

# Python程序语言入门与应用



python

Life is short, use Python  
人生苦短，我用Python



# 课程计划



部分	章节	授课教师
初识Python	第1章 Python语言概述与开发环境配置	郭长江
	第2章 Python程序实例解析	郭长江
深入Python	第3章 Python基本数据类型	郭长江
	第4章 Python程序控制结构	郭长江
	第5章 Python函数与模块	王海蛟
	第6章 Python组合数据类型	王海蛟
	第7章 Python文件操作	王海蛟
	第8章 Python程序设计方法	王海蛟



# 课程计划



部分	章节	授课教师
运用Python	第9章 Python科学计算和可视化	王海蛟
	第10章 Python爬虫与web开发	王海蛟
	第11章 Python图形用户界面 (GUI)	王海蛟
	第12章 Python手机应用开发	王海蛟
	第13章 Python办公自动化	郭长江
	第14章 Python机器学习	郭长江
	第15章 Python面向对象编程	郭长江
	第16章 Python其他应用场景	郭长江



# 课程考核



## ☞ 能力和素质 (40%)

☞ 考勤、上课提问、课堂讨论

☞ **课后作业与实践**

## ☞ 期末考试 (60%)

☞ 开卷笔试，以讲授的内容为主要考试内容范围



# Python与中国教育



## 🔗 2018年起

- 🔗 Python取代VB成为浙江省信息技术教材编程语言，将纳入高考内容之一
- 🔗 北京和山东也确定要把Python 编程基础纳入信息技术课程和高考的内容体系
- 🔗 山东省出版的小学信息技术六年级教材加入了 Python 内容
- 🔗 Python被列入全国计算机等级考试二级范围





新乡医学院

## Python程序语言入门与应用

# 第一章 Python语言概述与开发环境配置



郭长江

changjiangguo@xxmu.edu.cn

生命科学技术学院

新乡医学院



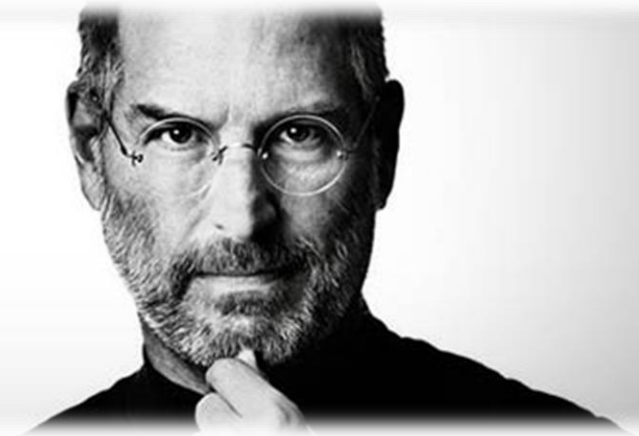
该二维码7天内(9月5日前)有效, 重新进入将更新



# Why?

**Everybody in this country should  
learn to program a computer...  
because it teaches you how to think**

Steve Jobs, co-founder and CEO of Apple Inc. (1955 - 2011)






# 学习目标



## 基本要求

### 掌握

-  Python语言开发和运行环境的配置方法

### 理解

-  Python语言的特点

-  编写程序的IPO方法

### 了解

-  Python语言版本更迭过程和新旧版本的主要区别





# 本课概要



## 第1章 Python语言概述与开发环境

- 1.1 计算机的概念
- 1.2 程序设计语言
- 1.3 Python语言概述
- 1.4 Python语言开发环境配置
- 1.5 程序的基本编写方法
- 1.6 Python语言的版本更迭



# 1.1 计算机的概念



## 计算机是根据指令操作数据的设备

- 功能性

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

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



超级计算机



服务器



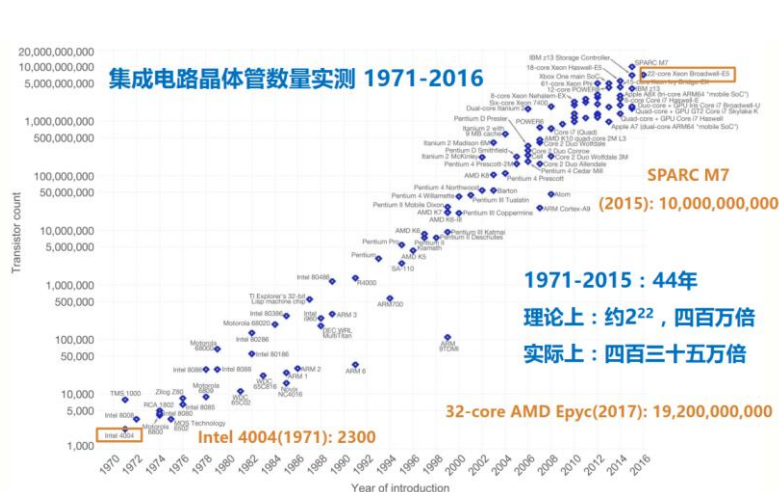
个人计算机



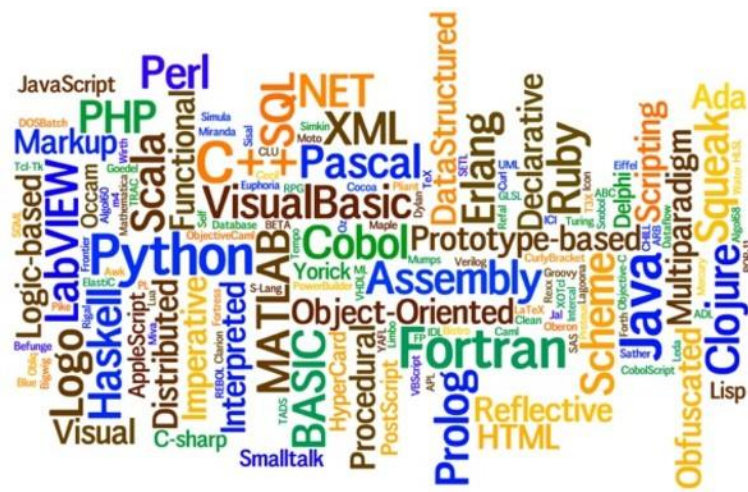
## 1.1 计算机的概念

# 计算机的发展

- 计算机硬件所依赖的集成电路规模参照摩尔定律发展
- 计算机运行速度接近几何级数快速增长
- 计算机高效支撑的各类运算功能不断丰富发展
- 程序设计语言从机器语言、汇编语言到高级语言发展



