

# 程序设计基础 ——绪论

林大 经管学院 瞿华

# 自我介绍

- ❖ 瞿华 (拼音: Qu Hua)
- ❖ 联系方式: 13911168184,  
[quhua@bjfu.edu.cn](mailto:quhua@bjfu.edu.cn)
- ❖ 教学材料下载:
  - 学校教学平台(教务处网站, 点击左侧“教学平台”)

# 绪论

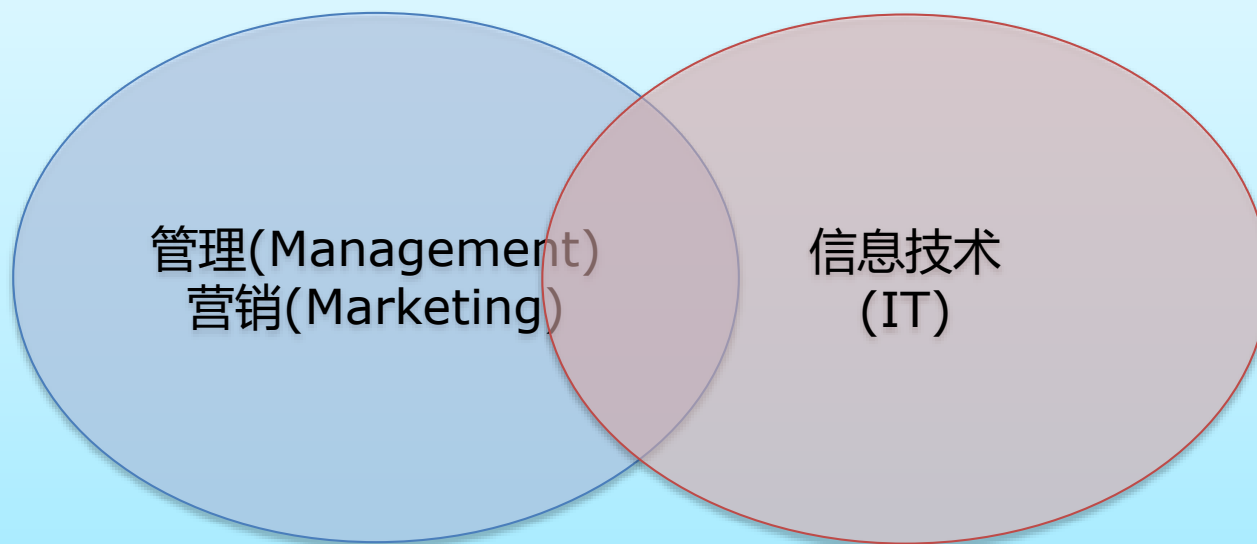
- 一. 课程简介
- 二. 程序设计基础知识
- 三. Python语言简介
- 四. 编写与运行Python程序
- 五. 小结

# 绪论

- 一. **课程简介**
- 二. 程序设计基础知识
- 三. Python语言简介
- 四. 编写与运行Python程序
- 五. 小结

# 一、课程简介

信息管理/电子商务



逻辑思维（数学等）/语言（汉语、英语）

**信管/电商专业核心知识体系**

# 1.1 课程定位

## ❖ 定位：

- 信息技术领域的基础课、入门课！
- 本课程学不好，后续的一系列课程都会很吃力！

## ❖ 名称：“程序设计基础(Python)”

- 核心是“程序设计基础”

## ❖ 核心内容：如何设计程序

- 程序：给计算机看，让计算机照着做的说明书
  - ✓ 必须用计算机能看得懂的语言来写

## ❖ 还需要顺带学一门计算机编程语言：Python语言

# 1.2 教学内容与目标

## ❖ 主要目标：

- 掌握基础的编程概念和术语
  - ✓ 编程是人和计算机交流的过程，也是人与人之间交流的过程
  - ✓ 用通用的术语来交流，更准确、顺畅
- 理解和熟练掌握基本的程序设计思维模式
  - ✓ 编程和作文、解数学题一样，有基本的套路
  - ✓ 熟练掌握这些基本的套路，是我们课程的核心内容
- 掌握Python语言的基础语法
  - ✓ 能用Python语言来表达编程的基本设计思路，编写简单的程序

# 1.2 教学内容与目标

## ❖ 次要目标:

- 了解计算机科学软件方面的主要应用和研究领域
  - ✓ 该领域要解决哪类问题?
  - ✓ 如何运用前述基本的程序设计思路解决这类问题?



