

Python语言程序设计



第1章 程序设计基本方法





计算机的概念



计算机的概念

- 计算机是根据指令操作数据的设备
- 具备功能性和可编程性两个基本特性

功能性指对数据的操作，表现为数据计算、输入输出处理和结果存储等。

可编程性指它可以根据一系列指令自动地、可预测地、准确地完成操作者的意图



计算机的概念

- 计算机硬件所依赖的集成电路规模按照摩尔定律以指数方式增长
- 计算机运行速度也接近几何级数快速增加
- 计算机所能高效支撑的功能不断丰富发展
- 程序设计语言经历了从机器语言到高级语言的发展过程，朝着更接近自然语言的方向发展。



摩尔定律 (Moore's Law)

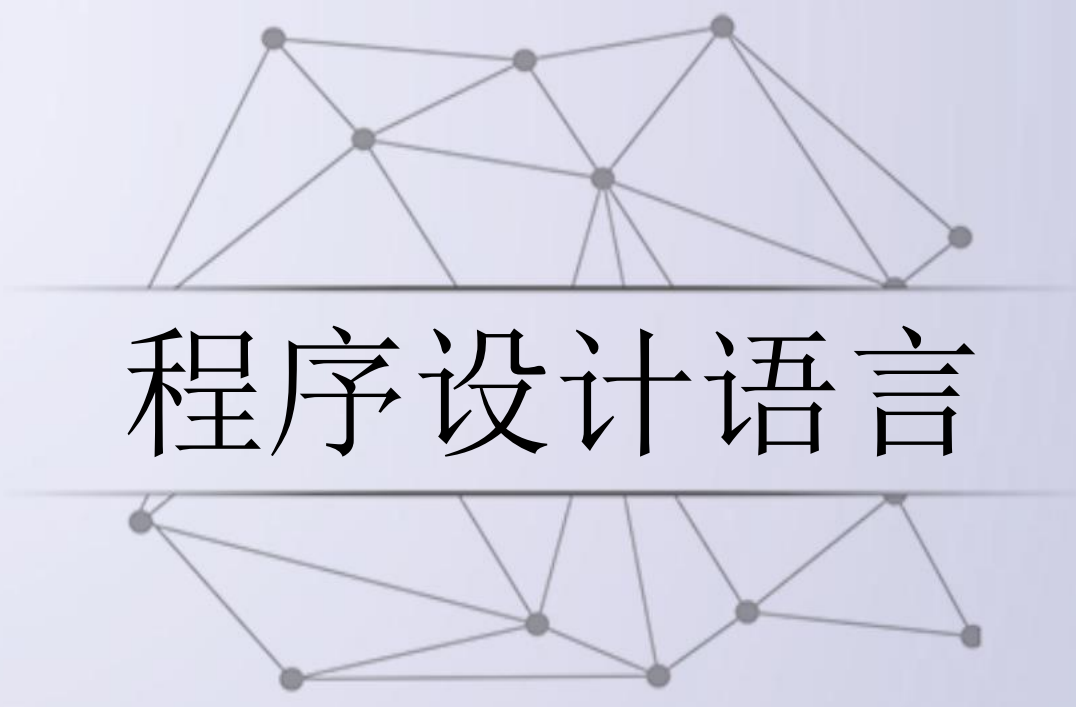
- 摩尔定律是计算机发展历史上最重要的预测法则

摩尔定律指出，单位面积集成电路上可容纳晶体管的数量约每两年翻一倍。由于计算机中几乎所有重要部件都由集成电路实现



计算机技术发展水平的时代性

- 第一阶段：1946年到1981年，“计算机系统结构阶段”
- 第二阶段：1982年到2007年，“计算机网络和视窗阶段”
- 第三阶段：2008年至今，“复杂信息系统阶段”。
- 第四阶段：约20年后未来的某个时期开始，“人工智能阶段”



程序设计语言



程序设计语言概述

■ 程序设计语言包括编译执行和解释执行两种方式

程序设计语言是计算机能够理解和识别用户操作意图的一种交互体系，它按照特定规则组织计算机指令，使计算机能够自动进行各种运算处理。按照程序设计语言规则组织起来的一组计算机指令称为计算机程序。



