



Chap 1 Programming & Python

第1章 程序设计和Python语言

Department of Computer Science and Technology
Department of University Basic Computer Teaching

1.1

程序设计概述

- **计算机中的程序**是指按特定的算法用某种计算机语言描述完成指定任务的处理过程。
- 程序语言**分类**

**按对机器依赖性、
接近自然语言的程度、
语言的发展过程**

{ 机器语言
汇编语言
高级语言

- CPU能理解且能直接执行的指令集合
- 用**机器语言**编写的程序形式：

10110000

00001101

00101100

00000101

11110100

依赖于具体计算机

难记忆

易出错

阅读理解困难

编程者需了解特定计算机的硬件及指令系统

- 汇编语言是一种符号化的机器语言，用助记符号代替指令操作码、操作数。
- 用**汇编语言**编写的程序形式：

MOV AL,13D

SUB AL,5D

HLT



汇编语言与机器语言的异同？

- 高级语言接近于自然语言和**数学语言**，在一定程度上与具体计算机无关的符号化语言。
- 用**高级语言**编写的程序形式（以C语言为例）：

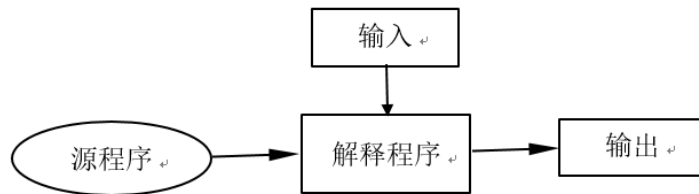
```
#include <stdio.h>
int main(void)
{
    printf("%d\n",13-5);
    return 0;
}
```

易学易用
代码易理解
易调试、修改和移植，并与机器无关

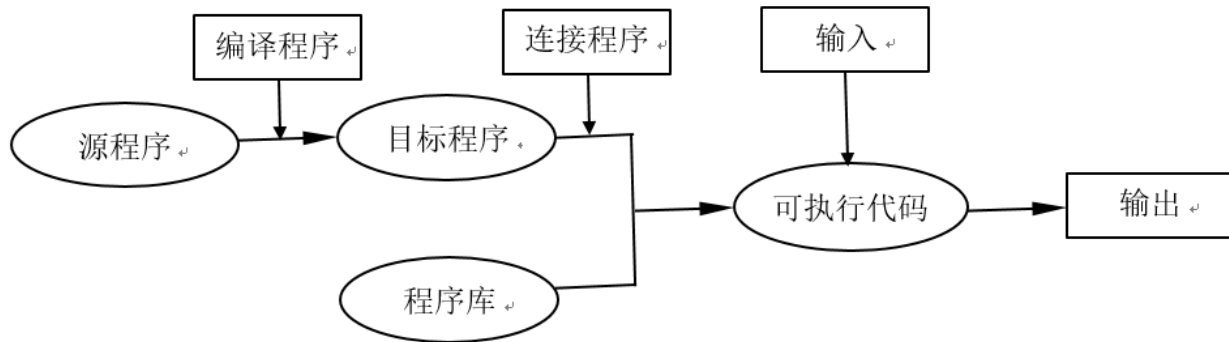
不支持对硬件的直接操作
代码需要翻译指令序列后才能执行

- 高级语言的翻译方式

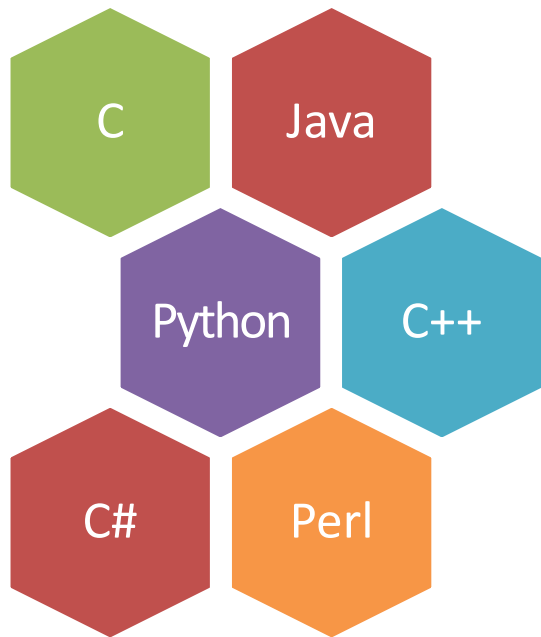
- 解释方式 (Interpreter 解释程序)