## 摩尔定律



# 编程语言



# 1.2 程序设计语言



## 程序设计

计算机可编程性的体现

- 亦称编程，深度应用计算机的主要手段
- 已成为当今社会需求量最大的职业技能之一
- 很多岗位将被计算机程序接管，程序设计将是生存技能

## 为什么学习计算机编程？

训练思维（计算思维）、增进认识、带来乐趣、提高效率、利于就业、。。。。



# 1.2 程序设计语言



## ❖ 程序设计语言

- ❖ 一种用于交互(交流)的人造语言
- ❖ 亦称编程语言，程序设计的具体实现方式
- ❖ 主要用于人类和计算机之间的交互

## ❖ 编程语言分类

❖ 机器语言	11010010 00111011	二进制代码
❖ 汇编语言	add 2, 3, result	助记符
❖ 高级语言	result = 2 + 3	接近自然语言

相比人类语言，程序设计语言的结构在语法上十分精密，在语义上定义准确。



## 编程语言种类

- 编程语言有超过600种，绝大部分都不再被使用

- C语言（1972年）是第一个被广泛使用的编程语言

- Python语言（1990年）是最流行最好用的编程语言

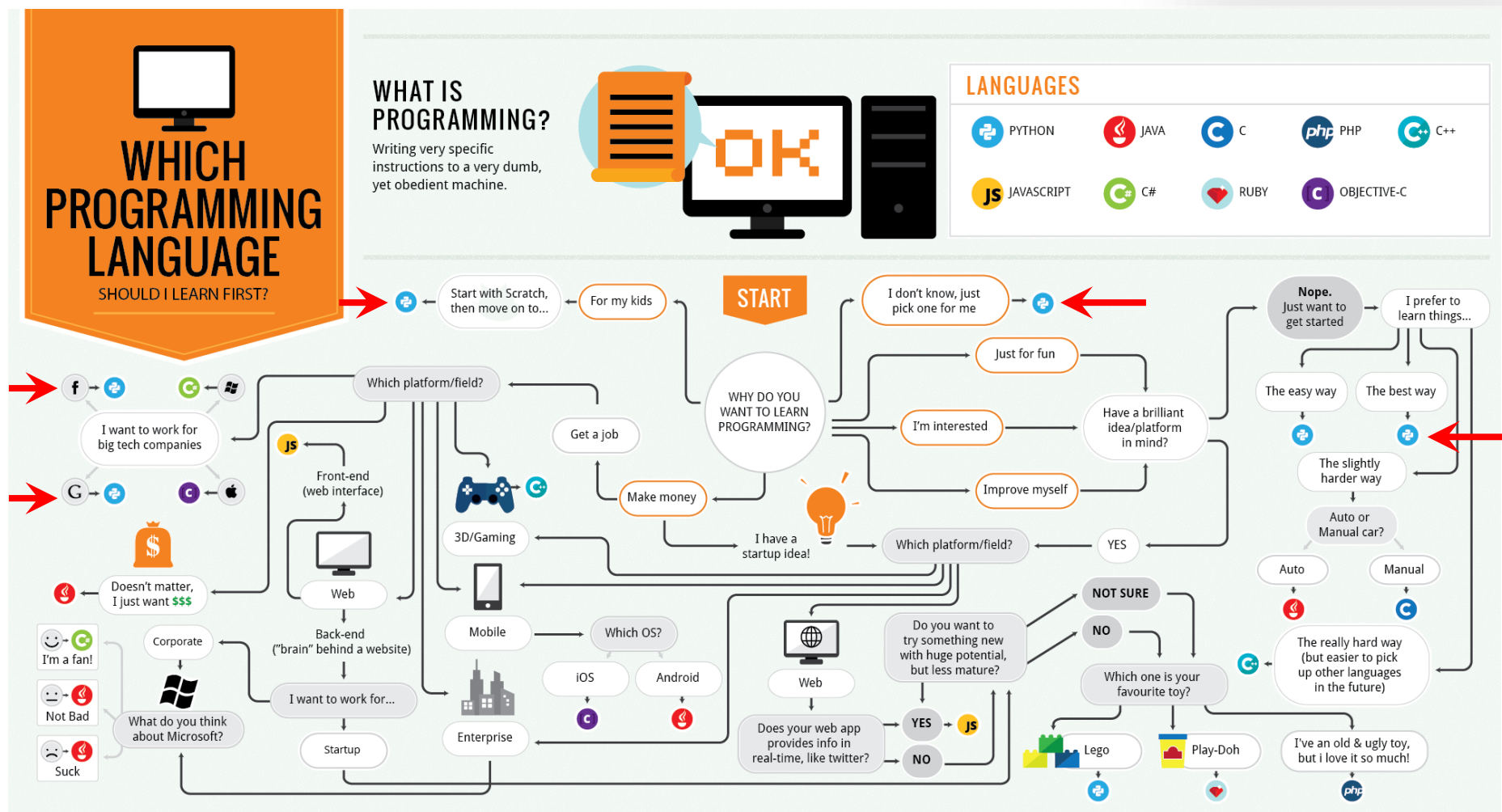


- 通用编程语言，如Python





# 1.2 程序设计语言



编程语言学习选择宝典



# 1.2 程序设计语言



## ☞ 编程语言的执行方式

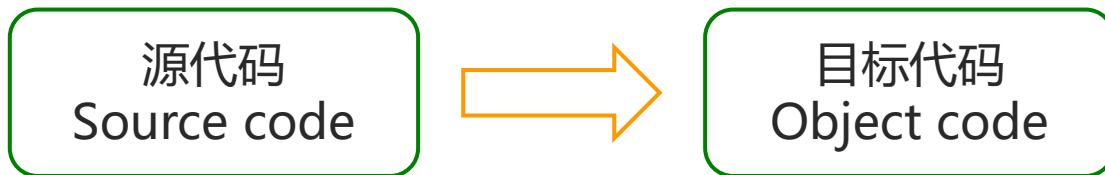
### 编译 or 解释

☞ 源代码 采用某种编程语言编写的计算机程序，人类可读

☞ 例如： `result = 2 + 3` (高级语言代码)

☞ 目标代码 计算机可直接执行，人类不可读 (专家除外)

☞ 例如： `11010010 00111011` (机器语言代码)