程序设计语言概述

■ 机器语言

机器语言是一种二进制语言，它直接使用二进制代码表达指令，是计算机硬件可以直接识别和执行的程序设计语言。

例如：执行数字2和3的加法，16位计算机上的机器指令为：11010010 00111011，不同计算机结构的机器指令不同



程序设计语言概述

■ 汇编语言

使用助记符与机器语言中的指令进行一一对应，在计算机发展早期帮助程序员提高编程效率

例如：执行数字2和3的加法，汇编语言指令为：

add 2, 3, result, 运算结果写入result。

- 机器语言和汇编语言都直接操作计算机硬件并基于此设计，所以它们统称为低级语言。



程序设计语言概述

■ 高级语言

高级语言区别于低级语言在于，高级语言是接近自然语言的一种计算机程序设计语言，更容易地描述计算问题并利用计算机解决计算问题。

例如：执行数字2和3加法的高级语言代码为：

```
result = 2 + 3
```



程序设计语言概述

- 超级语言 粘合整合已有程序。具备庞大计算生态
- 高级语言 接近自然语言，编译器，与CPU型号无关
- 汇编语言 有助记符，汇编器，与CPU型号有关
- 机器语言 代码直接执行，与CPU型号有关



编译和解释

高级语言按照计算机执行方式的不同可分成两类

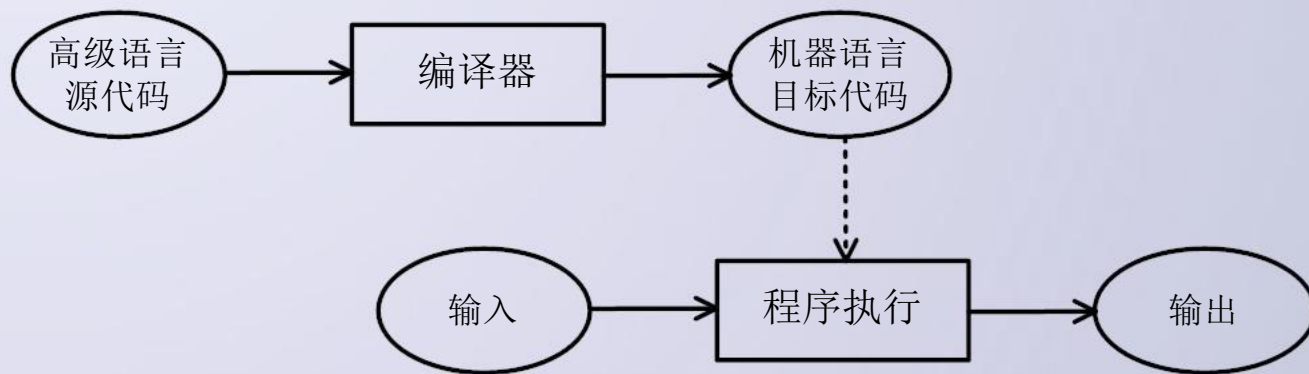
- 静态语言

- 脚本语言

这里所说的执行方式指计算机执行一个程序的过程，静态语言采用编译执行，脚本语言采用解释执行。

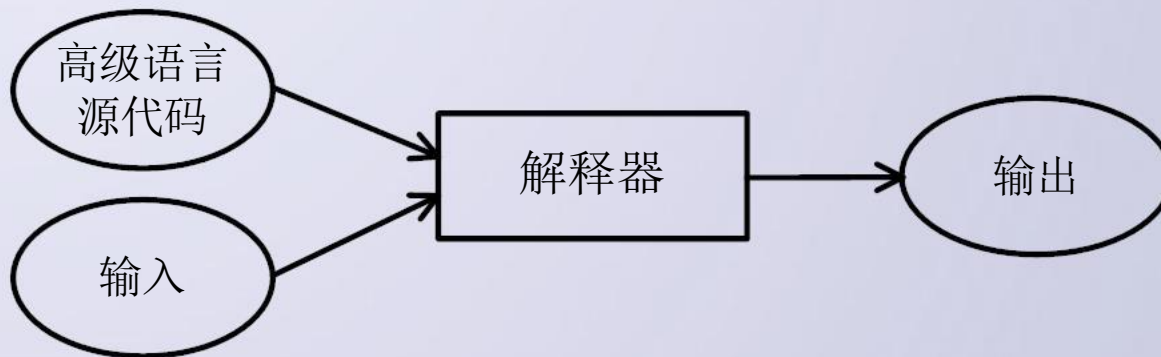
编译和解释

- 编译是将源代码转换成目标代码的过程，通常，源代码是高级语言代码，目标代码是机器语言代码，执行编译的计算机程序称为编译器



编译和解释

- 解释是将源代码逐条转换成目标代码同时逐条运行目标代码的过程。执行解释的计算机程序称为解释器。





编译和解释

编译是一次性地翻译，一旦程序被编译，不再需要编译程序或者源代码。

- 对于相同源代码，编译所产生的目标代码执行速度更快。
- 目标代码不需要编译器就可以运行，在同类型操作系统上使用灵活。



编译和解释

解释则在每次程序运行时都需要解释器和源代码。

- 解释执行需要保留源代码，程序纠错和维护十分方便。