- 编译方式 (Compiler 编译程序或称编译器)



程序设计语言



- 就程序本质而言
 - 刻画计算（数值计算与非数值计算，与处理同义）过程的处理对象（数据）与处理规则（算法）。
- 瑞士计算机科学家 Niklaus Wirth认为：
 - **程序 = 算法 + 数据结构**

程序 = 算法 + 数据结构

10

- **算法**是解决问题的步骤和方法，是指令的有限序列
- **数据结构**则是数据的组织形式，涉及操作对象以及它们之间的关系和操作

计算 $1+2+\dots+100$ 的和，算法和数据结构设计：

S1 : $\text{sum} \leftarrow 0$

S2 : $i \leftarrow 1$

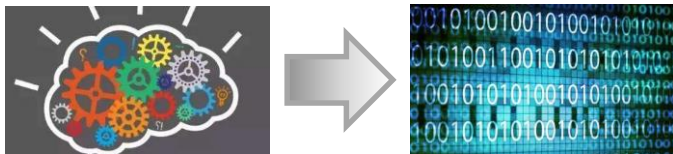
S3 : 如果 $i \leq 100$ 执行 $\text{sum} \leftarrow \text{sum} + i$ ，如果 $i > 100$ 转 S5

S4 : $i \leftarrow i + 1$ ，转 S3

S5 : 输出 sum

程序 = 数据结构 + 算法

- 算法是程序的**灵魂**，在程序设计中具有**核心**地位
- 实际应用中用户首先关心的是自己的数据能够被处理，其次才关心如何处理，用什么工具处理数据，所以从应用角度出发，程序设计的**重心**已转移到**数据**上了



- 结构化程序设计
 - 自顶向下 (top-down) , 逐步细化(stepwise refinement) , 由抽象到具体的功能分解过程 , 使用三种控制结构即顺序、选择和循环控制构造程序

- 面向对象程序设计
 - 是目前计算机开发中使用的主流方法，它是一种对现实世界理解并抽象的方法，将对象作为程序的基本单元，并将程序和数据封装在内，以提高软件的重用性、灵活性和扩展性。

```
类 狗狗
{
    过程 跑步;
    过程 打滚;
    .....
}
```

```
类 吉娃娃 继承自 狗狗
{
    过程 卖萌;
    过程 咬人;
    .....
}
```

一只叫
旺财的
吉娃娃

1.2

PYTHON简介

Python指数——PYPL

17

Worldwide, Aug 2017 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Java	22.7 %	-0.7 %
2		Python	16.3 %	+3.7 %
3		PHP	8.9 %	-1.1 %
4		C#	8.3 %	-0.5 %
5		Javascript	8.0 %	+0.5 %
6		C++	6.6 %	-0.2 %
7		C	6.4 %	-0.7 %
8	↑	R	3.6 %	+0.4 %
9	↓	Objective-C	3.6 %	-1.2 %
10		Swift	2.8 %	-0.3 %



编程语言流行指数 ([PyPL](#))

Python指数——TIOBE

18

Aug 2017	Aug 2016	Change	Programming Language	Ratings	Change
1	1		Java	12.961%	-6.05%
2	2		C	6.477%	-4.83%
3	3		C++	5.550%	-0.25%
4	4		C#	4.195%	-0.71%
5	5		Python	3.692%	-0.71%
6	8	▲	Visual Basic .NET	2.569%	+0.05%
7	6	▼	PHP	2.293%	-0.88%
8	7	▼	JavaScript	2.098%	-0.61%
9	9		Perl	1.995%	-0.52%
10	12	▲	Ruby	1.965%	-0.31%
11	14	▲	Swift	1.825%	-0.16%
12	11	▼	Delphi/Object Pascal	1.825%	-0.45%
13	13		Visual Basic	1.809%	-0.24%

Python指数——IEEE Spectrum

19

Language Types (click to hide)



Web



Mobile



Enterprise

























Embedded

Language Rank

Types

Spectrum Ranking

1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.4
4. C++	  	97.2
5. C#	  	88.6
6. R		88.1
7. JavaScript	 	85.5
8. PHP		81.4
9. Go	 	76.1
10. Swift	 	75.3