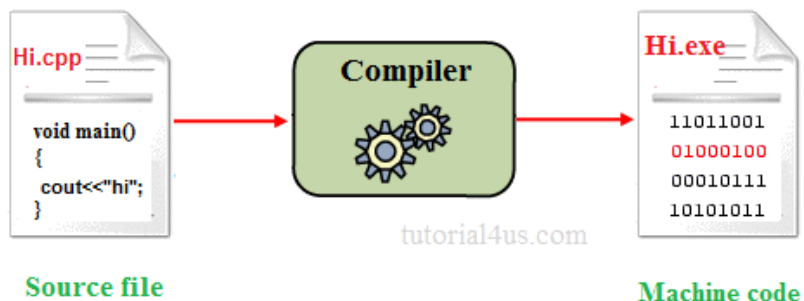
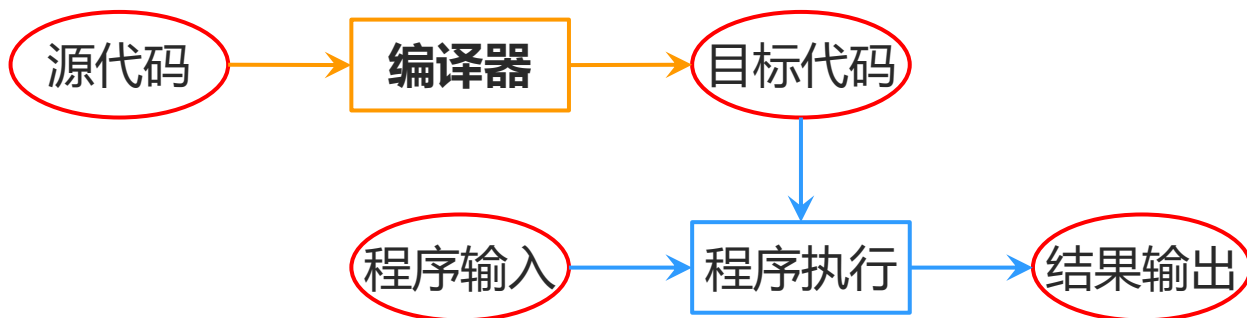


# 1.2 程序设计语言



## 编译 Compiler

将源代码一次性转换成目标代码的过程



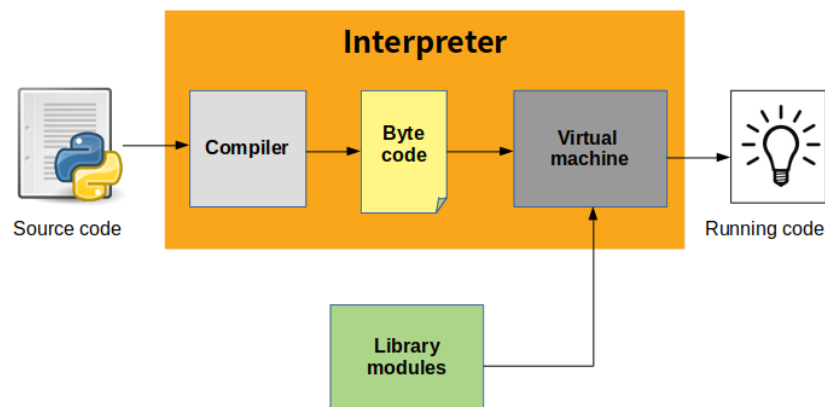


# 1.2 程序设计语言



## 解释 Interpreter

 将源代码逐条转换成目标代码同时逐条运行的过程



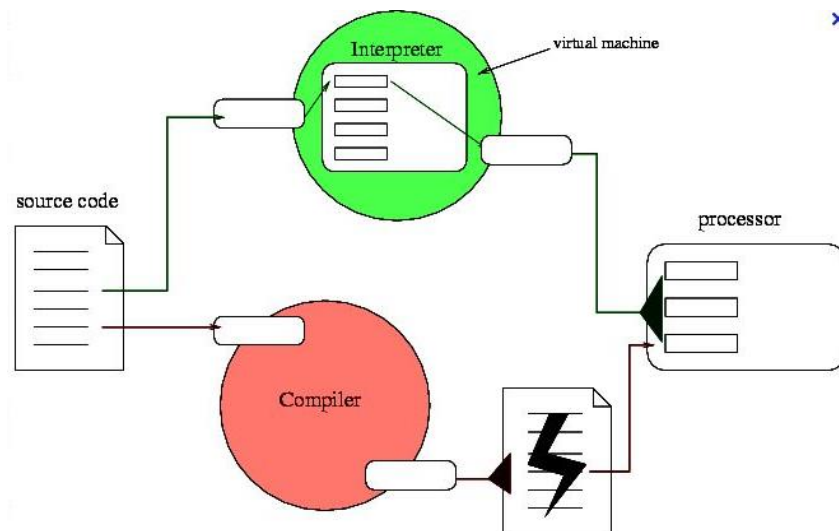


# 1.2 程序设计语言



## 编译 vs 解释

- 编译 一次性翻译，之后不再需要源代码（类似英文翻译）
  - 目标代码执行速度快，使用灵活(同类操作系统)
- 解释 每次程序运行时随翻译随执行（类似实时的同声传译）
  - 便于纠错与维护，可移植性好






# 1.2 程序设计语言



## 静态语言

 编译执行，如C/C++语言、Java语言

## 脚本语言

 解释执行，如Python、JavaScript、PHP语言

## Python语言

 一种被广泛使用的高级通用脚本编程语言

 具有将解释器与编译器结合的新解释器



## 1.2 程序设计语言

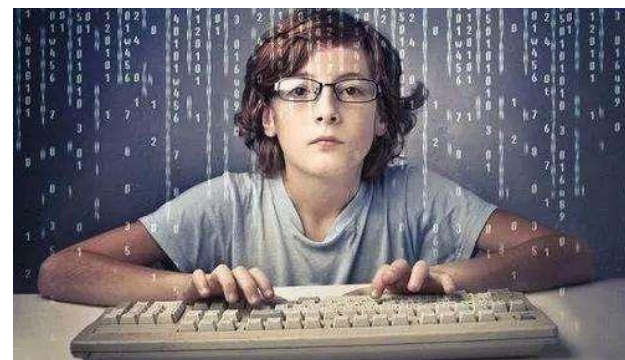


Q: 编程很难学吗? A: 掌握方法就很容易!

### 🔗 编程语言学习模式

- 🔗 首先, 掌握编程语言的语法, 熟悉基本概念和逻辑
- 🔗 其次, 结合计算问题思考程序结构, 会使用编程套路
- 🔗 最后, 参照案例多练习多实践, 学会举一反三

学习计算机编程的重点在于**练习**。





## 1.3 Python语言概述



- Python [ˈpaɪθən], 译为“蟒蛇”
  - 由**编程牛人**领导设计并开发的编程语言
  - 有**开放、开源**精神的编程语言
  - 应用于火星探测、搜索引擎、引力波分析等**众多领域**
- Python语言拥有者
  - Python Software Foundation(PSF)
  - PSF是非盈利组织，致力于保护Python语言开放、开源和发展