- 只要存在解释器，源代码可以在任何操作系统上运行，可移植性好



计算机编程

——为什么要学习计算机编程？

——因为“编程是件很有趣的事儿”！



计算机编程

计算思维是区别于以数学为代表的逻辑思维 and 以物理为代表的实证思维的第三种思维模式。

编程是一个求解问题的过程

- 首先需要分析问题，抽象内容之间的交互关系
 - 设计利用计算机求解问题的确定性方法，
 - 进而通过编写和调试代码解决问题
- 这是从抽象问题到解决问题的完整过程。



Python语言概述

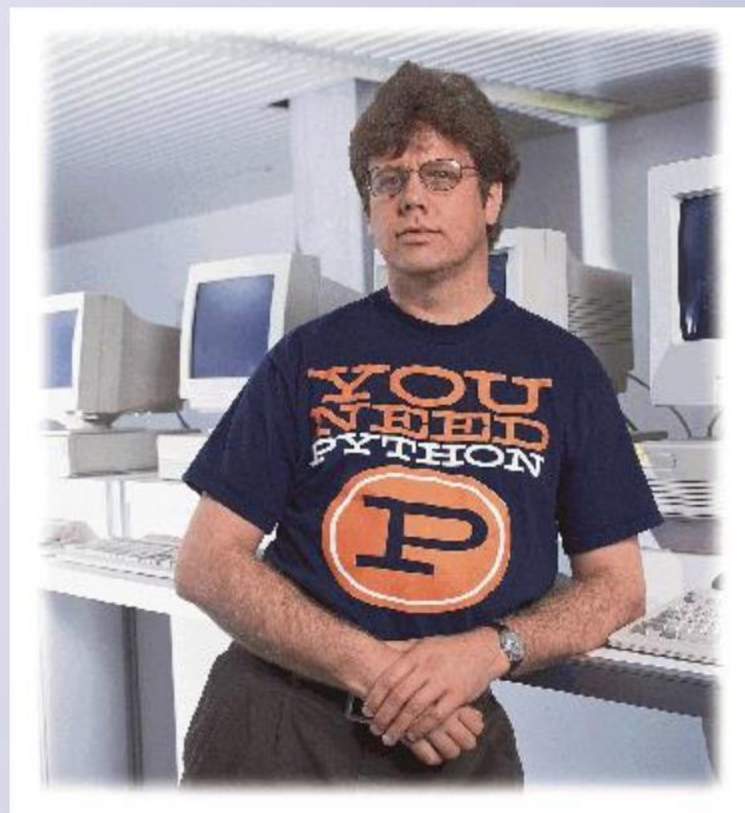
Python语言的诞生

Guido van Rossum

Python语言创立者

2000年, Python 2.x

2008年, Python 3.x





编写Hello程序

使用Python语言编写的Hello程序只有一行代码

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

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

第一行的“>>>”是Python语言运行环境的提示符

第二行是Python语句的执行结果



C语言的Hello程序

```
#include <stdio.h>
int main(void)
{
    printf("Hello World\n");
    return 0;
}
```

一般来说，同样功能的程序，Python语言实现的代码行数仅相当于C语言的1/5至1/10，简洁程度取决于程序的复杂度和规模。



Python语言的优势

脚本语言 + 语句执行

例1:

```
print("Hello World! 大家好!")
```

例2:

```
sum = 99999 * 99999  
print(sum)
```



Python语言的优势

例3:

```
months="JanFebMarAprMayJunJulAugSepOctNovDec"  
n = 4  
monthAbbrev = months[(n-1)*3:(n-1)*3+3]  
print(monthAbbrev)
```



