- ‘\u4e16\u754c\u60a8\u597d’ 代表汉字世界你好，“\u”表示Unicode码。

Python程序设计

7

UTF-8编码

- Unicode码并不是存储器中的编码，使用时是把Unicode编码转换字节或位，UTF-8编码就起这个作用。

每一个ASCII字符都有一个UTF-8编码，该UTF-8编码正好与8位的ASCII编码一样

编写Python3程序，默认的是UTF-8编码。

各种编码比较

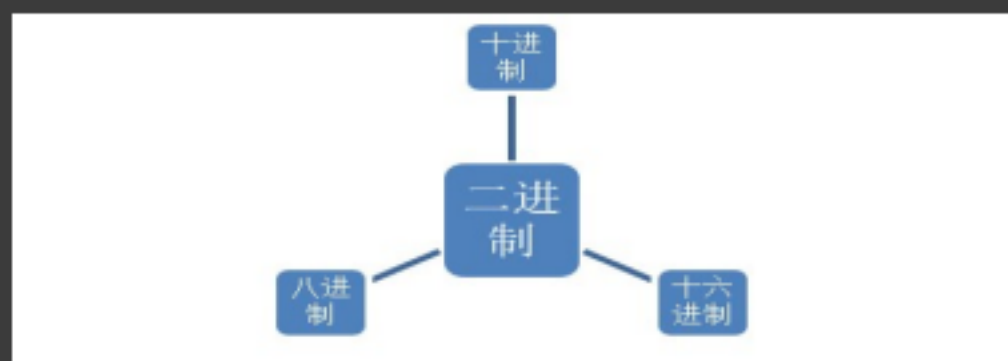
字符	ASCII	Unicode	UTF-8
Z	01111010	00000000 01111010	01111010
中	无	01001110 00101101	11100100 10111000 10101101

ASCII 编码可以看成UTF-8编码的一部分，支持ASCII编码的软件可以在UTF-8 编码下正常运行。

Python3语言里，在计算机内存中，缺省统一用Unicode编码。当需要保存到硬盘或网络传输时，可以转换成UTF-8编码。

1.1.3 进制转换和二进制运算

- 可以以二进制为中心，进行转换



- 1位八进制数等于3位二进制数，1位十六进制数等于4位二进制数。
 - $34.71_{(8)} = 0111100.111001_{(2)}$ $A02.D4_{(16)} = 101000000010.11010100_{(2)}$
 - $11.01_{(2)} = 011.010_{(2)} = 32_{(8)}$ $11000.01_{(2)} = 00011000.0100_{(2)} = 18.4_{(16)}$

十进制转成二进制

十进制数转二进制数需要整数部分和小数部分分别处理。

- 整数部分不断除以2，
- 直到商为0。倒序收集每次
- 得到的余数就可

$79_{(10)} = 1001111_{(2)}$

- $79 \div 2 = 39$ 余1
- $39 \div 2 = 19$ 余1
- $19 \div 2 = 9$ 余1
- $9 \div 2 = 4$ 余1
- $4 \div 2 = 2$ 余0
- $2 \div 2 = 1$ 余0
- $1 \div 2 = 0$ 余1

小数部分不断乘以2，直到小数部分为0。顺序收集每次得到的整数部分就可。

$0.625_{(10)} = 0.101_{(2)}$

- $0.625 \times 2 = 1.25$
- $0.25 \times 2 = 0.5$
- $0.5 \times 2 = 1.0$

二进制运算

- 加法运算
 - $0+0=0$
 - $0+1=1$
 - $1+0=1$
 - $1+1=10$
- 乘法运算
 - $0*0=0$
 - $0*1=0$
 - $1*0=0$
 - $1*1=1$

二进制减法运算

- 可以把减法变成加上一个负数： $a-b=a+(-b)$
- 用最高位是0表示正数，用最高位是1表示负数。这称为数的原码。如用8位二进制表示一个符号数，则3和-3分别表示为：

- $3 = 00000011$
- $-3 = 10000011$

原码、反码、补码

- 正数的原码，反码和补码都一样。负数的反码是用它的原码转化得到。
- 转化规则是符号位不变，其他位0变1,1变0。负数的补码是它的反码加1。
- 计算机内部用补码进行运算，两个数相减变成两个数的补码运算。如 $14-23=14+(-23)=-9$ 运算过程如下