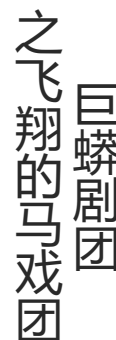




- 1990年，吉多·范罗苏姆 (Guido van Rossum) 设计开发
- 2000年，Python 2.x
- 2008年，Python 3.x



## “终身仁慈独裁者”



## Python语言命名来源

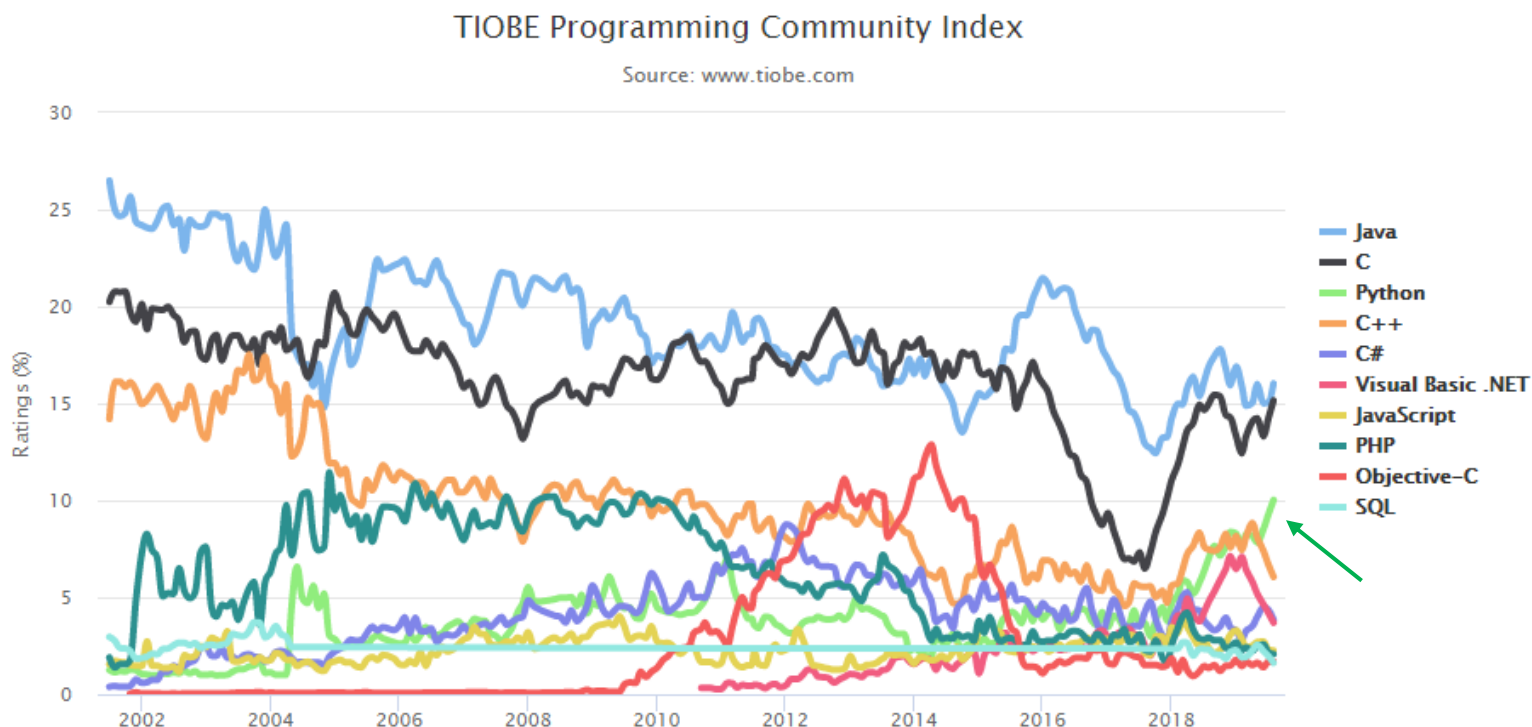




# 1.3 Python语言概述



## Python语言排行 (TIOBE, 10.02%)



Aug 2019

<https://www.tiobe.com/tiobe-index/>



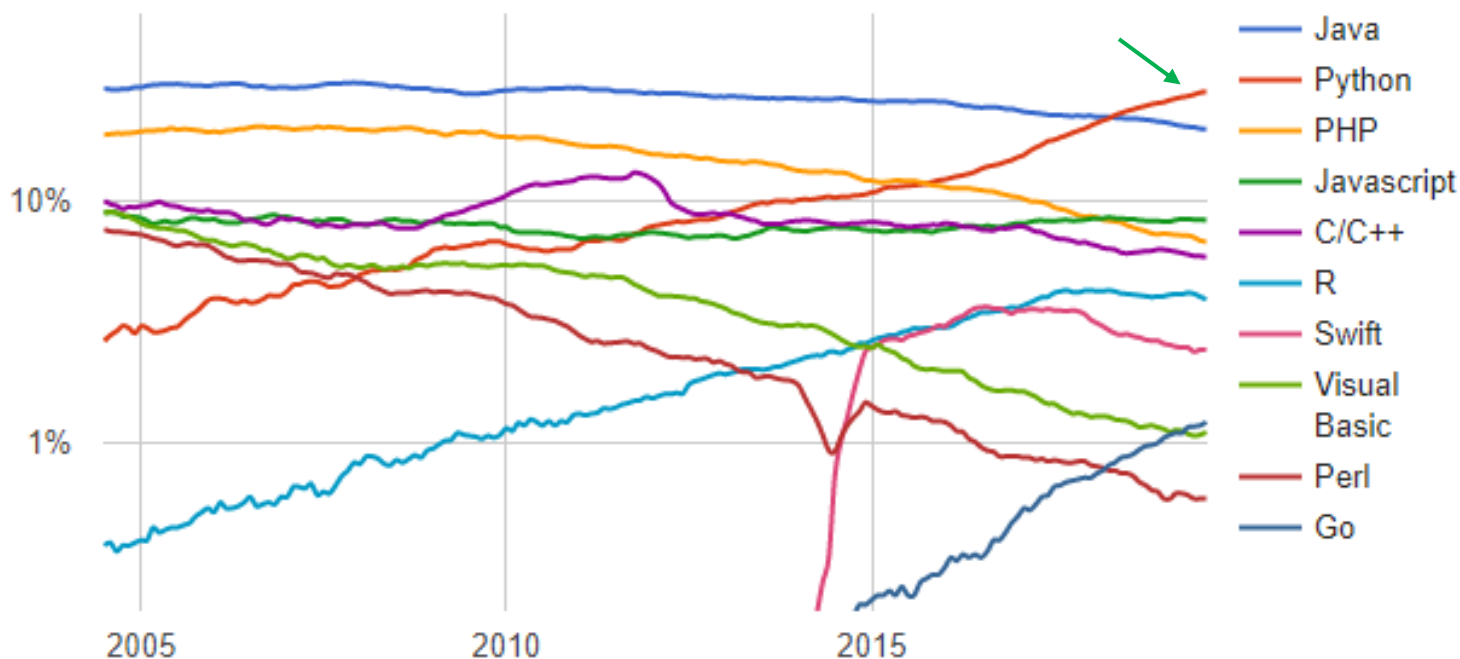


# 1.3 Python语言概述



Python语言排行 (PYPL, 28.73%)