Python语言的优势

简洁 + 强制可读性

例4:

```
def mean(numbers) :  
    s = 0.0  
    for num in numbers:  
        s = s + num  
    return s / len(numbers)  
  
nums = [0,1,2,3,4,5,6,7,8,9]  
print(mean(nums))
```



Python语言的优势

跨平台 + 开源

<http://pypi.python.org/>

目前有**93561**个开源库，覆盖各类计算问题

例5:

```
from random import random
rnd = random()*10
print(rnd)
```



Python语言的优势

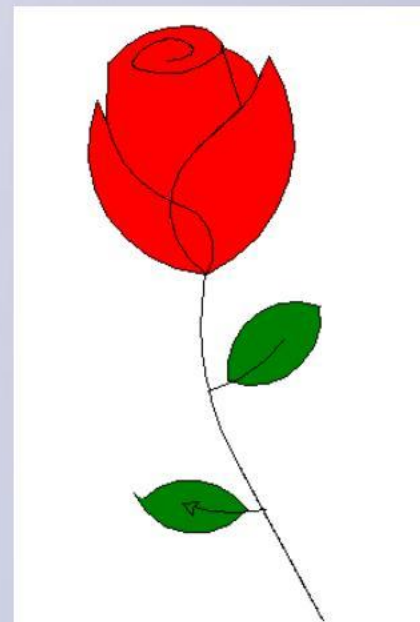
Python语言的优势：面向过程 + 面向对象

灵活的介绍程序设计理念

Python语言的优势：图形界面

Windows窗口

玫瑰花





Python语言特点

- Python语言是通用语言
- Python语言是脚本语言
- Python语言是开源语言
- Python语言是跨平台语言
- Python语言是多模型语言



Python开发环境配置



- 到Python主页下载并安装Python基本开发和运行环境，网址：www.python.org/downloads/
- 根据操作系统不同选择不同版本
- 下载相应的Python 3.0系列版本程序

安装

[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

Download the latest version for Windows

[Download Python 3.4.3](#)[Download Python 2.7.10](#)

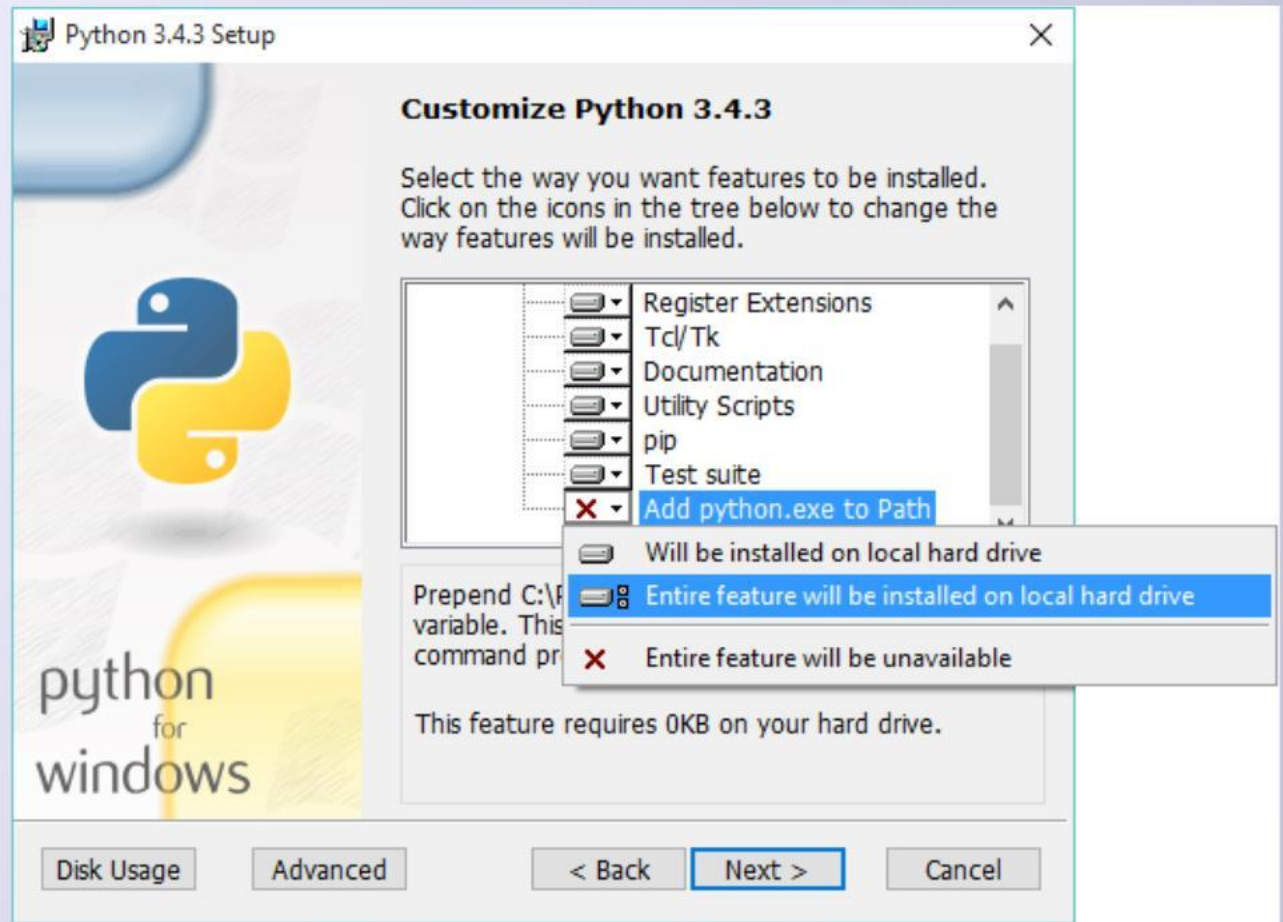
Wondering which version to use? [Here's more about the difference between Python 2 and 3.](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#)