# 1.3 考核方式

## ❖ 考核方式：

总成绩=平时成绩与考勤\*40%+考试\*60%

## ❖ 说明：

- 考勤、实验及最终考试都采用百分制
- 考勤：每缺勤一次（不论理由）平时成绩减一分。缺勤达8次以上者取消考试资格
- 课堂随机提问，回答正确者总成绩加分。

# 1.4 关于课程学习

- ❖ 为啥要学编程（如果我将来不想从事程序开发方面的工作？）
  - 提升分析问题、解决问题的能力
    - ✓ 写程序和工作中解决问题的基本思路是一致的
  - 培养和锻炼合理的安排工作的能力
    - ✓ 写程序就是告诉计算机第1步先做什么，第2步做什么。。。
    - ✓ 企业和经营管理不能光靠脑洞（战略）和情怀（营销宣传），关键在于一步一步的实干。
  - 更好的使用计算机工具，提高工作效率
    - ✓ 用VBA小程序来控制word、excel进行复杂的文字和数据处理等等

# 1.4 关于课程学习

- ❖ 喜欢计算机的人都“懒”
- ❖ 懒有两种：
  - 消极的懒：觉得太复杂太难就不做了
  - 积极的懒：想办法(利用工具,改进方法)让复杂的工作简单
- ❖ 学会编程是让我们积极偷懒的好帮手!
- ❖ 小例子：批量修改图像大小
- ❖ 小例子：名单比对

# 1.4 关于课程学习



要让娃去学编程吗？ x

← → ↻ 🏠

www.newsmth.net/nForum/#!article/ChildEducation/295333?p=7

🔍 📌 ⭐

应用

★ Bookmarks

📄 Save to Mend...

🚓 北京市公安局...

🌐 VOA Special E...

🔗 Open source a...

» 📁 其他书签

文章主题：要让娃去学编程吗？

分享到：

👍 🔄 📌

adrift

👤

💬 回复

📄 转寄

🔍 搜索

👤 只看此ID

第60楼

随风飘荡

身份

用户

文章

192

积分

22616

等级

楠木 (12)

发信人：adrift (随风飘荡)，信区：ChildEducation

标 题：Re: 要让娃去学编程吗？

发信站：水木社区 (Sun Feb 28 20:30:33 2016)，站内

我家男孩7岁半，酷爱玩Minecraft游戏，我就引导他学习python进行Minecraft编程，可以瞬间生成城堡、围栏等，基本学会。后来不满足于python对Minecraft的简单控制，想自己开发想要的模块，强烈要求学习java。目前基本学会了java的语法、数据类型、容器、类继承、读写文件、异常等，准备开始开发他需要的模块了。因为是自己玩游戏用，所以学习动力很强。

