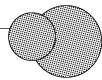


实验 1

资料查询



1. 实验任务与目的

(1) 上网查找计算机科学与技术以及 IT 产业方面的相关资料。了解学习“大学计算机基础”课程的必要性。

(2) 上网查找“大学计算机基础”课程涉及的几个重要名词和概念的相关资料。为进一步学习做必要的准备。

2. 实验步骤

(1) 上网查找并浏览计算机科学技术的相关内容。

- 计算机的诞生与计算机科学技术的发展历程。
- 计算机软件的概念与分类。
- 操作系统的主要功能以及常用操作系统的特点。
- 常见高级程序设计语言的谱系(传承关系)。
- 计算机的应用范围(按其技术特征来分类)。

(2) 上网查找并浏览计算机学科的相关内容。

- 计算机学科的概念。
- 计算机学科的主要研究方向。
- 计算机学科方法论概述。
- 计算机学科的主要应用领域。
- 计算机科学与技术专业的主要课程。

(3) 上网查找并浏览 IT 产业的相关内容。

- 什么是 IT 产业?
- IT 产业的几个主要组成部分。
- IT 产业的发展历程。
- 世界 IT 产业的现状。
- 中国 IT 产业的现状及预测。

(4) 上网查找以下内容。

- 软件版权保护的概念以及版权意义上的软件分类。
- 自由软件的发行、获取方式以及常用自由软件(如 Linux)的功能与特点。

- 图灵机的概念及基本工作方式。
- 冯·诺依曼计算机的工作方式。
- 普适计算的概念。
- 云计算的概念。

(5) 编写一篇文章。

从上面列出的内容中选择一项或多项,深入探讨相关内容,编写成一个 Word 文档。例如,可上网查找软件版权保护方面的相关内容并按以下提纲编写文章。

标题:软件版权保护

内容:

- 软件版权的概念。
- 软件版权保护的意義。
- 版权意义上的软件分类。
 - ◆ 商品软件。
 - ◆ 免费软件。
 - ◆ 共享软件。
 - ◆ 公用软件。
- 相关法律法规及道德原则。
- 自由软件的概念。

3. 练习题

(1) 第四代计算机的主要元器件是由什么材料构成的?最早的操作系统是用于第几代计算机的?

(2) 什么是软件?以下产品中哪些不是软件?为什么?

Linux 操作系统;

IBM DB II 数据库管理系统;

一个用 C++ 语言编写的求解一元二次方程的程序;

Python 集成开发环境;

Python 解释器的使用说明书。

(3) 简述从最早的高级语言到 C++ 语言的继承路线。

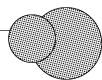
(4) 简单解释下列名词:

CAD;实时操作系统;模式识别;DBMS;普适计算;云计算;CPU;计算机主板;数字签名。

(5) 简述冯·诺依曼计算机的基本工作方式。

实验 2

程序设计基本训练



实验 2-1 Python 编程环境的安装和使用

1. 实验任务与目的

- (1) 下载 Python 编程环境软件。了解自由软件的使用方式。
- (2) 安装 Python 编程环境。理解程序设计语言与支撑环境的区别与联系。
- (3) 分别在 Python 编程环境的命令提示符方式下以及 Windows 系统的命令提示符窗口中进行基本的程序设计练习。了解 Python 编程的一般方法并体会具备基本功能(输入、输出和算术运算)的程序的编辑和运行方法。

2. 实验步骤

- (1) 下载 Python 软件。

Python 的下载地址为 <http://www.python.org/download/>。

目前有两个版本: Python 2.7.1 和 Python 3.1.3。已有的第三方软件与 Python 2.7 的兼容性更好,而 Python 3 并没有考虑向下兼容。不过可以使用 2to3 对源程序进行转换。

基于功能、易用性等方面的考虑,Python 3 做了很多修改,例如,将用于输出的 print 语句改成了 print 函数。因此,Python 2.7 程序中的输出语句:

```
print "提示信息"
```

在 Python 3 中应改为:

```
print("提示信息")
```

因为目前 2.X 版本的教材很多,学习时应注意分辨。

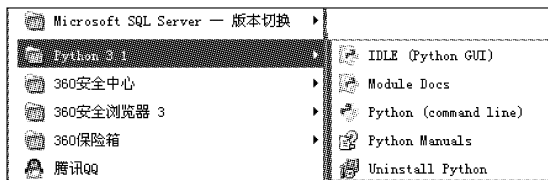
本实验请下载 Python 3.X 版。

- (2) 安装 Python 软件。

下载后的安装过程与一般软件的安装类似。默认安装路径是 c:\python31,选项出现在 Windows 开始菜单中,如实验图 2-1 所示。

有以下三种使用 Python 的方式:

- Python 编程环境中的图形用户界面。
- Python 编程环境中的命令行方式。



实验图 2-1 开始菜单中的 Python 选项

- Windows 操作系统的命令行方式。

(3) Python 命令行方式下的输出和简单计算。

S1 单击“开始”按钮,选择菜单项:“程序”|Python 3.1|IDLE(Python GUI),打开 Python Shell 窗口。

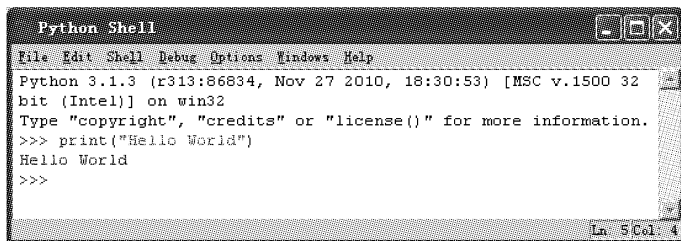
S2 在“>>>”提示符下输入

```
print("Hello World")
```

并回车(按 Enter 键),即可执行该语句并显示结果

```
Hello World
```

如实验图 2-2 所示。



实验图 2-2 Python 编程环境的窗口

S3 “>>>”提示符再出来后,输入

```
6+9
```

并回车,即可执行该语句并显示结果

```
15
```

S4 “>>>”提示符再出来后,按 Ctrl+Q 键或使用菜单项 file|Exit,即可关闭 Python Shell 窗口,即退出用户和 Python 编程环境的交互方式。

(4) Python 命令行方式下的变量定义和使用。

S1 再次打开 Python 解释器窗口。

S2 在“>>>”提示符后输入并执行(回车)。

```
x=10
```

S3 “>>>”提示符再出来后,输入并执行

```
2 * x+1
```

S4 “>>>”提示符再出来后,输入并执行

```
print("y=",x * 3-2 * x+1)
```

S5 “>>>”提示符再出来后,输入并执行

```
print("判断:",x+1<10)
```

S6 关闭 Python Shell 窗口。

(5) Windows 命令提示符窗口中的输出和计算。

S1 单击“开始”按钮,选择菜单项:“程序”|“附件”|“命令提示符”,打开 Windows 的“命令提示符”窗口。

S2 输入命令

```
盘符号+冒号:"+↵"
```

如“C:↵”或“D:↵”,切换到系统盘根目录。

注:“↵”表示 Enter 键。

S3 输入命令

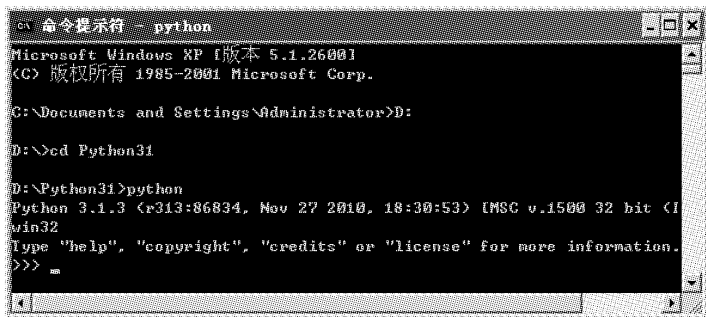
```
CD Python31↵
```

进入 Python 系统所在的文件夹。

S4 输入命令

```
python↵
```

启动 Python 解释器,如实验图 2-3 所示。



实验图 2-3 在 Windows“命令提示符”窗口启动 Python 解释器

S5 Python 解释器启动之后,便可像在 Python Shell 窗口中一样进行 Python 语言所支持的各种操作了。请按实验图 2-4 所示的操作完成本步操作。

S6 按 Ctrl+Z 或输入命令

```
exit()↵
```

退出 Python 解释器,切换到 Windows 命令提示符状态。



```

命令提示符 - python
Type "help", "copyright", "credits"
>>> x=-0.9
>>> import math
>>> print("x是正数吗?",x>=0)
x是正数吗? False
>>> -math.sin(1-x)+9
8.053699912312586
>>> =