安装





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
>>>
```



启动

- 方法2：调用IDLE来启动Python图形化运行环境

```
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
>>>
```



- **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。

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



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



- 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



Hello程序

采用上述某个方法，执行：

```
>>> print("Hello World")
Hello World
>>> print("世界，你好")
世界，你好
>>>
```



程序的基本编写方法



IPO程序编写方法

- 输入数据
- 处理数据
- 输出数据



IPO程序编写方法

■ 输入数据

输入（Input）是一个程序的开始。程序要处理的数据有多种来源，形成了多种输入方式，包括：文件输入、网络输入、控制台输入、交互界面输出、随机数据输入、内部参数输入等。



IPO程序编写方法

■ 处理数据

处理（Process）是程序对输入数据进行计算产生输出结果的过程。计算问题的处理方法统称为“算法”，它是程序最重要的组成部分。可以说，算法是一个程序的灵魂。



IPO程序编写方法

■ 输出数据

输出（Output）是程序展示运算成果的方式。程序的输出方式包括：控制台输出、图形输出、文件输出、网络输出、操作系统内部变量输出等。



问题的计算部分

一个待解决问题中，可以用程序辅助完成的部分

- 计算机只能解决计算问题，即问题的计算部分
- 一个问题可能有多种角度理解，产生不同的计算部分
- 问题的计算部分一般都有输入、处理和输出过程



IPO程序编写方法

■ 微实例1.1圆面积的计算

输入：圆半径radius

处理：计算圆面积 $area = \pi * radius * radius$ （此处， π 取3.1415）

输出：圆面积area



编程解决问题的步骤

6个步骤 (1-3)

- 分析问题：分析问题的计算部分，**想清楚**
- 划分边界：划分问题的功能边界，**规划IPO**
- 设计算法：设计问题的求解算法，**关注算法**



使用计算机解决问题

6个步骤 (4-6)

- 编写程序：编写问题的计算程序，**编程序**
- 调试测试：调试程序使正确运行，**运行调试**
- 升级维护：适应问题的升级维护，**更新完善**



求解计算问题的精简步骤

3个精简步骤

- 确定**IPO**：明确计算部分及功能边界
- 编写程序：将计算求解的设计变成现实
- 调试程序：确保程序按照正确逻辑能够正确运行



使用计算机解决问题

- 分析问题分析问题的计算部分
- 划分边界划分问题的功能边界