PYPL Popularity of Programming Language



Aug 2019

<https://www.tiobe.com/tiobe-index/>



# PYTHON

难度☆☆☆☆☆

欢迎度★★★★★

始于1991



# JAVA

难度★★★★☆

欢迎度★★★★★

始于1995



# C

难度★★★☆☆

欢迎度★★★★★

始于1972



# PHP

难度★★★★☆

欢迎度★★★★★

始于1990

Python是服务端解释器的、开源的、非编译语言，它用于自身或其他框架，如Django。

python 用来做什么

网页开发

视频游戏开发

桌面GUI

软件开发

Python的优点

易于学习：  
Python广泛认为最易学的编程语言。

库：  
大量令人惊讶的库和函数使得制作东西极为容易。

物联网：  
Python也许会成为物联网最受欢迎语言，如树莓派这样的新平台基于Python。

Python的缺点

速度：  
作为解释型语言，比编译型语言慢得多。

移动端：  
Python在移动计算方面很弱，很少有智能机应用由Python开发。

设计：  
Python是动态型，它需要更多测试以及错误仅在运行时展示。

Python工作市场

平均薪水\$107,000

工作数26,856

顶尖工作位置  
New York City  
Mountain View  
San Francisco

顶级雇主 intel amazon DELL

Python编写了...

Instagram

YouTube

Spotify

Python趣事

Python以电视节目Monty Python's Flying Circus命名。

Python由程序员Guido van Rossum创造。

Python是谷歌的官方语言。

Java是一种开发者用来创造计算机应用程序的语言。你下载程序需要Java运行环境并确保它安装在你的系统上。Java也有一些Web插件允许你在浏览器中运行应用。

用来做什么

安卓和iOS应用开发

视频游戏开发

桌面GUI

软件开发

Java的优点

Java开发者被大量需求：  
2015年Dice.com 报告表明Java开发是需求量最高的软件开发技能。

一种进化的语言：  
这个平台在继续增加新功能，如lambda功能。

安卓应用开发：  
谷歌的安卓移动平台是世界第一的移动平台。编写安卓应用，开发者主要使用Java。

Java的缺点

使用大量内存：  
Java比C++使用更多内存，更大内存占用。

学习曲线：  
不是学起来最简单的语言，但也不是最难的。

缓慢的启动时间：  
每一个试着用Java写安卓应用的人都知道同样的代码在模拟器中启动多么缓慢。

Java工作市场

平均薪水\$102,000

工作数72,226

顶尖工作位置  
New York City  
San Jose  
Washington D.C

顶级雇主 ebay amazon IBM

Java编写了...

MINECRAFT

Java是安卓操作系统的基础

Java趣事

Java的曾用名是Oak，因为在James Gosling的窗外有一棵巨大的橡树（Oak tree）。

你可以用Java玩在线游戏。

你能用Java做什么

在线虚拟旅行

大部分网站允许你通过Java上传图片到他们的平台

使用Java插件你可以使用在线地图

你可以用Java玩在线游戏

C是多用途的、重要的计算机程序语言，支持结构化编程。词法变量范围和递归。是开发固件或移动应用的理想语言。通常被用于编写系统软件。

C 用来做什么

操作系统开发

软件开发

硬件

C的优点

可移动性：  
C是高度可移动的。你在不做改动或仅做很小改动的情况下可以开发运行在不同平台的软件。

C很小：  
C完全基于变量、宏命令、函数和架构，他们不是特别多。因此，C可以嵌入几乎所有现代微型处理器，从冰箱到闹钟。

学会C学会一切：  
因为几乎所有编程语言都由C实现。学会就能使你方便的学会一切编程语言。

C的缺点

运行时间：  
C语言没有运行时检查机制。

面向对象编程：  
不支持面向对象编程，这就是为什么创造C++。

相当大的学习曲线：  
不是最好学的编程语言也不是最难的。适用于有其他编程基础的人。

C工作市场

平均薪水\$102,000

工作数38,456

顶尖工作位置  
New York City  
Washington D.C  
San Francisco

顶级雇主 intel amazon DELL

C编写了...

amazon kindle

Linux

OpenGL

C趣事

C在1969年到1973年间在AT贝尔实验室由Dennis Ritchie造并用于Unix操作系统的再开发。

很多后期语言是从C中直接或者间接的借出来的，其中包括Java、JavaScript、C++、Python、PHP和Perl。

PHP是一种服务端脚本语言，专为网页开发而设计，但也是一门用途广泛的语言。

php 用来做什么

WordPress 插件

Web 开发

创建数据库网页

PHP的优点

容易上手  
有许多文档和教程，一般认为有太多的新手（因为php容易学），二是因为一些开源项目已经输出超过十年，有点过时了。

社区强大  
有非常活跃社区，被广泛使用，开发者遇到的大多数问题都已经有人解决方了。

数据库  
对数据库友好，比如文件系统、图片系统等。

PHP的缺点

太容易学了  
因为太多新手（因为php容易学），二是因为一些开源项目已经输出超过十年，有点过时了。

速度  
经过验证，PHP通常比它的竞争对手运行得慢。

错误处理  
和其他语言相比，PHP的错误处理考虑的不够周全，主要是因为PHP的开发历史比较久远，需要支持老版本。

PHP工作市场

平均薪水\$89,000

工作数32,589

顶尖工作位置  
New York City  
Seattle  
San Francisco

顶级雇主 IBM facebook

PHP编写了...