	原码	反码	补码
14	00001110	00001110	00001110
-23	10010111	11101000	11101001

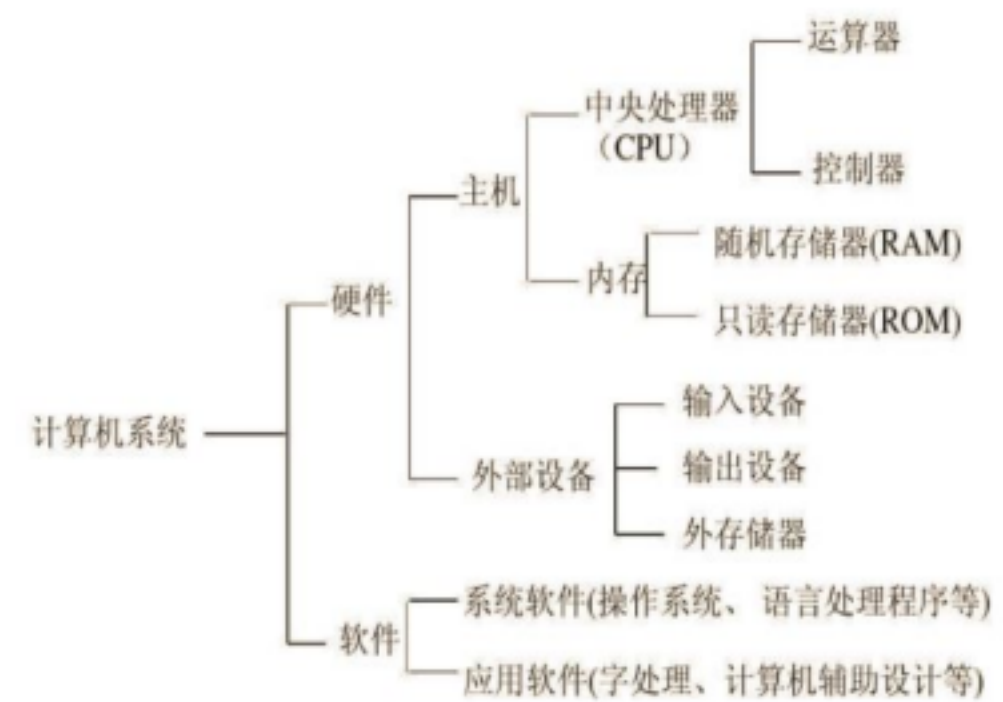
计算机内部用补码进行运算。用补码做加法运算，运算结果也是补码。

```
00001110
+ 11101001
-----
11110111
```

11110111是补码，要变成原码才能知道本来的值。补码的补码就是原码。
11110111的原码是10001001，变成十进制是-9。

1.1.4 计算机系统

计算机系统组成图



内存模型

单元地址	单元内容
0000.....0000	0 1 0 1 0 1 0 1
0000.....0001	1 1 0 0 1 1 0 0
0000.....0010	1 0 1 1 0 1 0 0
⋮	
1111.....1101	0 0 1 1 0 0 1 1
1111.....1110	1 0 0 1 0 0 1 1
1111.....1111	0 1 1 0 0 0 1 0

存储单位

单位		实际字节数	近似表示方法
B(Byte)	字节	1	1
KB(K Byte)	千字节	2^{10}	10^3
MB(M Byte)	兆字节	2^{20}	10^6
GB(G Byte)	千兆字节(吉)	2^{30}	10^9
TB(T Byte)	兆兆字节(太拉)	2^{40}	10^{12}

