

大学计算机基础

1

第二章 计算的基础

2

课程内容

- 计算机的数（原码、反码、补码）
- 逻辑与门电路
- 程序设计

3

计算机的数

4

计算机中的数

计算机内部使用门电路，只有0和1两种状态，所以计算机内部使用二进制表示

5

无符号数

$0000\ 0000 = 0$

$0000\ 1000 = 8$

缺点：不能表示负数

6

机器数

机器数：带符号的二进制数，最高位作为符号位。

7

原码

- 带符号的二进制数
- 最高位（最左侧）表示符号，0为正，1为负
- 其余有效值部分用二进制绝对值表示

8

原码

- 带符号的二进制数
- 最高位（最左侧）表示符号，0为正，1为负
- 其余有效值部分用二进制绝对值表示

0000 0001

1000 0001

9

原码

1的原码是0000 001， -1的原码是1000 001

缺点：

正负相加不为0，例如0000 0001+1000 0001 = 1000 0010

存在两个：-0和+0，例如1000 0000， 0000 0000

10

反码

为保证正负相加等于0

原码为正，反码与原码一致

原码为负，反码对原码各位置取反（符号位除外）。

11

反码

解决正负相加不等于0的问题：

缺点：依然存在两个0

$1 + (-1)$

$= 0000\ 0001 + 1111\ 1110$

$= 1111\ 1111$

$= -0$

12

补码

原码为正，补码与原码一致

原码为负，反码+1

13

补码

$1 + (-1)$

$= 0000\ 0001 + 1111\ 1111$

$= 1\ 0000\ 0000$ (丢弃最高位)

$= 0$

只有一个0

正负可相加

14

原码，反码和补码总结

原码：带符号的二进制数，最高位（最左侧）表示符号，0为正，1为负

反码：原码为正，反码与原码一致；原码为负，反码为原码各位置取反（符号位除外）。

补码：原码为正，补码与原码一致；原码为负，补码为反码加1。

15

原码，反码和补码总结

+1010010的原码？反码？补码？

-1010010的原码？反码？补码？

16

原码、反码、补码的范围

原码的范围？最小值？最大值？

反码的范围？

补码的范围？

17

原码、反码、补码的范围

原码的范围？

[1111 1111, 0111 1111] [-127, 127]

反码的范围？

补码的范围？

18

原码、反码、补码的范围

原码的范围？

[1111 1111, 0111 1111] [-127, 127]

反码的范围？

[1000 0000, 0111 1111] [-127, 127]

补码的范围？

19

原码、反码、补码的范围

原码的范围？

[1111 1111, 0111 1111] [-127, 127]

反码的范围？

[1000 0000, 0111 1111] [-127, 127]

补码的范围？

[1000 0000, 0111 1111] [-128, 127]

20

原码、反码、补码练习

1. 已知原码1011 0100, 求补码? 1 100 1100
2. 已知补码1110 1110, 求原码? 1 001 0010
3. 假设某计算机的字长为8, 则10进制数-66的补码为? 1011 1110
4. 假设某计算机的字长为8, 则10进制数+75的反码为? 0100 1011