## 北京新闻2012-10-02

2012-10-02 | 网友评论0条



# 1.4 关于课程学习

## ❖ 不需要：

- 英语很好
- 数学很好

## ❖ 只要：

- 会四则运算
- 正常人的逻辑思维能力和记忆能力

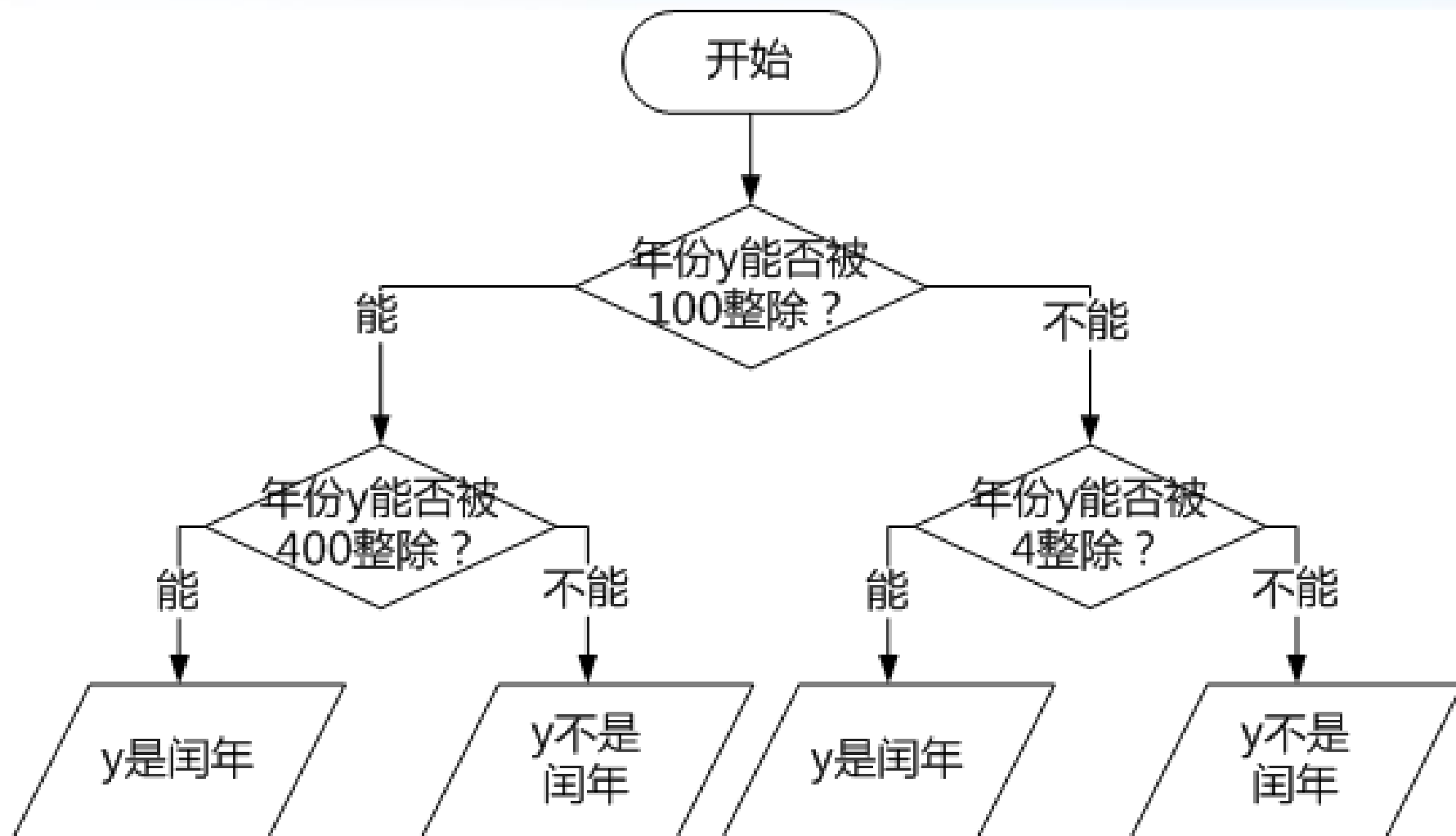
# 1.4 关于课程学习

❖ 例：如何判断年份 $y$ 是否是闰年？

❖ 闰年规则：

- 不能被4整除：非闰年
- 能被4整除但是不能被100整除：闰年
- 能被100整除但是不能被400整除：非闰年
- 能被400整除：闰年

# 1.4 关于课程学习





# 1.4 关于课程学习

```
if (y%100)==0:
    if (y%400)==0:
        print("y是闰年")
    else:
        print("y不是闰年")
else:
    if (y%4)==0:
        print("y是闰年")
    else:
        print("y不是闰年")
```

如果y能被100整除，那么：

如果y能被400整除，那么：

显示“y是闰年”

否则

显示“y不是闰年”

否则：

如果y能被4整除，那么

显示“y是闰年”

否则

显示“y不是闰年”

**程序的逻辑表达和人的  
正常思维逻辑是一致的。**

只是需要稍微习惯一下  
它所使用的语法格式。

# 1.5 对同学的要求

- ❖ 课后多练习!
- ❖ 课后多练习!
- ❖ 课后多练习!
- ❖ 课堂时间有限，必须依靠大家自己充分利用课余时间
  - 课上听懂了不等于你真的掌握了、记住了!
  - 前面不多练习熟练掌握的话，后面会慢慢跟不上的。

# 1.5 对同学的要求

## ❖ 练什么：

- 课后练习
- 课上的例子

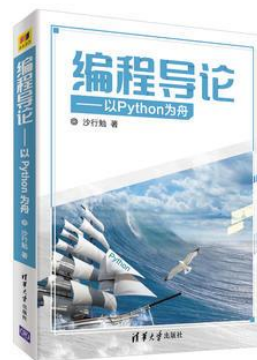
## ❖ 如何练

- 自己输入完整的程序
- **不要**直接复制\粘贴例子,然后在上面修改
- 直接写程序有困难,可以先手写草稿

## ❖ 多反馈、沟通

## 1.6 推荐教材

- ❖ 《编程导论——以Python为舟》沙行勉著。清华大学出版社



- ❖ 《程序设计导论——Python语言实践》。罗伯特·塞奇威克等著。机械工业出版社。



# 推荐阅读

- ❖ 编码：隐匿在计算机软硬件背后的语言  
[美]Charles Petzold著



# 1.7 关于试验课

## ❖ 内容：

- 检查（抽查同学们的课后练习情况），请同学记得来上课时带上所做的练习程序（U盘或网盘皆可）
  - ✓ 检查时程序读不出来，当没做处理
- 解决同学们课后练习时发现的问题。