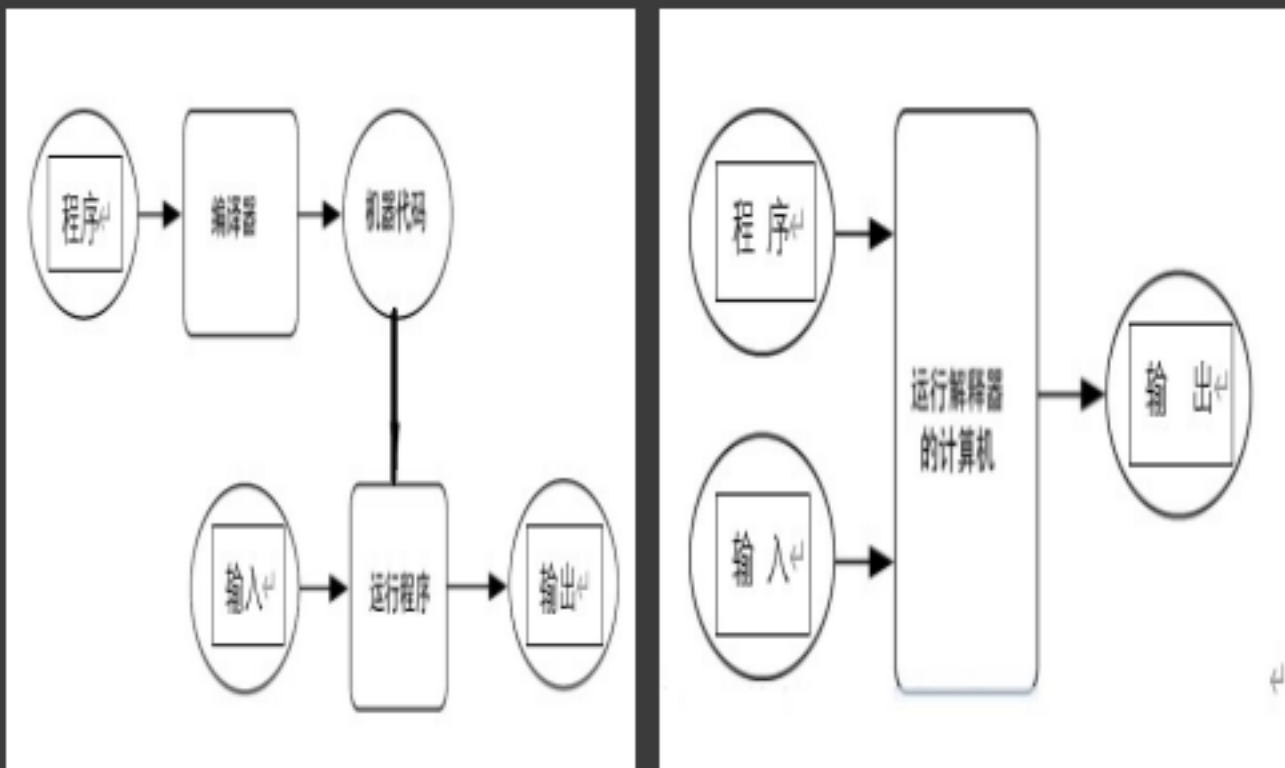
1.1.5 操作系统

- 操作系统 (Operating System, 简称OS) 是管理和控制计算机硬件与软件资源的计算机程序，是直接运行在“裸机”上的最基本的系统软件
- Unix/Linux
- Windows
- Mac OS

1.1.6 程序设计语言

- 机器语言
- 汇编语言
- 高级语言
 - C/C++ 语言
 - Java 语言
 - Python 语言

翻译程序-编译器和解释器



Python程序设计

20

1.2 Python语言简介

- Python是一种面向对象的解释型计算机程序设计语言，由荷兰人Guido van Rossum于1989年发明，第一个公开发布版发行于1991年。
- Python的设计哲学是“优雅”、“明确”、“简单”
- Python是自由软件之一，免费、开源。
- Python已经被移植到许多平台上。这些平台包括Unix/Linux、Windows、Mac OS。



Python程序设计

21

Python集成开发环境

- IDLE: Python安装包内置开发环境
- Thonny, Python IDE for beginners
- 网址是 <https://thonny.org/>
- jupyter: 安装Anaconda自带的高级IDE, 数据科学家首选开发环境
- PyCharm: 由著名的JetBrains公司开发, 带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具。程序员的开发环境