21

实数

实数是带有整数部分和小数部分的数字, 例如23.7是一个实数整数部分是23, 而小数部分是0.7。

实数在计算机中表示, 包括定点数表示法和浮点数表示法。

22

定点数表示

定点纯小数格式，把小数点固定在数值部分最低位的右面



23

定点数表示

定点数的处理对硬件要求比较简单，但它的表示范围受到限制，带有很大的整数部分或很小的小数部分的实数不应该用定点表示。

24

浮点数

该表示法允许小数点浮动，取值范围更大，一个浮点数分阶码和尾数两个部分。阶码表示小数点在该数中的位置也是一个带符号的整数，尾数表示数的有效值，定义了该数的精度。

数的符号	阶码符号	阶码值	尾数
1 bit	1 bit	7 bit	23bit

25

程序设计基础

26

Python的启动

27

Python启动

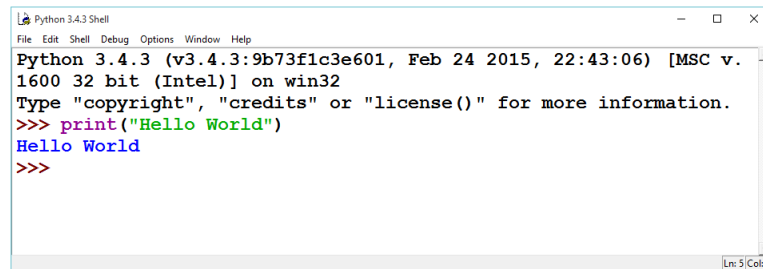
方法1：启动Windows命令行工具，输入python。

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World")
Hello World
>>>
```

28

Python启动

方法2：调用IDLE来启动Python图形化运行环境。

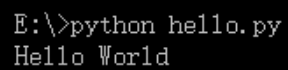
A screenshot of a Windows command prompt window titled "Python 3.4.3 Shell". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The text inside the window shows the Python version and build information: "Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32". It then prompts the user to type "copyright", "credits", or "license()" for more information. The user has entered the command ">>> print('Hello World')", and the output "Hello World" is displayed. The prompt ">>>" is shown again on the next line. The status bar at the bottom right indicates "Ln: 5 / Col: 4".

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.
1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('Hello World')
Hello World
>>>
```

29

Python启动

方法3：按照语法格式编写代码，编写可以用任何文本编辑器，保存为文件。

A screenshot of a black command prompt window. The text inside shows the command "E:\>python hello.py" being entered, followed by the output "Hello World".

```
E:\>python hello.py
Hello World
```

30

Python启动

方法4：打开IDLE，点击Ctrl+N打开一个新窗口，输入语句并保存，使用快捷键F5即可运行该程序

31

Python启动

方法5：将Python集成到Eclipse、PyCharm等面向较大规模项目开发的集成开发环境中。

32

Python语法元素分析

33

Python书写规则

1. 单行书写：

原则上我们希望看见的是一行只有一条Python 语句。如果要在一行中写入多条Python 语句，那么需要使用分号 (;) 来特别地标明这种用法。分号表示一条语句的结束。

34

Python书写规则

2. 缩进:

空白在Python中是重要的。它称为缩进。在命令行首的空白（空格和制表符）用来决定命令行的缩进层次，从而用来决定语句的分组。同一层次的语句必须有相同的缩进。

35

Python书写规则

单层缩进

```
#e1.1TempConvert.py
TempStr = input("请输入带有符号
if TempStr[-1] in ['F','f']:
    C = (eval(TempStr[0:-1]) -
    print("转换后的温度是{:.2f}C
elif TempStr[-1] in ['C','c']:
    F = 1.8*eval(TempStr[0:-1]
    print("转换后的温度是{:.2f}F
else:
    print("输入格式错误")
```

多层缩进

```
DARTS = 1000
hits = 0.0
clock()
for i in range(1, DARTS):
    x, y = random(), random()
    dist = sqrt(x ** 2 + y **
    if dist <= 1.0:
        hits = hits + 1
pi = 4 * (hits/DARTS)
print("Pi的值是{:.2f}".format
```

36

Python书写规则

3. 注释：

在#符后面写的文字，一律认为是注释语句，不参与运行。注释语句对于阅读程序很有帮助，命令行后加注释语句是一个程序员的优良品质之一。此外，程序员在测试代码时，也可以使用注释来暂时禁用一行代码。

37

注释

注释：程序员在代码中加入的说明信息，不被计算机执行

注释的两种方法：

单行注释以#开头

```
#Here are the comments
```

多行注释以'''开头和结尾

```
'''
```

```
This is a multiline comment
```

```
used in Python
```

```
'''
```