# 绪论

- 一. 课程简介
- 二. **程序设计基础知识**
- 三. Python语言简介
- 四. 编写与运行Python程序
- 五. 小结

# 2.1 程序与语言

- ❖ **程序 (Program)** : 指挥计算机工作的一系列指令
- ❖ **程序设计语言** (Programming Language) : 用来描述程序的语言, 也称为程序设计语言。
  - 在不会混淆的情况下, 通常简称为语言。

```
sa=input("请输入整数a:");  
sb=input("请输入整数b:");  
a=int(sa);  
b=int(sb);  
c=a+b;  
print(c);
```

程序及程序设计语言示例 (python语言)



# 2.1 程序与语言

## ❖ 程序设计语言类型：

- **机器语言**：CPU能直接理解的数字指令。
- **汇编语言**：使用英文缩写（助记符）来表示的CPU指令。
- **高级语言**：用与人类语言（英语）接近的形式表示给计算机的命令
  - ✓ 主要用高级语言编程
  - ✓ C、Java、C#、Python

## ❖ 程序又可以分为两类：

- **可执行文件**：由机器语言组成，计算机可以直接执行的程序。
- **代码（Code）**：用除机器语言外的其他语言编写的程序文件。

# 2.1.1 机器和汇编语言

intel®



每个CPU有若干个引脚

引脚接电代表1，不接电代表0

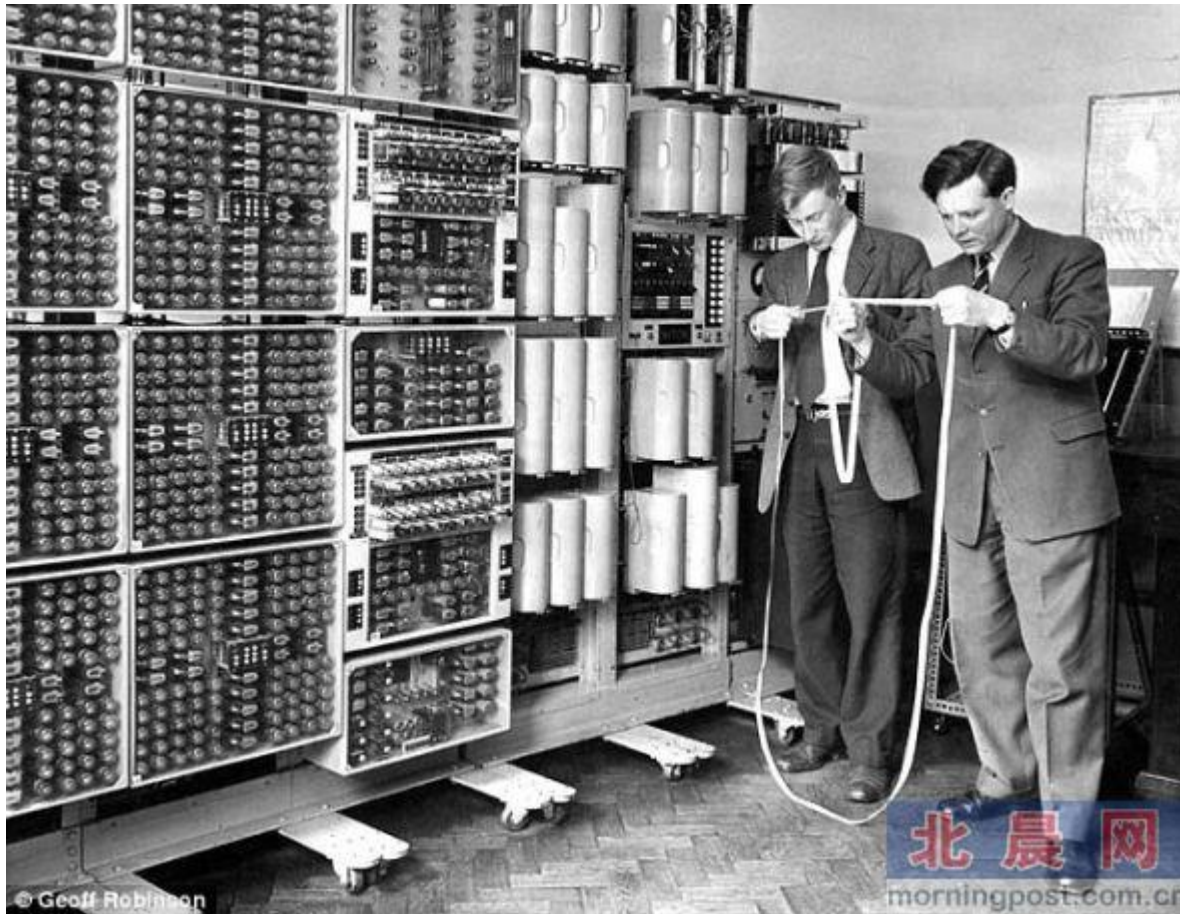
每种接电方式会导致CPU内部电路执行一个相应的处理，即执行一条指令

每个二进制数字（010101...）对应一组引脚的接电方式

机器语言就是直接用（二进制）数字表示的CPU接电指令

汇编语言就是用英文缩写单词来表示对应的CPU指令

## 2.1.1 机器和汇编语言



最早的语言——二进制机器语言，以纸带为存储载体

## 2.1.1 机器和汇编语言

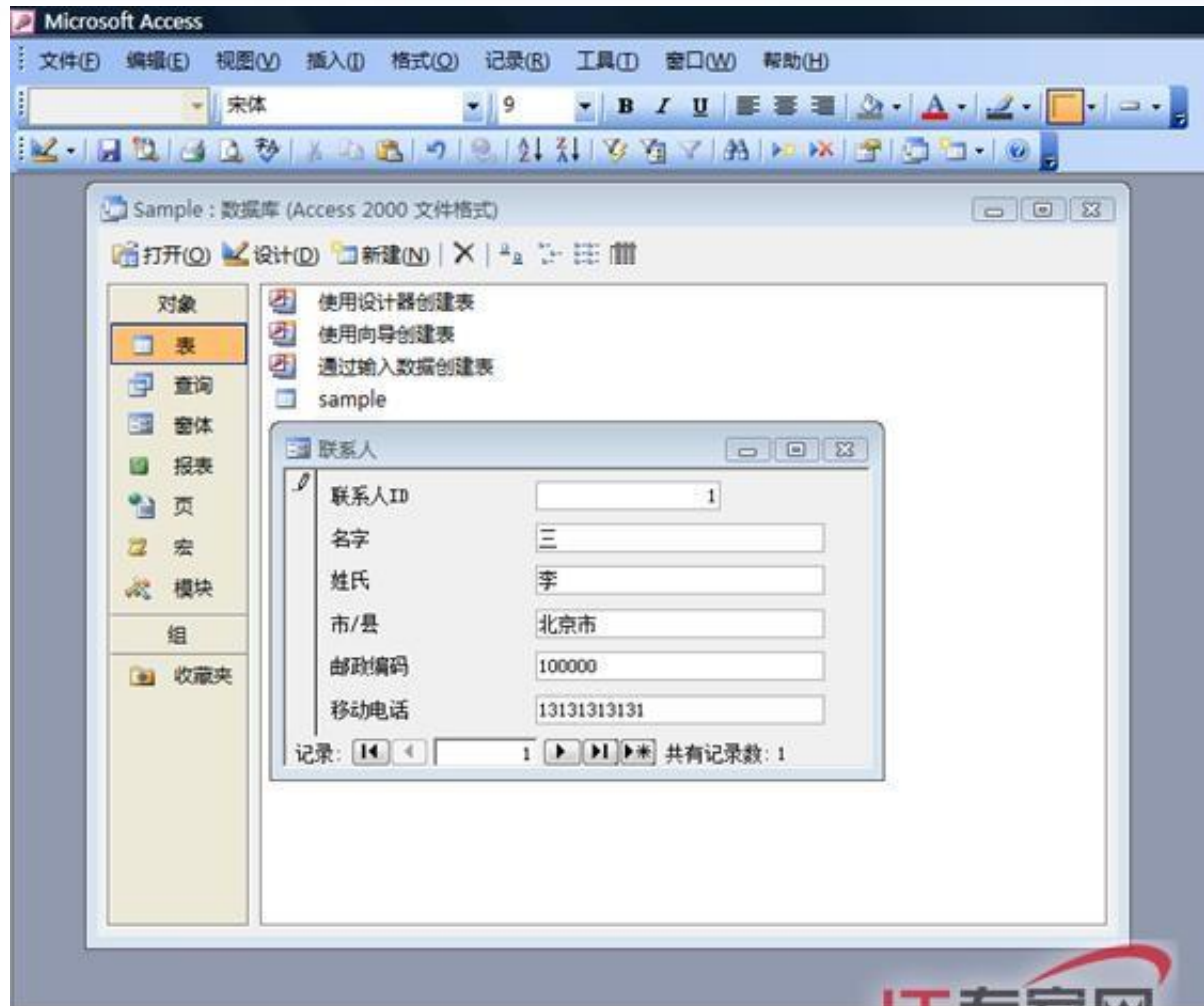
```
80483e2:      83 ec 24          sub    $0x24, %esp
80483e5:      8d 45 f4          lea    0xffffffff4(%ebp), %eax