facebook

WordPress

PHP趣事

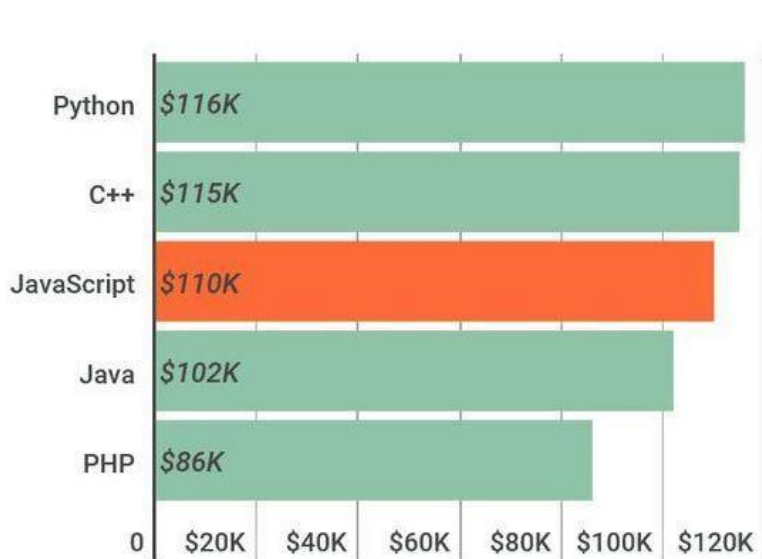
PHP起源于Personal Home Page，现在则是PHP，他们的名字。

Rasmus Lerdorf在1994年PHP，他开发了一系列CGI脚本的个人主页。

PHP能做什么？

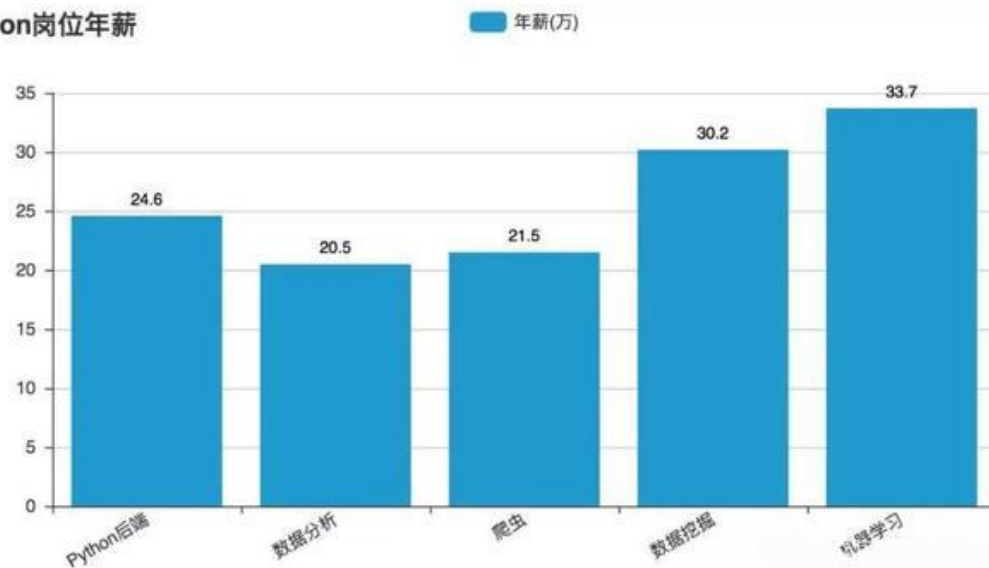
PHP能支持如下：  
PHP Code Text HTML CSS JavaScript

PHP还能  
连接数据库 创建、打开、读写数据库文件 形成动态页面 页面访问 创建Cookie



Python 116K 超过 C++、JS,  
薪酬排行第一  
(来自最大招聘网站Indeed.com数据)

Python岗位年薪



2019年北上广深杭Python就业薪资



# 1.3 Python语言概述



## Python语言的特点

- 语法简洁、跨平台、粘性扩展、开源理念、通用灵活、强制可读、支持中文、模式多样、类库丰富

## Python之禅 (The Zen of Python)

- 优美胜于丑陋，明了胜于晦涩
- 简洁胜于复杂，复杂胜于凌乱
- 扁平胜于嵌套，间隔胜于紧凑
- 。 。 。

Python设计哲学：优雅、明确、简单。

Python开发哲学：用一种方法，最好是只有一种方法来做一件事。





# 1.3 Python语言概述



## Python代码示例

```
# 圆的半径是25
radius = 25
#  $\pi$ 值
pi = 3.1415
# 圆的面积公式
area = pi * radius * radius
# 输出面积
print(area)
# 只输出两位小数
print("%.2f" % area)
print("{:.2f}".format(area))
```

### “Hello, World”

- C

```
#include <stdio.h>

int main(int argc, char ** argv)
{
    printf("Hello, World!\n");
}
```
- Java

```
public class Hello
{
    public static void main(String argv[])
    {
        System.out.println("Hello, World!");
    }
}
```
- now in Python

```
print "Hello, World!"
```

Monday, June 14, 2010

2

简洁性：实现同样功能，Python代码行数仅相当于C语言的1/10~1/5。



更少的程序错误、更快的开发速度、更好的可读性




# 1.3 Python语言概述



## Python语言重要性

### 广泛的应用领域

-  科学计算、数据处理、图形处理、文本处理

-  人工智能、Web开发、游戏开发、网络爬虫

-  办公自动化、软件开发、系统运维、云计算

### 教育语言

-  便于入门，普及编程和人工智能教育

-  培养计算思维，为其他语言学习奠定基础



# 1.4 Python语言开发环境配置



## Python解释器安装

### Python语言主网站



 <https://www.python.org/downloads/>

### Anaconda/Miniconda



 开源的Python包管理器

 <https://www.anaconda.com/distribution/#download-section>

### Download the latest version for Windows

Download Python 3.7.4

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

### Python 3.7 version

Download

64-Bit Graphical Installer (486 MB)  
32-Bit Graphical Installer (418 MB)