Python程序设计

22

1.3 Python IDLE开发环境

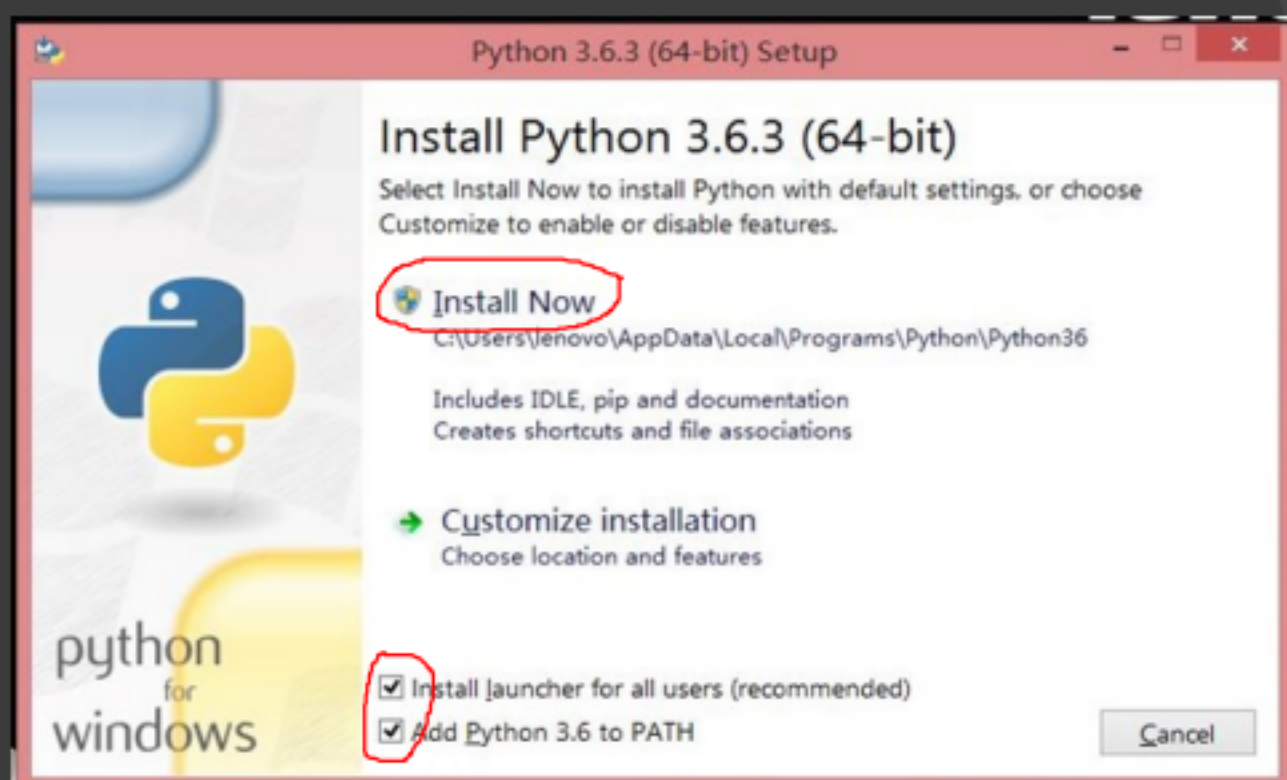
- Python: <https://www.python.org/downloads>
- 选择Python版本
- 选择操作系统

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPG
Getipped source tarball	Source release		9de6494314ea199e363211606735965	22710891	SIG
XZ compressed source tarball	Source release		1325134d525b4a2c3272a1a0214dd54	16992824	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	9fba50521dffa9238ce8ad640abaa92	27778156	SIG
Windows help file	Windows		17cc49512c3a2b8762ed8022e0afe92	8041937	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors	d2fb546fd4b189146dbefeb85e7266b	7162335	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors	bee5746dc6ce6ab9573a9f54b5d0a1	31684744	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors	21525b3d132ce15cae6ba96d74961b5a	1320128	SIG
Windows x86 embeddable zip file	Windows		15802be75a6246070d85b87b3f43f83f	6400788	SIG
Windows x86 executable installer	Windows		67e1a9bb336a5eca9efcd481c9f262a4	30653888	SIG
Windows x86 web-based installer	Windows		6c8ff748c554559a385c986453d28ef	1294088	SIG