- 设计算法设计问题的求解算法
- 编写程序，编写问题的计算程序
- 调试测试，调试和测试程序
- 升级维护，适应问题的升级维护



Python语言版本更迭



Python语言的版本更迭

- 更高级别的3.0系列不兼容早期2.0系列
- 2008年至今，版本更迭带来大量库函数的升级替换，Python语言的版本更迭痛苦且漫长
- 到今天，Python 3.x系列已经成为主流



本章小结

本章具体讲解了计算机的基本定义、计算机的功能性和可编程性、程序设计语言分类、编译和解释、**Python**语言的历史和发展、配置**Python**开发环境等内容，最后给出了**Python**版本的主要区别供参考。



单元小结

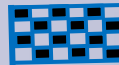


CC BY-NC-SA 4.0 嵩天



程序设计基本方法

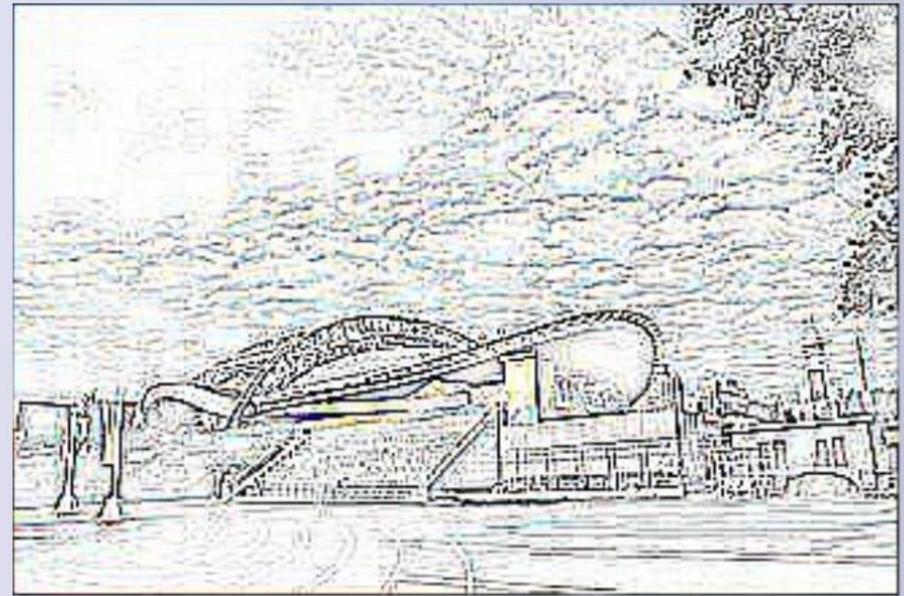
- 计算机的功能性和可编程性
- 编译和解释、静态语言和脚本语言
- **IPO**、理解问题的计算部分
- 掌握计算机编程的价值



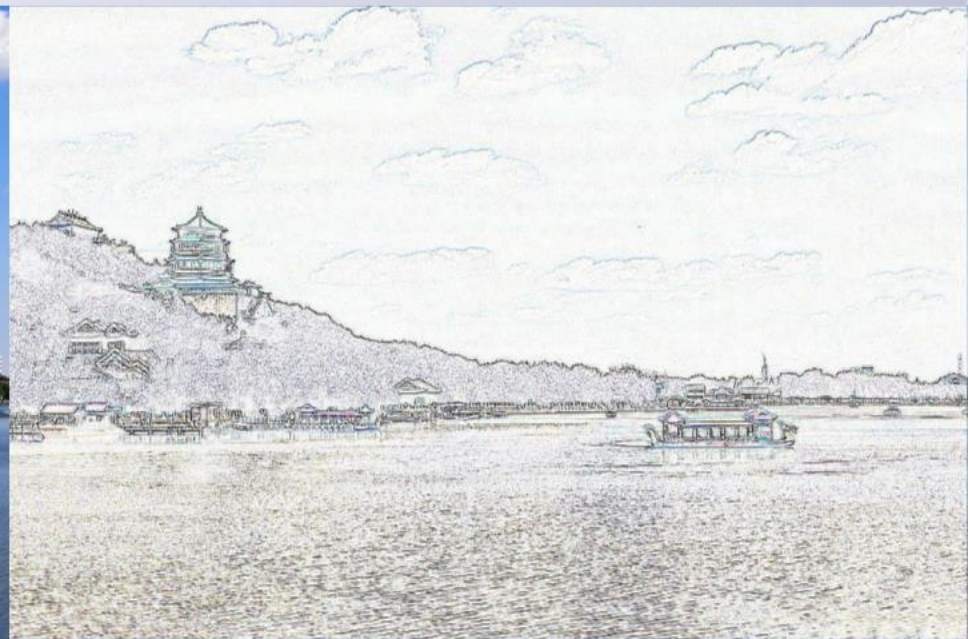
python
语言程序设计

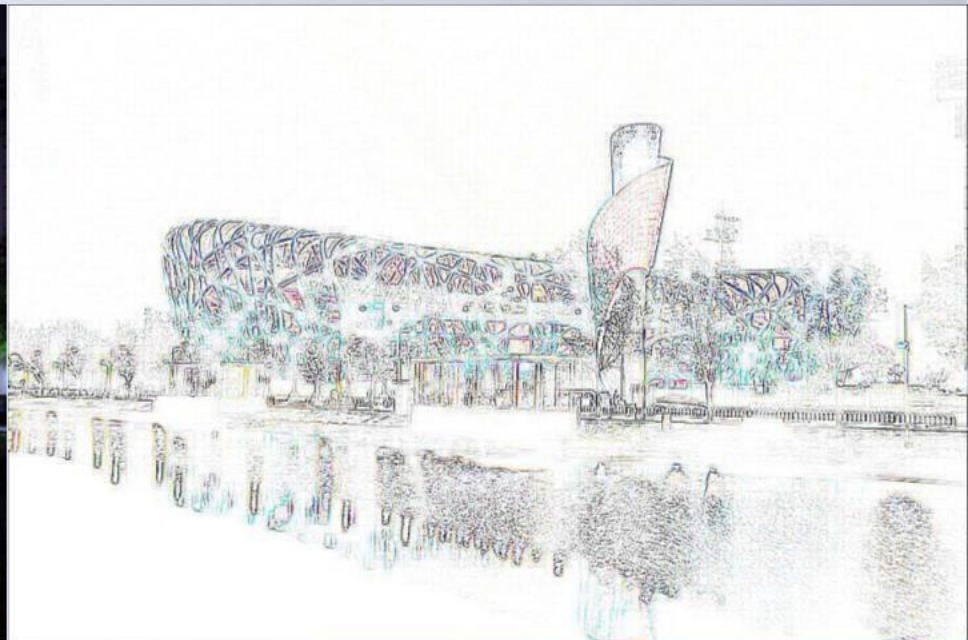


Python实例展示

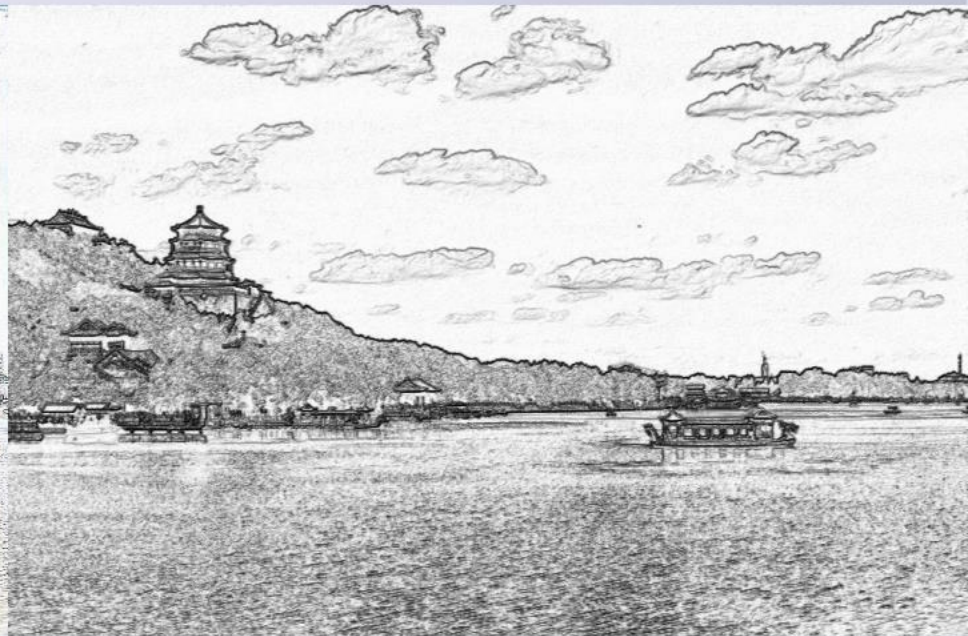
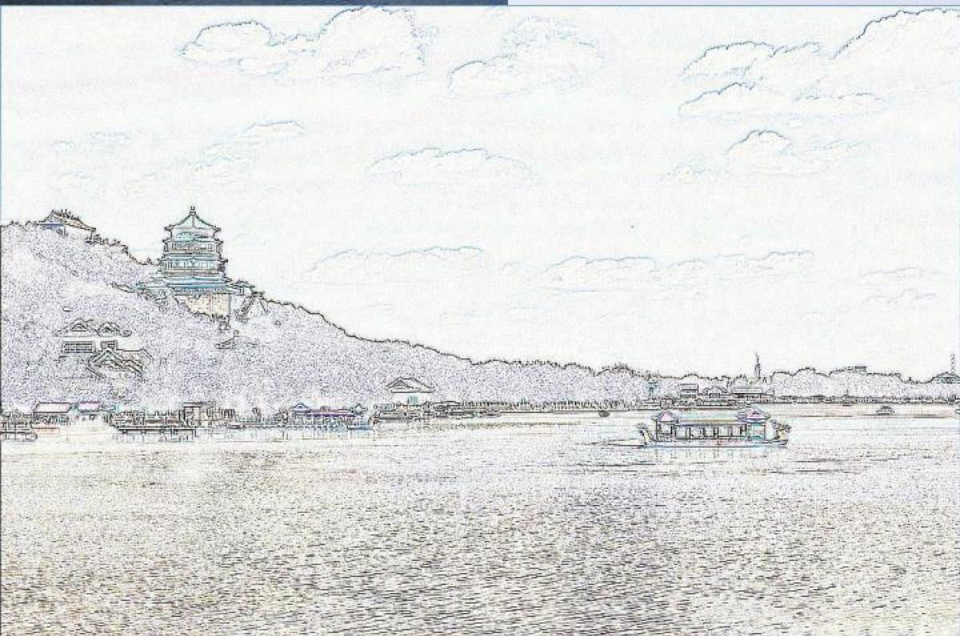