# 1.4 Python语言开发环境配置



## Python解释器安装 (Windows10为例)

### 64位安装程序下载链接:

<https://www.python.org/ftp/python/3.7.4/python-3.7.4-amd64.exe>

### 32位安装程序下载链接:

<https://www.python.org/ftp/python/3.7.4/python-3.7.4.exe>





# 1.4 Python语言开发环境配置



## Python解释器安装 (Windows10为例)



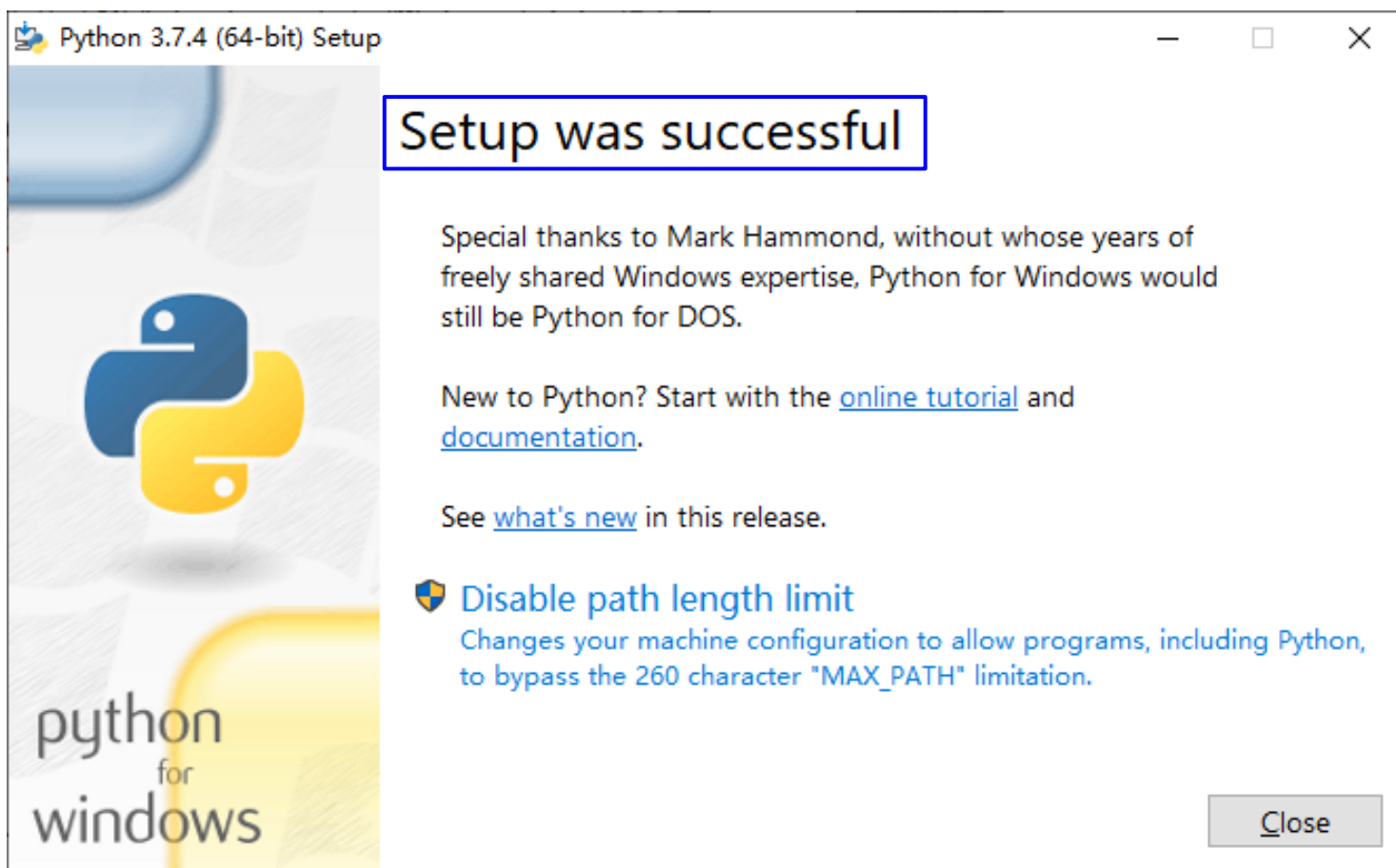
Python解释器安装程序引导过程的启动界面



# 1.4 Python语言开发环境配置



## Python解释器安装 (Windows10为例)



Python解释器安装程序引导过程的成功页面





# 1.4 Python语言开发环境配置





## 运行Hello程序

### 交互式

-  Python解释器即时响应每条代码，并给出输出结果
-  一般用于调试少量代码

### 文件式

-  也称批量式，将Python程序写在文件中，启动Python解释器批量执行文件中的代码
-  最常用的Python编程方式



# 1.4 Python语言开发环境配置



## 交互式 方法1

- 启动Windows命令行工具 (cmd.exe)
- 在 ">>>" 后输入代码: `print("Hello World")`
- 按Enter键后输出结果: **Hello World**
- 输入`exit()`或者`quit()`退出Python运行环境

```
C:\Windows\system32\cmd.exe - python
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\iSynBio>python
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello World")
Hello World
>>>
```

运行环境提示符

Windows命令行下的Python交互运行模式



# 1.4 Python语言开发环境配置



## 交互式 方法2

- 启动Python IDLE (idle.pyw)
- 在 ">>>" 后输入代码: `print("Hello World")`
- 按Enter键后输出结果: **Hello World**
- 输入`exit()`或者`quit()`退出Python运行环境

```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello World")
Hello World
>>> |
```

语法高亮

Ln: 5 Col: 4

Python IDLE下的Python交互运行模式



# 1.4 Python语言开发环境配置



## 🔗 文件式 方法1

- 🔗 启动Windows命令行工具 (cmd.exe) , 并切换到程序文件所在目录
- 🔗 输入: `python hello.py`
- 🔗 按Enter键后输出结果: **Hello World**

```
C:\Windows\system32\cmd.exe

F:\xxmu\教学\选修课申请\Python程序语言入门与应用\2019课件\材料>python hello.py
Hello World

F:\xxmu\教学\选修课申请\Python程序语言入门与应用\2019课件\材料>
```

Windows命令行下运行Python程序文件

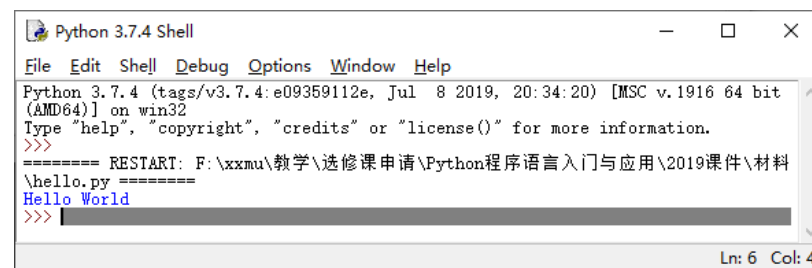
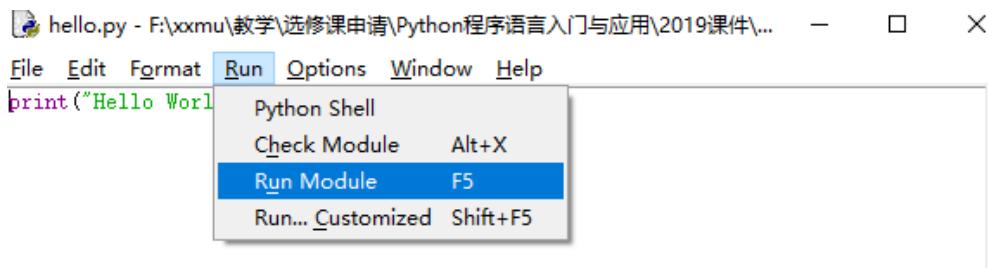


# 1.4 Python语言开发环境配置



## 文件式 方法2

- Python IDLE (idle.pyw)
- 通过File菜单打开文件 `hello.py`
- 在菜单中选择Run→Run Module选项运行程序文件，输出结果：`Hello World`