Data and Computation

23

安装选项



Python程序设计

24

运行Python IDLE开发环境



Python程序设计

25

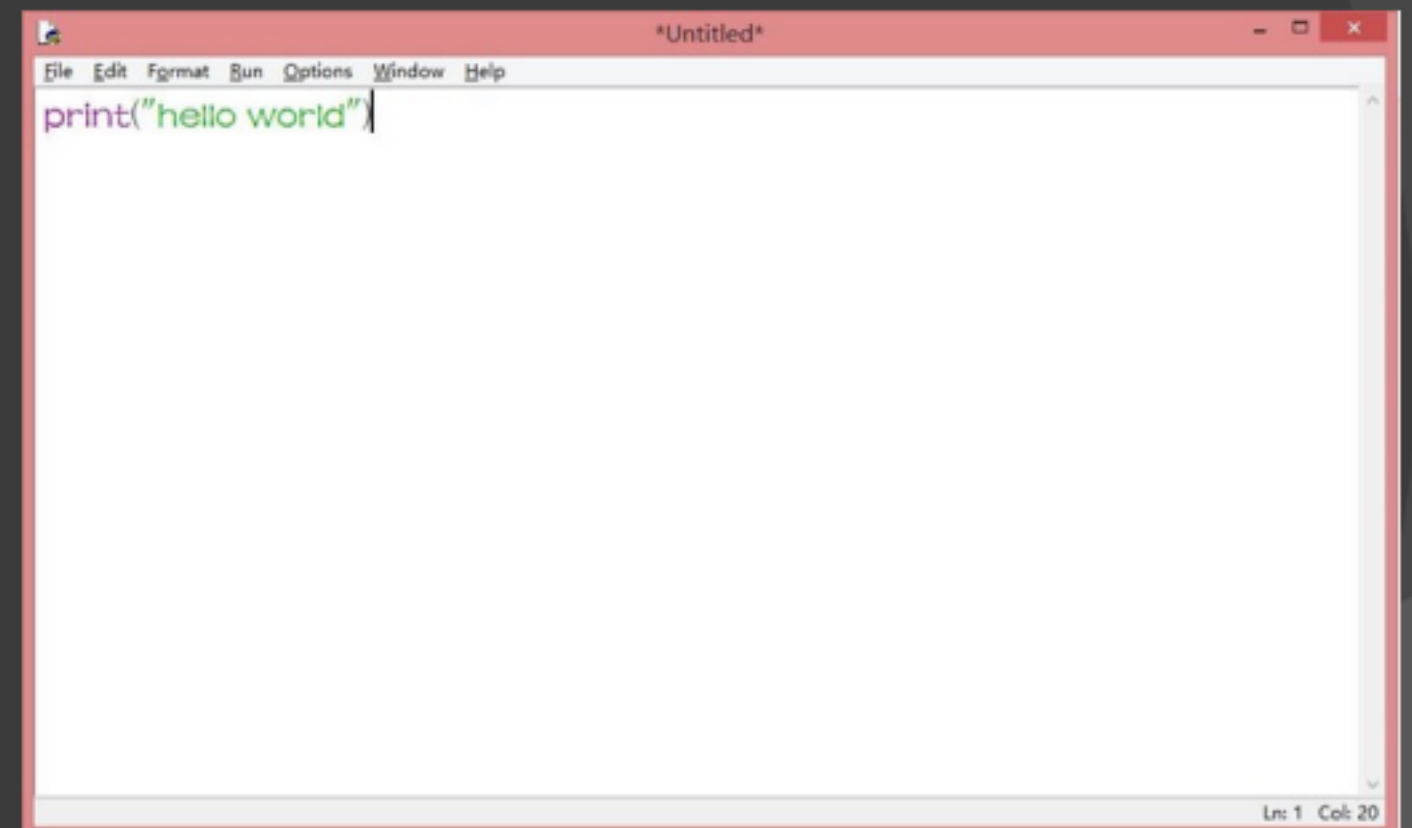
运行程序:交互式解释器执行

- `>>>3+5*6`
- `33`
- `>>>print("hello world")`
- `hello world`