```

实验图 2-4 在 Windows“命令提示符”窗口启动 Python 解释器

S7 关闭 Windows“命令提示符”窗口。

3. 练习题

- (1) Python 编程环境与 Python 解释器有什么联系与区别?
- (2) 在使用 $\sin(x)$ 函数之前,为什么要输入 `import math` 命令?
- (3) 在 Python Shell 窗口中,定义其值分别为 3.5 和 6.8 的 `a` 和 `b` 两个变量,并计算以 `b` 为底边 `a` 为高的三角形的面积。
- (4) 在 Python Shell 窗口中,分别用不同的变量来保存一个学生的学号、姓名、性别和一门课的分,并在一行中输出这几个变量的值。
- (5) 在 Windows 命令提示符窗口中,计算半径为 3.1416 的圆的周长和面积。

实验 2-2 程序中的输入/输出及分支结构

1. 实验任务与目的

(1) 在 Python 编程环境中,编写并运行程序:输入半径值,计算相应的圆周长、圆面积和球体积。掌握 Python 程序设计的一般方法。

(2) 在 Python 编程环境中,编写并运行求解分段函数

$$y = \begin{cases} 2x + 1 & (x \geq 0) \\ -x & (x < 0) \end{cases}$$

的程序。理解分支结构程序并进一步体会 Python 程序设计的一般方法。

(3) 在 Windows 命令提示符窗口中,编写并运行求解分段函数

$$y = \begin{cases} 2x + 1 & (x > 0) \\ 0 & (x = 0) \\ -x & (x < 0) \end{cases}$$

的程序。了解 Windows 环境中程序设计的一般方法,体会集成开发环境(Python 编程环境)的优越性。

2. 实验步骤

(1) 在 Python 编程环境中,编写输出字符串的程序。

S1 打开 Python Shell 窗口。

S2 选择菜单项: File|New Window, 打开 Python 编程窗口。刚打开时, 窗口的名称为 Untitled, 编辑并保存了源程序之后, 替换为所保存的文件名。

S3 编辑以下程序, 并以“周长面积体积.py”为文件名保存于 Python 系统所在的文件夹中。

```
r=1.9
pi=3.1415926
L=2*pi*r
S=pi*r*r*2
print("圆周长:",L)
print("圆面积:",S)
print("球体积:",4*pi*r*r*3/3)
```

S4 选择菜单项: Run|Run Module, 运行程序并切换到 Python Shell 窗口。

S5 查看运行结果, 关闭 Python Shell 窗口, 即退出 Python 编程环境。

(2) 在 Python 编程环境中, 编写包含分支结构的程序。

S1 打开 Python Shell 窗口, 并在其中打开 Python 编程窗口。

S2 编辑求解分段函数

$$y = \begin{cases} 2x + 1 & (x \geq 0) \\ -x & (x < 0) \end{cases}$$

的程序

```
x=float(input("x=?"))
if x>=0:
    y=2*x+1
else:
    y=-x
print("y=",y)
```

并以“分段函数.py”为文件名保存在 Python 系统所在的文件夹中。

S3 选择菜单项: Run|Run Module, 运行程序并切换到 Python Shell 窗口。

S4 按 Python Shell 窗口中的提示输入 x 的值(如 10), 并按 Enter 键。

S5 查看运行结果, 关闭 Python Shell 窗口。

(3) 在 Windows 命令提示符窗口中, 编写并运行包含分支结构的 Python 程序。

S1 单击“开始”按钮, 选择菜单项: “程序”|“附件”|“记事本”, 打开 Windows 系统的记事本窗口。

S2 编辑求解分段函数

$$y = \begin{cases} 2x + 1 & (x < 0) \\ 0 & (x = 0) \\ -x & (x > 0) \end{cases}$$

的程序

```
x=float(input("x=?"))
```

```

if x>0:
    y=2 * x+1
elif x==0:
    y=0
else:
    y=-x
print("y=",y)

```

并以“分段函数.py”为文件名保存在系统盘(如 D 盘)的 Backup 文件夹中。

注: 这里的 if 语句也可以写成

```

if x>0:
    y=2 * x+1
else:
    if x==0:
        y=0
    else:
        y=-x

```

S3 打开 Windows 的“命令提示符”窗口,并切换到 Python 系统所在的文件夹。

S4 输入命令

```
python D:\Backup\分段函数.py ✓
```

启动 Python 解释器并运行保存在 D:\Backup\文件夹中的“分段函数.py”程序。

S5 按显示出来的提示信息,输入 x 的值并按 Enter 键。

```
python D:\Backup\分段函数.py ✓
```

S6 查看运行结果,然后按 Ctrl+Z,退出 Python 解释器。

S7 关闭 Windows 的“命令提示符”窗口。

3. 练习题

(1) 在 Python 编程环境中编写并运行程序: 输出由 26 个“#”字符围成的矩形图案,其中上、下两条边线各用 10 个“#”字符,左、右两条边线各用 5 个“#”字符。