Python应用实例



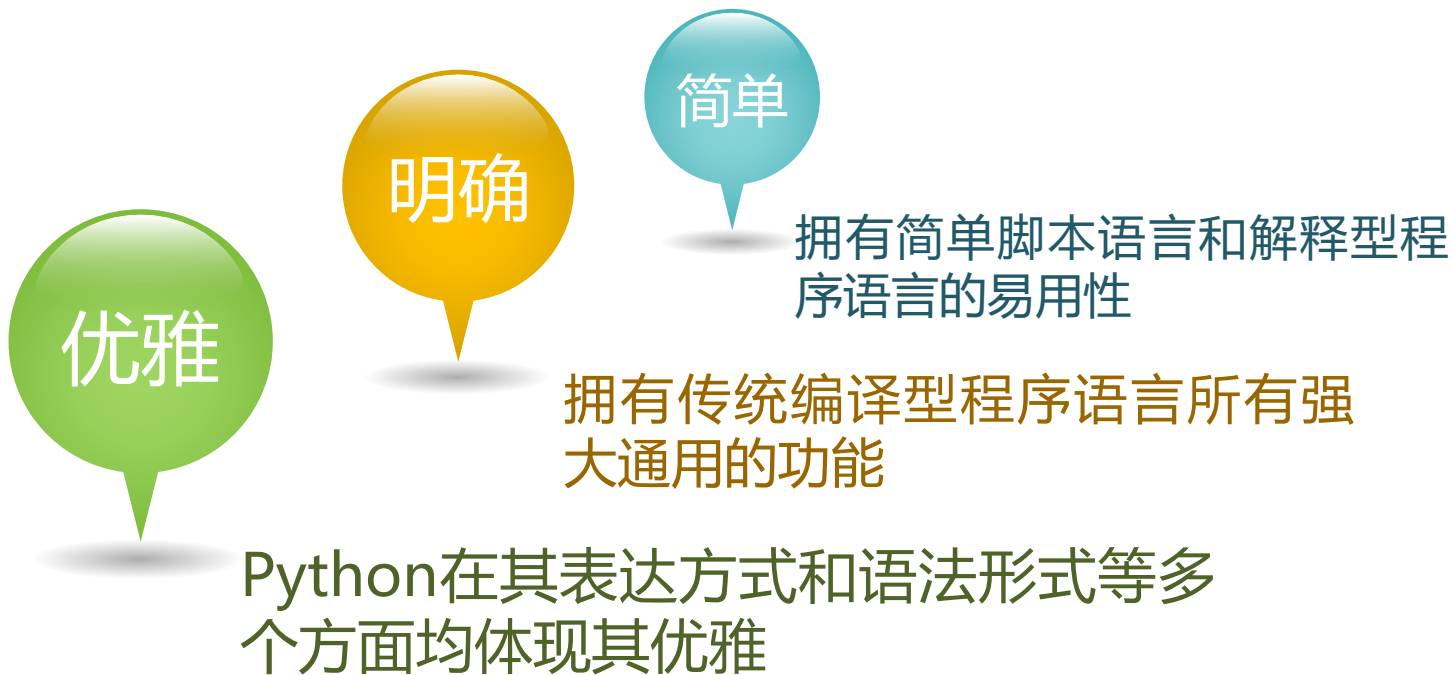
1.2.1 PYTHON的历史和特性

python 的诞生



Guido van Rossum

- 第1个Python编译器/解释器于1991年诞生
- Python名称来自Guido挚爱的电视剧Monty Python's Flying Circus
- Python介于C和Shell之间、功能全面、易学易用、可扩展



- 胶水语言 Glue Language

- 很容易和其他著名的程序语言连接（C/C++），集成封装

- 脚本语言 Script Language

- 高级脚本语言，比脚本语言只能处理简单任务强大

- 面向对象语言 Object-Oriented Language

- 完全支持继承、重载、派生、多继承

Python的库——具有强大吸引力

25



Python提供各种库，具有极其便捷强大的数据处理和统计功能

python提供的re模块能支持正则表达式



Python的库——具有强大吸引力



Python定义了WSGI标准应用接口来协调http服务器与基于Python的Web程序之间的沟通

用wxPython或者PyQt来开发跨平台的桌面软件



Python的库——具有强大吸引力



操作系统

大多数Linux发布版以及NetBSD、OpenBSD和Mac OS X都集成了Python，Python标准库包含了多个调用作业系统功能的库

用于将计算机游戏中图形渲染模块封装为Python扩展类库



多媒体

1.2.2 PYTHON的版本

Python的2个版本

29

A blue folder icon with a white label "Python 2.x". An orange tab is visible at the top right of the folder.