Python程序设计

26

运行程序: File | New,Save,Run



Python程序设计

27

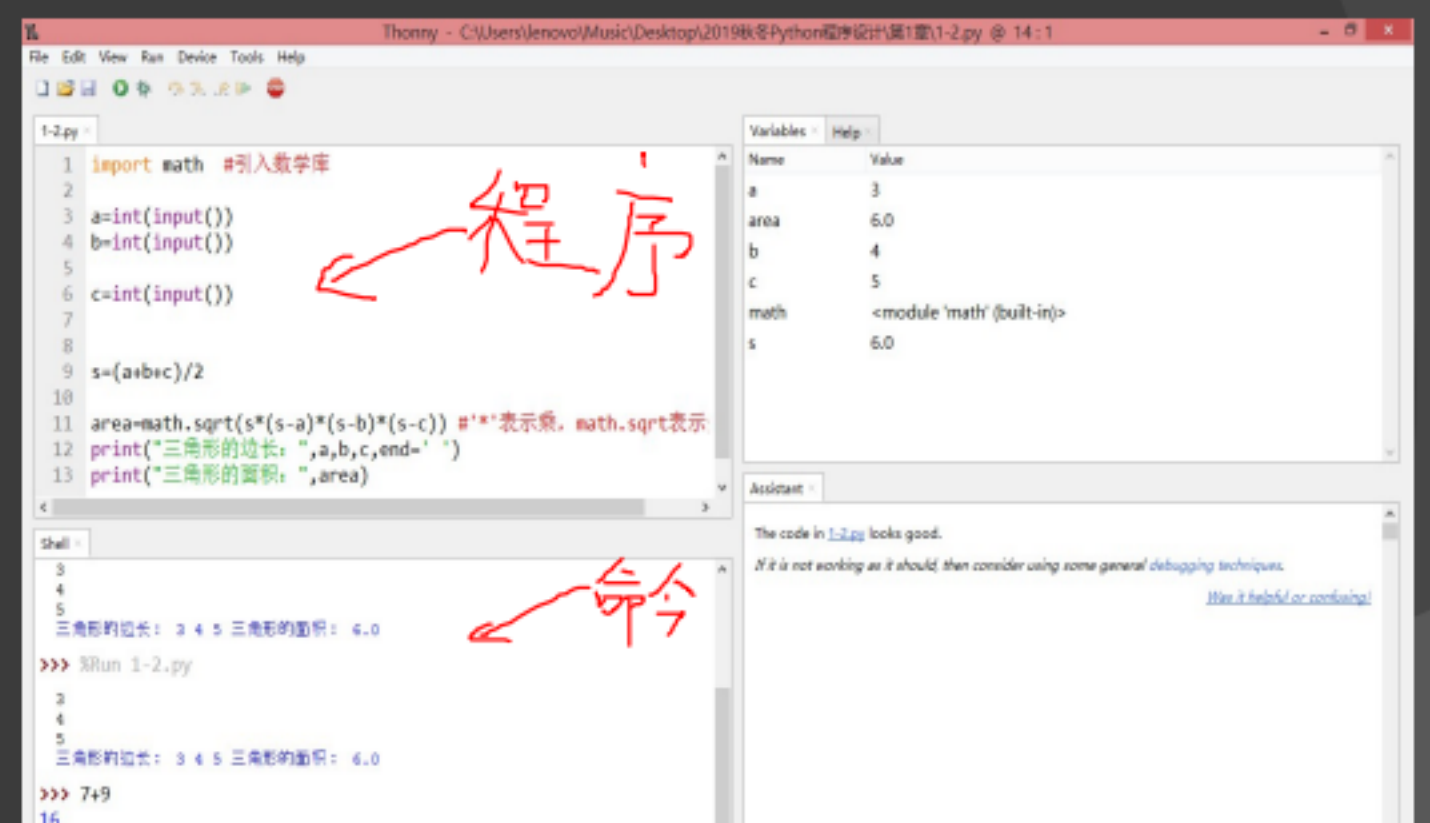
运行程序:命令行环境运行

- `d:>python hello.py`
- `hello.py`是python程序, 放在d盘根目录
- `hello.py`:
`print("hello world")`