(2) 在 Python 编程环境中编写并运行程序: 用户从键盘输入 x,计算如下分段函数的值并打印。

$$y = \begin{cases} x^2 & (0 \leq x \leq 1) \\ 2 - x & (1 < x \leq 2) \end{cases}$$

(3) 在 Windows“命令提示符”窗口中编写并运行程序: 计算运费

运费 = 重量(吨数) * 路程(公里数) * 折扣(按路程优惠)

其中,700 公里以上按 90%计算,1000 公里以上按 85%计算。

实验 2-3 程序中的循环结构

1. 实验任务与目的

(1) 在 Python 编程环境中,编写并运行求累加和

$$1+2+3+\cdots+N$$

的程序。体会具有循环结构的程序的特点。

(2) 在 Python 编程环境中,编写并运行求累加和

$$1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\frac{1}{5}-\cdots$$

并在当前项小于 0.001 时终止的程序。进一步理解循环结构的作用和特点。

2. 实验步骤

(1) 编写求累加和的程序。

S1 打开 Python Shell 窗口,然后打开 Python 编程窗口。

S3 编辑求解累加和

$$1+2+3+\cdots+N$$

的程序,其中 N 值在程序运行时由用户输入。

```
#用户输入 N 值,计算并输出 1+2+3+...+N 的值
#sum1n.py
N=int(input("请输入 N "))           #输入数值并转换为整数
Sum=0                                #输入数值并转换为整数
for i in range(1,N+1,2):             #对 [1, N+1) 中从 1 开始间隔为 1 的每一个整数
    Sum=Sum+i                         #累加当前 i 值
print("Sum=",Sum)                    #输出累加和
```

S4 选择菜单项: Run|Run Module,运行程序并切换到 Python Shell 窗口。

S5 按屏幕提示输入 N 的值并按 Enter 键。

S6 查看运行结果,然后退出 Python 编程环境。

(2) 修改程序,计算较为复杂的级数之和。

S1 打开 Python Shell 窗口,然后打开 Python 编程窗口。

S2 编辑求累加和

$$1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\frac{1}{5}-\cdots$$

并在当前项小于 0.001 时终止程序

```
#级数求和
```

```

Sum=0                #累加和置初值
t=1                  #当前项置初值
i=1                  #循环变量置初值
while 1/i>0.001:    #当前项的绝对值不小于0.001时,循环
    Sum=Sum+t        #累加当前项
    i=i+1            #循环变量加1
    t=-1/i           #计算下一项
print("Sum=",Sum)    #输出累加和

```

S3 选择菜单项: Run|Run Module,运行程序并切换到 Python Shell 窗口。

S4 查看运行结果,然后退出 Python 编程环境。

3. 练习题

(1) 比较本实验中的两个程序,用流程图或伪代码写出求累加和程序的通用算法。

(2) 编辑并运行程序(使用 for 循环): 用户输入 N 的值,计算并输出 $1+3+5+\dots+N$ (N 为偶数时不含 N) 的值。

(3) 编辑并运行程序: 用户输入 10 个整数,输出其中最大的数。

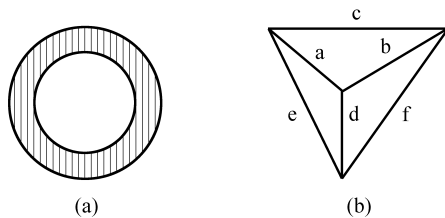
实验 2-4 函数的定义和调用

1. 实验任务与目的

(1) 在 Python 与用户的交互方式下,定义求圆面积的函数并两次调用它计算如实验图 2-5(a)所示的圆环面积。

(2) 编辑并运行程序: 定义求三角形面积的函数并 4 次调用它计算如实验图 2-5(b)所示的四面体的表面积。

通过本实验掌握函数定义和调用的一般方法,加深对子程序的形式和意义的理解。



实验图 2-5 圆环与四面体

2. 实验步骤

(1) 在 Python 与用户的交互方式下,计算圆环面积。

假设大圆和小圆的半径分别为 2.85 和 1.85,按以下步骤计算并输出圆环面积。

S1 输入语句:

```

def circleS(r):
    return 3.1415926 * r * r

```

定义求圆面积的函数。

S2 输入语句:

```

print("圆环的面积是",circleS(2.95)-circleS(1.85))

```