Python 2.x

A blue folder icon with a white label "Python 3.x". An orange tab is visible at the top right of the folder.

Python 3.x



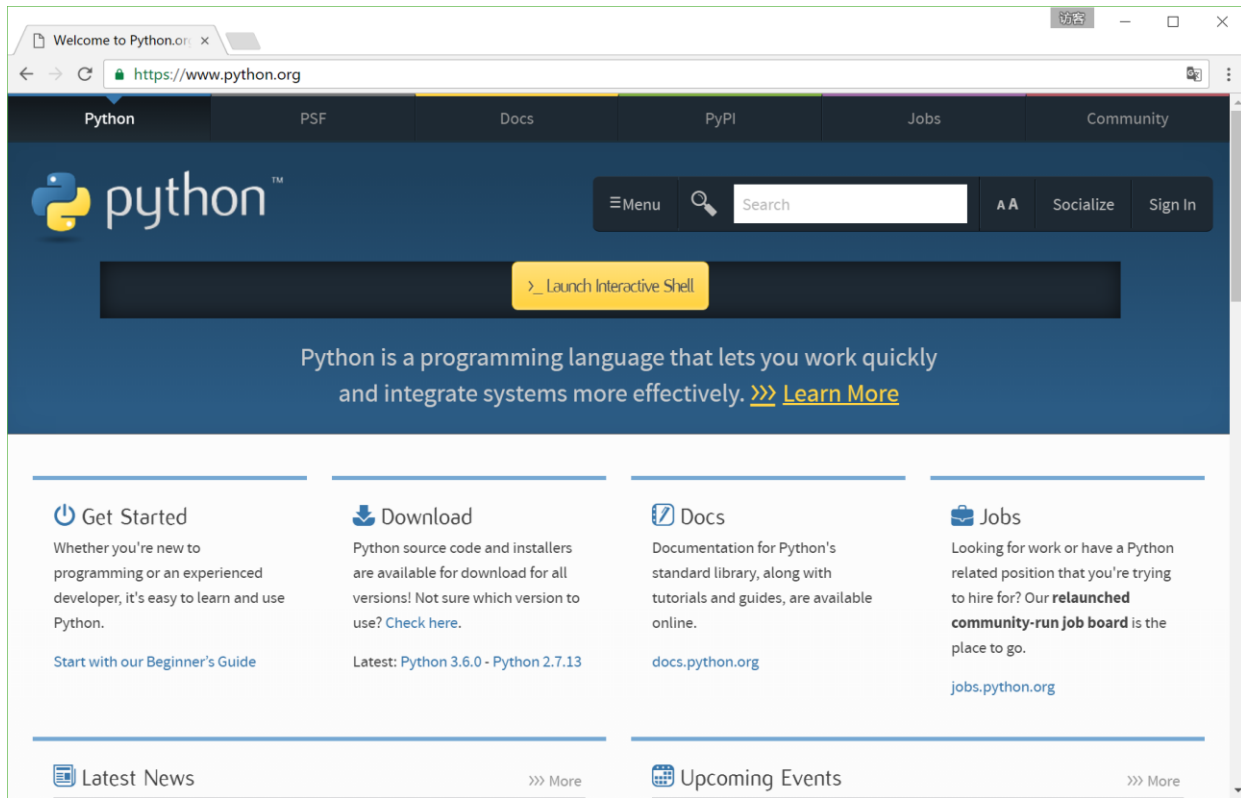
互不兼容

本书以Python3.x为主



Python官方网站

30



Python 格言

The Zen of Python

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than **right** now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

```
>>> import this
```