Python程序设计

28

Thonny开发环境 (推荐使用)



Python程序设计

29

Thonny下载地址

- <https://thonny.org/>
- Run菜单运行程序
- Thonny视频教程
<https://www.bilibili.com/video/av88021729/>

Python程序设计

30

1.4 标识符和变量

- 标识符是指用来标识某个实体的一个符号, 在不同的应用环境下有不同的含义
- 标识符由字母、下划线和数字组成, 且不能以数字开头
- python中的标识符是区分大小写的, `Andy`与`andy`是不同的标识符
- 如: `my_test`
`_123`

Python程序设计

31

Python关键字

- python一些特殊的组合，是所谓的关键字。关键字不允许作为标识符。
- Python关键字：

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

Python程序设计

32

常量和变量

- 常量就是不能改变的量，比如常用的数学常数3.14159就是一个常量
- 变量就是程序为了方便地引用内存中的值而为其取的名称。Python变量名是大小写敏感的
- >>>a=7 # “=”是赋值号
- >>>a
- 7
- 7是一个对象，可以通过变量a引用这个对象

Python程序设计

33

id函数

- Python变量有一个非常重要的性质：变量是将名字和对象关联起来。赋值操作并不会实际复制值，它只是为数据对象取个相关的名字。名字是对象的引用而不是对象本身
- id是Python的内置函数，返回对象的标识
- 标识通常用对象的地址表示

Python程序设计

34

id函数用法

```
>>> id(a)
1566532000

>>> id(7)
1566532000

>>> a=5
>>> a
5
>>> id(a)
1566531936
>>> id(5)
1566531936
>>> id(7)
1566532000
```



Python程序设计

35

1.5 输入及输出函数

- 输入函数：input() *(str)*
- input从键盘输入一个字符串。‘9’表示是一个字符串，它的ASCII码值是57
- >>>a=input()
9
>>>a
‘9’

Python程序设计

36

输入数字

- 用int()函数输入数字
- >>>a=int(input())
9
>>>a
9

Python程序设计

37

一行输入多个值

```
>>>m,n=input("请输入多个值：").split()
```

请输入多个值： 3 5

```
>>>m
```

```
'3'
```

```
>>>n
```

```
'5'
```

input("请输入多个值：")函数中的参数是输出提示字符串

Python程序设计

38

输出函数：print()

- print是输出函数，参数是输出值

- >>> print(3) #输出1个数字

- 3

- >>>print(3,7) #输出2个数字

- 3 7

- >>> b,c=3,4 #输出1个数字，两个变量

- print(b, c, 5)

- 3 4 5

- 井号“#”常被用作单行注释符号，在代码中使用“#”时，它右边的任何数据都会被忽略，当做是注释

Python程序设计

39

不换行输出

- 每行输出一个值

- >>>print(3)

- >>>print(4)

- >>>print(5)

- 用end参数，一行输出三个值，

- print(3,end=' ')

- print(4,end=' ')

- print(5,end=' ')

Python程序设计

40

输入三角形的三边长度3,4,5，求这个三角形的面积

编程可以用IPO模式完成

输入input 处理process 输出output

```
import math #引入数学库
```

```
#输入
```

```
a=int(input())
```

```
b=int(input())
```

```
c=int(input())
```

```
#处理
```

```
s=(a+b+c)/2
```

```
area=math.sqrt(s*(s-a)*(s-b)*(s-c)) # '*'表示乘，math.sqrt表示开根号
```

```
#输出
```

```
print("三角形的边长：",a,b,c,end=' ')
```

```
print("三角形的面积：",area)
```

Python程序设计

41

画五角形

```
import turtle
```

```
turtle.forward(200)
```

```
turtle.right(144)
```

```
turtle.forward(200)
```

```
turtle.right(144)
```

```
turtle.forward(200)
```

```
turtle.right(144)
```

```
turtle.forward(200)
```

```
turtle.right(144)
```

```
turtle.forward(200)
```

```
turtle.done()
```

Python程序设计

42