80483e8:      89 44 24 04       mov    %eax, 0x4(%esp)
80483ec:      c7 04 24 10 85 04 08 movl   $0x8048510, (%esp)
80483f3:      e8 e0 fe ff ff    call   80482d8 <scanf@plt>
80483f8:      8d 45 f0          lea    0xffffffff0(%ebp), %eax
80483fb:      89 44 24 04       mov    %eax, 0x4(%esp)
80483ff:      c7 04 24 10 85 04 08 movl   $0x8048510, (%esp)
8048406:      e8 cd fe ff ff    call   80482d8 <scanf@plt>
804840b:      8b 55 f4          mov    0xffffffff4(%ebp), %edx
804840e:      8b 45 f0          mov    0xffffffff0(%ebp), %eax
8048411:      8d 04 02          lea    (%edx, %eax, 1), %eax
8048414:      89 45 f8          mov    %eax, 0xffffffff8(%ebp)
8048417:      8b 45 f8          mov    0xffffffff8(%ebp), %eax
804841a:      89 44 24 04       mov    %eax, 0x4(%esp)
804841e:      c7 04 24 13 85 04 08 movl   $0x8048513, (%esp)
8048425:      e8 be fe ff ff    call   80482e8 <printf@plt>
804842a:      b8 00 00 00 00    mov    $0x0, %eax
804842f:      83 c4 24          add    $0x24, %esp
```