by Tim Peters



Alex Martelli

2002 Activators' Choice Award和2006 Frank Willison award为Google开发商业智能软件



Daniel Greenfeld

之前在美国宇航局做开发，目前是Cartwheel Web的负责人



Miguel Grinberg

为Harmonic做视频软件。C++是主要语言，但用Python写的自动化单元测试框架更很有趣

1.3

PYTHON开发 环境和运行方式

经典 Hello World

34

```
myString = 'Hello, World!'
```

```
print(myString)
```

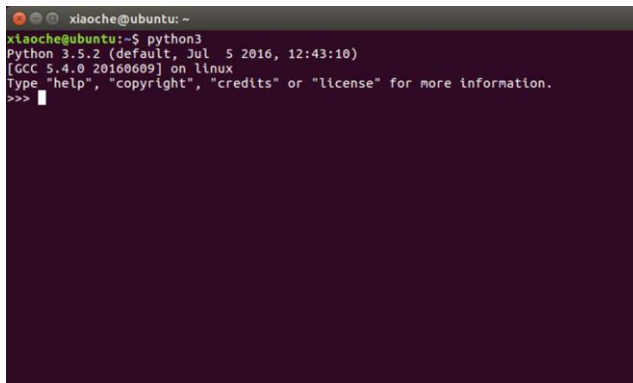
1.3.1 PYTHON开发环境

Python 集成开发环境 (IDE)

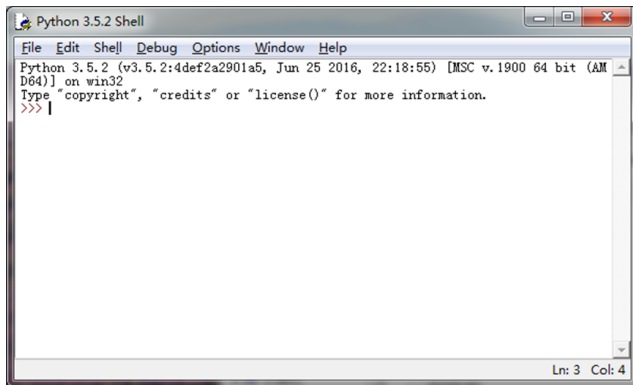
Python IDE

- Mac OS & Linux下
 - \$ python
 - \$ python3
- Python内置IDE
 - IDLE

(<https://www.python.org/ftp/python/3.5.2/python-3.5.2-amd64.exe>)
- 其他IDE
 - Ipython
 - PyCharm



```
xiaochu@ubuntu: ~  
xiaochu@ubuntu:~$ python3  
Python 3.5.2 (default, Jul 5 2016, 12:43:10)  
[GCC 5.4.0 20160609] on linux  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>
```



```
Python 3.5.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>
```

PyPI提供软件包索引

- Python的特点是有极其丰富的软件包（第三方库）
- PyPI (the Python Package Index) 是Python官方提供的软件包索引，它可以给现有的超过十万个Python软件包提供索引

(<https://pypi.python.org/pypi>)

包的安装

安装包

- 用pip命令安装包（大多数Python IDE中已集成，无需额外安装）

① 下载 get-pip.py

(<https://pip.pypa.io/en/latest/installing/>)

② 在操作系统终端依次执行如下命令

```
> python get-pip.py
```

```
> pip install atx
```

- **Anaconda集成开发平台**

- 下载安装包

- 官网：<https://www.continuum.io/downloads>

- 或国内清华和豆瓣等镜像网站

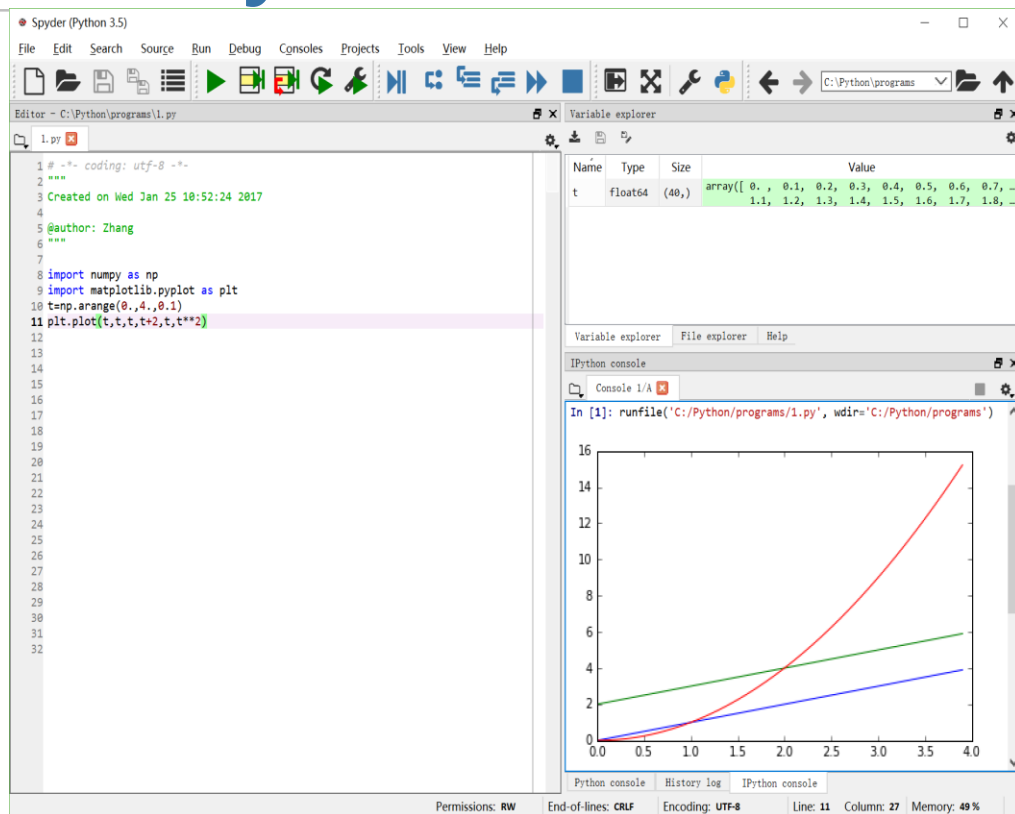
- 清华：<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>