38

命名与保留字

常量：程序中值不发生改变的元素

变量：程序中值发生改变或者可以发生改变的元素

Python语言允许采用大写字母、小写字母、数字、下划线(_)和汉字等字符及其组合给变量命名，但名字的首字符不能是数字，中间不能出现空格，长度没有限制

注意：标识符对大小写敏感，python和Python是两个不同的名字

39

命名与保留字

保留字，也称为关键字，指被编程语言内部定义并保留使用的标识符。

程序员编写程序不能定义与保留字相同的标识符。

每种程序设计语言都有一套保留字，保留字一般用来构成程序整体框架、表达关键值和具有结构性的复杂语义等。

掌握一门编程语言首先要熟记其所对应的保留字。

40

命名与保留字

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

41

变量

计算机程序中变量的概念基本上与初中代数的方程变量是一致的。主要的不同之处有三点：

1. 变量的命名规则
2. 等号(=)是赋值语句，例如a=1
3. 数据类型

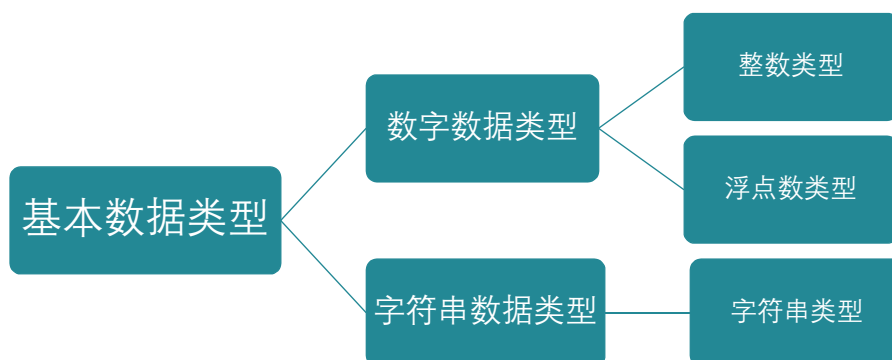
42

变量的驼峰命名法

1. 小驼峰命名法:第一个单词以小写字母开始;第二个单词的首字母大写,例 myName,aDog
2. 大驼峰命名法:每一个单词的首字母都采用大写字母,例如:FirstName,LastName
3. 不过在程序员中还有一种命名法比较流行,就是用下划线来连接所有单词,例如:send_buf

43

Python数据类型



44

数字类型操作

Python中，数字类型的算术操作符如下：

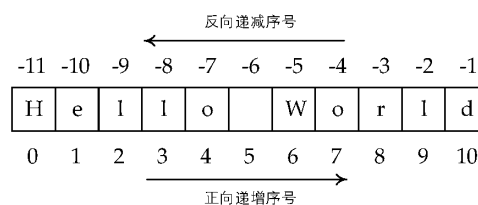
操作符	操作	例子	求值为
**	指数	2 ** 3	8
%	取模/取余数	22 % 8	6
//	整除/商数取整	22 // 8	2
/	除法	22 / 8	2.75
*	乘法	3 * 5	15
-	减法	5 - 2	3
+	加法	2 + 2	4

45

字符串

Python语言中，字符串是用两个双引号" "或者单引号' '括起来的一个或多个字符。

Python字符串的两种序号体系



46

字符串五大操作总结

操作符	描述
x+y	连接字符串x与y
x*n	复制n次字符串x
str[i]	索引，返回第i个字符
str[N:M]	切片，返回索引第N到第M的子串，其中不包含M
x in s	如果x是s的字符串，返回True

47

input()函数

获得用户输入之前，input()函数可以包含一些提示性文字

<变量> = input(<提示性文字>)

```
>>>input("请输入: ")
请输入: python
'python'
>>> input("请输入: ")
请输入: 1024.256
'1024.256'
```

48

eval()函数

eval(<字符串>)函数是Python语言中一个十分重要的函数，它能够以Python表达式的方式解析并执行字符串，将返回结果输出

```
>>>x = 1
>>>eval("x + 1")
2
>>>eval("1.1 + 2.2")
3.3
```

49

print()函数

print()函数用来输出字符信息，或以字符形式输出变量。

print()函数可以输出各种类型变量的值。

print()函数通过%来选择要输出的变量。

50

温度转换案例

温度刻画存在不同体系，摄氏度以1标准大气压下水的结冰点为0度，沸点为100度，将温度进行等分刻画。华氏度以1标准大气压下水的结冰点为32度，沸点为212度，将温度进行等分刻画。

51

温度转换案例

问题：如何利用Python程序进行摄氏度和华氏度之间的转换

步骤1：分析问题的计算部分：采用公式转换方式解决计算问题

52

温度转换案例

步骤2：确定功能

输入：华氏或者摄氏温度值、温度标识

处理：温度转化算法

输出：华氏或者摄氏温度值、温度标识

F表示华氏度，82F表示华氏82度

C表示摄氏度，28C表示摄氏28度

53

温度转换案例

步骤3：设计算法

根据华氏和摄氏温度定义，转换公式如下：

$$C = (F - 32) / 1.8$$

$$F = C * 1.8 + 32$$

其中，C表示摄氏温度，F表示华氏温度

54

温度转换案例

步骤4：编写程序

```
#TempConvert.py
val = input("请输入带温度表示符号的温度值(例如: 32C): ")
if val[-1] in ['C', 'c']:
    f = 1.8 * float(val[0:-1]) + 32
    print("转换后的温度为: %.2fF"%f)
elif val[-1] in ['F', 'f']:
    c = (float(val[0:-1]) - 32) / 1.8
    print("转换后的温度为: %.2fC"%c)
else:
    print("输入有误")
```

55

温度转换案例

步骤5：调试、运行程序

在系统命令行上运行如下命令执行程序：

C:\>python TempConvert.py

或者：使用IDLE打开上述文件，按F5运行（推荐）

输入数值，观察输出

56