机器语言（中）和汇编语言（右）示例

## 2.1.2 4GL

- ❖ 4GL：无需写代码，只要用鼠标拖拽和连接图形就能生成程序的语言。
  - 本质上不是语言，而是图形编辑与代码生成工具。（将图形转换成代码）
  - 典型4GL：VB, Access, Power Builder
  - 曾经流行一时。
  - 现在多作为代码辅助工具，而不是独立的语言。

# 2.1.2 4GL



4GL示例 (Access)

## ❖ 总结：程序语言的发展趋势

- 从贴近计算机硬件到贴近人类语言和思维

## ❖ 未来：

- 自然语言编程？

## 2.2 编译与解释

❖ 计算机运行非机器语言程序（代码）的两种方式：

- 编译执行
- 解释执行



# 2.2.1 编译

## ❖ 编译执行(先翻译再执行)：

1. 先将整个高级语言程序转化为对应的机器语言程序（编译）
  - ✓ 通常直接转换为可执行文件
2. 然后执行转化后的机器语言程序

## ❖ 优点：

- 编译后的可执行文件可以直接运行

## ❖ 例:编译运行00add.c（C语言程序）

# 2.2.1 解释

## ❖ 解释（边翻译边执行）：

1. 逐行将高级语言程序转换为对应的机器指令，并执行
2. 每转换完一行，就执行一行。

## ❖ 缺点：

- 每次运行都需要先解释，执行效率较低

## ❖ 例：解释执行“加法.py”(python语言程序)

# 绪论

- 一. 课程简介
- 二. 程序设计基础知识
- 三. **Python语言简介**
- 四. 编写与运行Python程序
- 五. 小结

# 3.1 Python语言的优点

## ❖ 优点1：基础语法简单

- Python语言的基本语法非常简单，python 3.7只有35个关键字（简单地说就是系统可以理解的词语）。
- Python语言使用最常用的英语单词或单词的简写来作为关键字，关键字的长度大多在3-5个字母左右。非常简单好记。比如：
  - ✓ if else
  - ✓ for
  - ✓ while
  - ✓ import from
  - ✓ def
- 在本课程中我们会学习的关键字只有20个左右，常用的10多个。

# 3.1 Python语言的优点

## ❖ 优点2：动态类型，python vm自动决定变量类型，简单省事

- Python是动态类型语言，不需要先声明变量再使用（C、Java等语言需要先声明变量及其类型再使用）
  - ✓ Python类似于食堂，去了找个座位就可以吃
  - ✓ C和Java等像医院，需要先挂号，拿到挂号条才能去找大夫看病

```
#include <stdio.h>
int main(){
    int a,b,c;
    scanf("%d",&a);
    scanf("%d",&b);
    c=a+b;
    printf("%d",c);
    return 0; }
```

C语言程序

```
a = input("请输入整数a:")
b = input("请输入整数b:")
a = int(a)
b = int(b)
c = a + b
print(c)
```

Python语言程序  
(3-1.add.py)