```
from PIL import Image
from PIL import ImageFilter
im = Image.open('np.jpg')
e33 = im.filter(ImageFilter.CONTOUR)
e33.save('E(33).jpg')
```

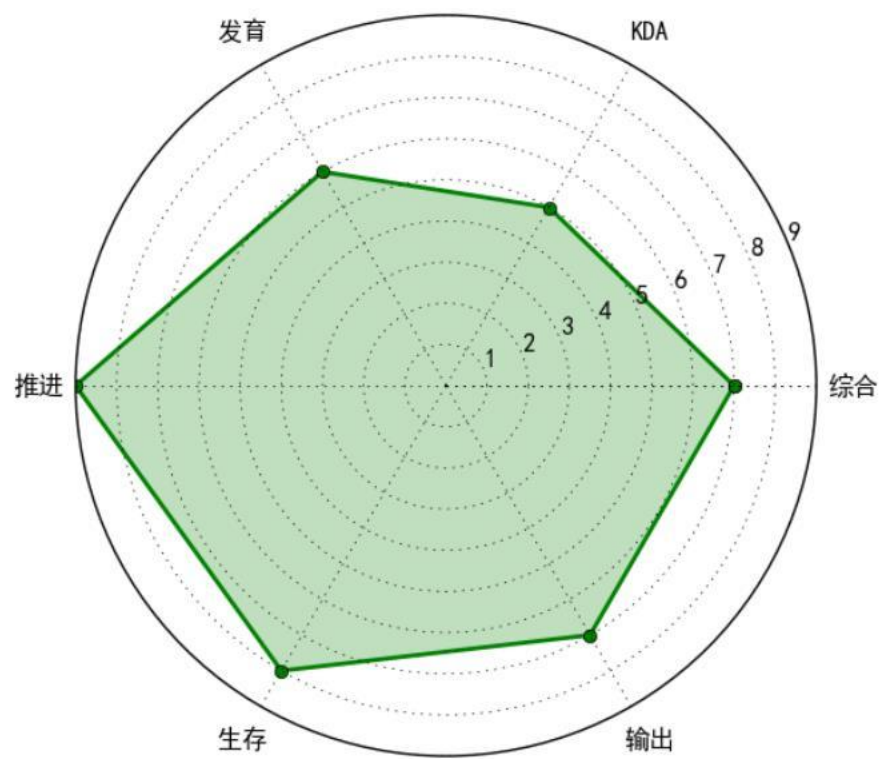









DOTA能力值雷达图



霍兰德人格分析 (Self-Directed Search)