- 安装(Mac OS and Linux , 版本号为例)

- **command line installer in MacOS:** `bash Anaconda3-4.3.0-MacOSX-x86_64.sh`
 - **graphical installer in MacOS:** download the graphical installer .pkg and follow the instructions
 - **command line installer in Linux:** `bash Anaconda3-4.3.0-Linux-x86_64.sh`

Python开发平台

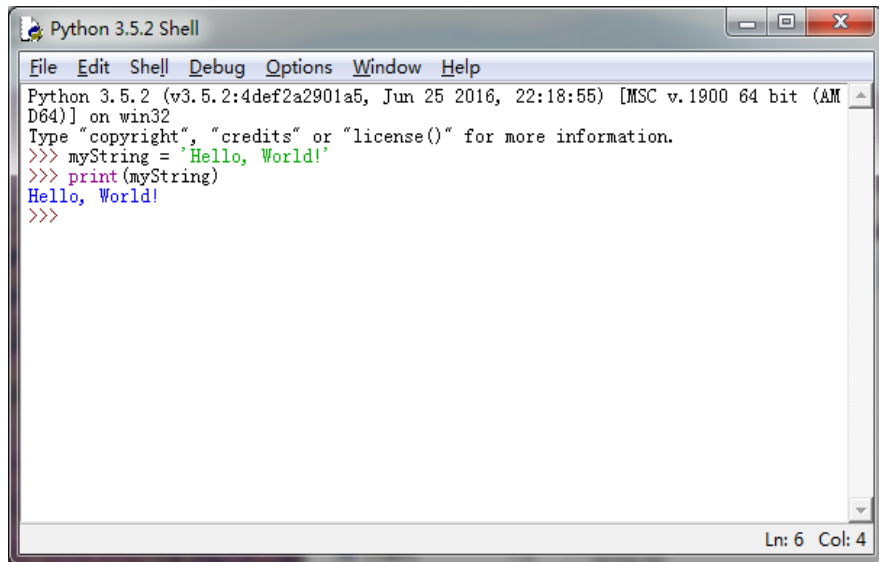
40



1.3.2 PYTHON运行方式

Python的运行方式（一）

Shell方式



The screenshot shows a 'Python 3.5.2 Shell' window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The text area contains the following text:

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> myString = 'Hello, World!'
>>> print(myString)
Hello, World!
>>>
```

The status bar at the bottom right indicates 'Ln: 6 Col: 4'.

- Shell是交互式的解释器
- 输入一行命令，解释器就解释运行出相应结果

文件执行方式

- 在Python的IDE环境中，创建一个以py为扩展名的文件
- 用Python解释器在Shell中运行出结果



经典 Hello World



```
>>> myString = 'Hello, World!'
>>> print(myString)
Hello, World!
>>> myString
'Hello, World!'
```



```
# Filename: helloworld.py
myString = 'Hello, World!'
print(myString)
```

Shell方式和文件执行方式

- 都是解释运行方式
 - 如果整个代码段较短，则优先考虑选择用Shell交互方式
 - 如果代码段较长，则建议使用文件执行方式



1.4

小结

- 程序设计语言
- 程序设计
- Python简介
- Python开发环境
- Python运行方式