# 3.1 Python语言的优点

## ❖ 优点3：语法糖多，能够写出更加简练的程序

- 语法糖（Syntactic sugar指计算机语言中添加的某种语法，这种语法没有为语言增加额外的功能，但能让程序员写出更加简洁的程序。（语言中的“快捷方式”）
- 例：计算列表[1,2,3,4,5,6]中所有奇数的平方

```
lst1=[1,2,3,4,5,6]

lst2=[]
for x in lst1:
    if x%2==1:
        lst2.append(x**2)

print(lst2)
```

标准写法：使用循环遍历计算

```
print(lst2)

lst3=[x**2 for x in lst1 if
x%2==1]

print(lst3)
```

Python特有语法：列表推导式  
(3-2.语法糖.py)

# 3.1 Python语言的优点

## ❖ 优点4：开源第三方库函数丰富

- Python自带的标准库中包含200多个模块，上千个函数，涵盖了文本处理、时间处理、数学计算、文件系统操作、网络操作等多种功能
- 除此之外，Python Package Index( <https://pypi.org>)网站还提供了由开源社区用户提供的16多万个第三方模块项目，可以自由使用
- 意味着很多功能不需要我们去写程序实现，只要调用已有的实现即可！

# 3.1 Python语言的优点

## ❖ 优点4：开源第三方库函数丰富

从网络上抓取股票价格信息(3-3.查询股票信息)

```
from urllib import request
import easygraphics.dialog as dlg

code = "sh601006"
response = request.urlopen('http://hq.sinajs.cn/list=' +
code)
raw_data = response.read().decode('gbk')
quote1 = raw_data.find('"')
quote2 = raw_data.find('"', quote1 + 1)
raw_data = raw_data[quote1 + 1:quote2]
datas = raw_data.split(',')
result = "股票{0}目前的价格是{1},最高价{2},最低价
{3}".format(datas[0], datas[3], datas[4], datas[5])
dlg.set_dialog_font_size(14)
dlg.show_message(result, "股票信息")
```



## 3.2 Python语言的缺点

### ❖ 缺点1：速度慢

- 例：分别用python和C实现一个 $1+2+3+\dots+n$ 的函数（模块），各计算1000次1到10万的和，看看各自所用的时间（见"3-4.运算性能"示例程序）
- **python本身的运算速度比C或Java要慢一个数量级！**
- python不适合用于对性能有要求的场合
- 也可以通过将耗时的操作用C开发，python调用C写好的模块来解决性能问题（如常用的scipy、numpy库）
- 尽量使用python自带的函数实现（如sum()）

## 3.2 Python语言的缺点

- ❖ **缺点2：解释执行，在客户端运行必须提供给用户源代码，难以保护知识产权**
- ❖ **缺点3：动态类型，无法做静态检查，由于类型错误导致的运行问题要到运行时才能发现(3-5.参数类型检查)**
  - 类比，去不用挂号的医院看感冒。找到一个有空的医生，坐下就诊。医生说，“我是看妇科病的！”
  - C和Java在编译时即可发现函数调用的参数类型错误（不符合契约），python不行
  - 在复杂系统或者多人协作开发时，容易出问题
- ❖ **缺点4：语法糖和高级用法多，学习曲线长**

## 3.3 适用场合

- ❖ 每种语言都有其各自的优缺点，和适用的场合
- ❖ Python适用于各种**无**复杂业务逻辑、**非**大规模团队协作开发、**不**需要担心知识产权或设计机密泄露的场合
  - 个人自己使用的工具小程序（脚本,Script）
  - 数据分析处理（统计、大数据、数据挖掘）
  - 科研计算
  - 原型(Prototype)或快速开发
  - 企业运维（负责管理和维护企业的计算机和信息系统）
  - 大型复杂系统中的胶水语言(Glue Language)

## 3.3 适用场合

- ❖ **（界面交互）原型**：给用户评估用的示范软件，通常只是用户界面交互和最终系统基本一致，而实际功能并未真正实现，
- ❖ **快速开发系统**：为了能够快速给客户提供一个能用（或者能进行评估）的系统，而开发出来的简化版软件；与正式的系统相比，通常在非核心功能/异常情况处理/性能/安全性/可用性/可扩展性等方面做出了牺牲。
- ❖ **胶水语言**：由于企业的业务规则（如，图书馆的借阅规则；超市的促销）经常变化，相关的系统中的程序逻辑也需要随之变化。利用Python简便不需要编译的特点，用C/C++或Java开发系统主要功能模块，而用python来编写业务规则代码，调用各模块完成业务处理。业务规则变化时，只需要修改python代码即可。
  - 例如：超市促销，这周可能是打9折，下周可能是满100减20；打x折和满x减y的基本处理方法是不变的，用C/Java来编写对应的模块；但这周到底该打折，还是满减，打折到底是9折还是85折，由python脚本程序来负责。
  - 在这样的程序结构中，python就像是胶水，把各个模块连接成一个完整的系统，因此称为胶水语言

## 3.4 Python 2与3

- ❖ Python 3版本与2版本相比，有重大的变化，二者的程序不完全兼容。
- ❖ Python 3最重要的变化之一是完善了对中文等非英语语言的支持（Unicode支持）
- ❖ 显然，我们应该选择使用Python 3.x版本

## 3.5 中文文档

- ❖ 这里有完整的python 3.5.2中文文档：
  - [https://yiyibooks.cn/xx/python\\_352/index.html](https://yiyibooks.cn/xx/python_352/index.html)
- ❖ 其中“库参考”一节中提供了python标准库各模块和函数的详细说明，在使用遇到疑问时，可以到这里查找。

# 绪论

- 一. 课程简介
- 二. 程序设计基础知识
- 三. Python语言简介
- 四. **编写与运行Python程序**
- 五. 小结

# 4.1 Python交互环境

- ❖ Python有两种执行模式：
  - 缺省模式（解释执行模式）：执行完整一个完整的python程序
  - 交互模式：用户输入一行，执行一行，同时显示本行的执行结果
- ❖ 后者适合的情况：
  - 简单计算处理（计算器）
  - 探索式数据分析
  - 对程序设计思路进行实验
- ❖ 注：ipython和qtconsole提供了功能更强大的交互处理环境（需要预先安装）；在命令行输入jupyter qtconsole启动qtconsole



## 4.2 集成编译环境

- ❖ “工欲善其事，必先利其器”
- ❖ 高效率的开发工具应该不仅能够帮助我们来编写源代码，还应该提供：
  - 好用的文本编辑功能（拷贝粘贴，查找替换等等）
  - 好用的代码辅助编辑功能（语法高亮显示，智能提示、语法错误提醒等）
    - ✓ Word、记事本显然不具备这些功能
  - 编译、连接源代码，以及运行编译好的程序
  - 方便的调试功能等等
    - ✓ 断点和单步执行

## 4.2 集成编译环境

- ❖ **集成开发环境** (Integrated Development Environment, 简称**IDE**) 就是具备上述功能的一套图形界面的程序开发工具软件。
- ❖ 目前最常用的Python语言IDE有:
  - Spyder
  - WingIDE
  - PyCharm
  - Visual Studio Code

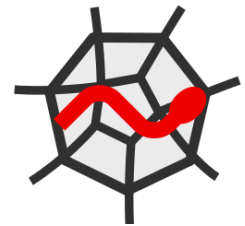
## 4.3 Spyder



- ❖ Spyder是一个适用于探索式数据分析（科研）或者脚本小程序开发的IDE，其最大的特点是同时集成了交互式执行环境和程序编辑执行环境
- ❖ 用户可以一边进行探索式分析试验，一边编写完善程序

## 4.3 Spyder

- ❖ Spyder是一个适用于探索式数据分析（科研）或者脚本小程序开发的IDE，其最大的特点是同时集成了增强的交互式执行环境和程序编辑执行环境
- ❖ 用户可以一边进行探索式分析试验，一边编写完善程序



## 4.4 WingIDE

- ❖ WingIDE是一个相对轻量级的IDE
- ❖ 与Spyder相比:
  - (缺点) 只提供了最基本的交互式执行环境
  - (优点) 提供了更强的实时智能提示和调试功能
  - (优点) pro版提供了重构等更多强大的功能
- ❖ 适用于中小型软件项目开发

## 4.5 第一个python程序

```
yourName=input("你叫什么名字? ")  
print(yourName+",你好! ")
```

# 五 小结

- ❖ 学习软件开发的意義？
- ❖ 最终成绩如何计算？（有申请免修的同学吗？）
- ❖ 本课程的学习目的？
- ❖ Python语言有哪些优缺点？适用场合？
- ❖ Python语言的相关概念：源代码、编译等
- ❖ 什么是集成编译环境？

# 课后练习

- ❖ 按照"A1.Python开发环境安装"说明，安装相关软件
- ❖ 尝试：
  - 在python的交互环境（模式）中输入和执行第一个程序
  - 使用Spyder,pycharm或者WingIDE编辑第一个程序，然后完整执行
- ❖ **使用WingIDE的同学，请务必按照"A1.Python开发环境说明"中第5.2.1节中的说明，修改WingIDE的缺省文件编码为utf-8。否则python程序中的中文内容在运行时会变成乱码！**