Python IDLE下运行Python程序文件



# 1.4 Python语言开发环境配置



## Python IDE(集成开发环境)



Vim



Emacs



Visual  
Studio



Visual  
Studio Code



Sublime  
Text



Notepad++



Atom



Pycharm



Eclipse with PyDev



THE INTELLIGENT DEVELOPMENT ENVIRONMENT FOR PYTHON

WingIDE



Komodo  
Edit







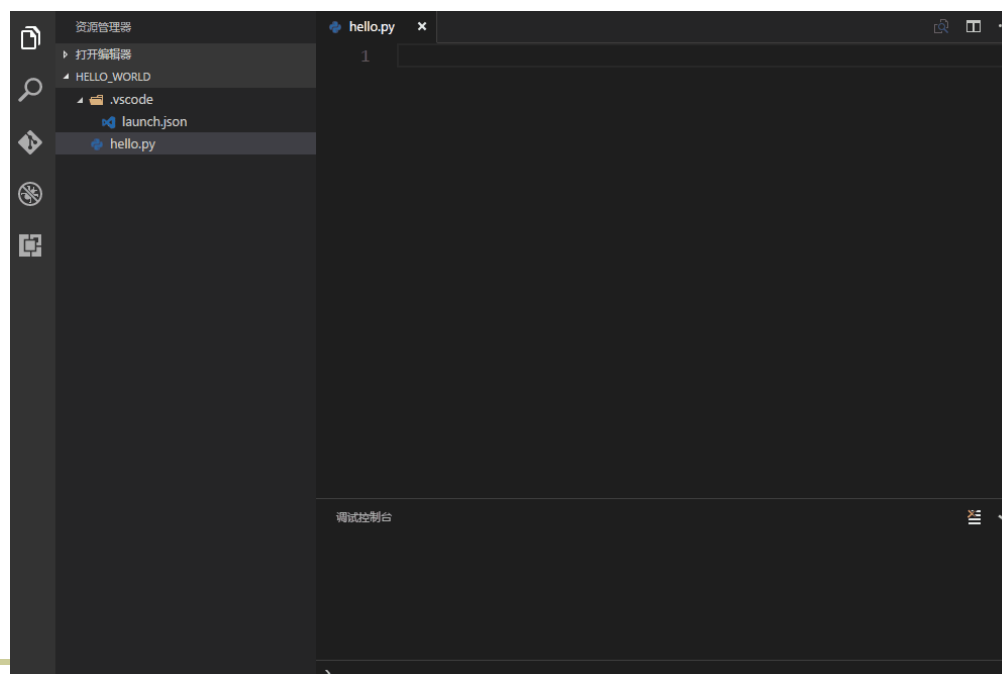
# 1.4 Python语言开发环境配置



## Visual Studio Code

-  <https://code.visualstudio.com/>
-  启动快速，插件丰富，即插即用，集成了现代编辑器所具备的各种特性，包括语法高亮，可定制的热键绑定，括号匹配以及代码片段收集。

Download for Windows			
Stable Build			
		Stable	Insiders
macOS	Package	↓	↓
Windows x64	User Installer	↓	↓
Linux x64	.deb	↓	↓
	.rpm	↓	↓
<a href="#">Other downloads</a>			





# 1.4 Python语言开发环境配置



## Python小程序运行实例1

### 圆面积的计算

```
>>> radius = 25          # 圆的半径是25
>>> pi = 3.1415          #  $\pi$  值
>>> area = pi * radius * radius  # 圆的面积公式
>>> print(area)
1963.43750000000002
>>> print("%.2f" % area)    # 只输出两位小数
1963.44
>>> print("{:.2f}".format(area))
1963.44
```

IDLE交互环境

```
radius = 25          # 圆的半径是25
pi = 3.1415          #  $\pi$  值
area = pi * radius * radius  # 圆的面积公式
print(area)
print("%.2f" % area)    # 只输出两位小数
print("{:.2f}".format(area))
```

文件式内容

1.1CircleArea.py



# 1.4 Python语言开发环境配置



## Python小程序运行实例2

### 简单人名对话

```
>>> name = input("输入姓名: ")
输入姓名: 郭靖
>>> print("{}同学, 学好Python, 前途无量!".format(name))
郭靖同学, 学好Python, 前途无量!
>>> print("{}大侠, 学好Python, 大展拳脚!".format(name[0]))
郭大侠, 学好Python, 大展拳脚!
>>> print("{}哥哥, 学好Python, 人见人爱!".format(name[1:]))
靖哥哥, 学好Python, 人见人爱!
```

IDLE交互环境

```
name = input("输入姓名: ")
print("{}同学, 学好Python, 前途无量!".format(name))
print("{}大侠, 学好Python, 大展拳脚!".format(name[0]))
print("{}哥哥, 学好Python, 人见人爱!".format(name[1:]))
```

文件式内容

1.2NameDialogue.py



# 1.4 Python语言开发环境配置



## Python小程序运行实例3

### 斐波那契数列计算

```
>>> a, b = 0, 1

>>> a, b = 0, 1
>>> while a < 1000:
    print(a, end=',')
    a, b = b, a + b

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987,
```

IDLE交互环境

增加缩进表示语句块的开始  
减少缩进表示语句块的退出

```
a, b = 0, 1
while a < 1000:
    print(a, end=',')
    a, b = b, a + b
```

文件式内容

1.3FibonacciSequence.py



# 1.4 Python语言开发环境配置

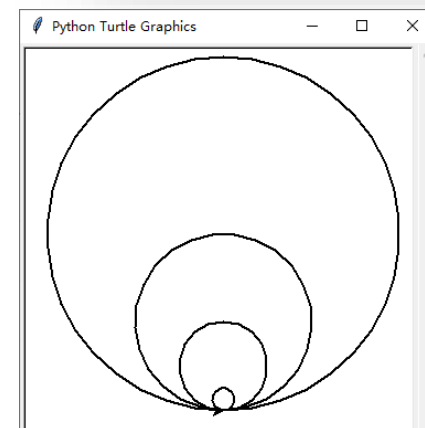


## Python小程序运行实例4

### 同切圆绘制

```
>>> import turtle          # 引用turtle库
>>> turtle.pensize(2)      # 设置画笔宽度为2像素
>>> turtle.circle(10)      # 绘制半径为10像素的圆
>>> turtle.circle(40)      # 绘制半径为40像素的圆
>>> turtle.circle(80)      # 绘制半径为80像素的圆
>>> turtle.circle(160)     # 绘制半径为160像素的圆
```

```
import turtle              # 引用turtle库
turtle.pensize(2)          # 设置画笔宽度为2像素
turtle.circle(10)          # 绘制半径为10像素的圆
turtle.circle(40)          # 绘制半径为40像素的圆
turtle.circle(80)          # 绘制半径为80像素的圆
turtle.circle(160)         # 绘制半径为160像素的圆
```



IDLE交互环境

文件式内容

1.4CircleDraw.py



# 1.4 Python语言开发环境配置



## Python小程序运行实例5

### 日期和时间输出

```
>>> from datetime import datetime # 引用datetime库
>>> now = datetime.now() # 获得当前的日期和时间信息
>>> print(now)
2019-08-16 17:24:18.376137
>>> now.strftime("%x") # 输出其中的日期部分
'08/16/19'
>>> now.strftime("%X") # 输出其中的时间部分
'17:24:18'
```

IDLE交互环境

```
from datetime import datetime # 引用datetime库
now = datetime.now() # 获得当前的日期和时间信息
print(now)
now.strftime("%x") # 输出其中的日期部分
now.strftime("%X") # 输出其中的时间部分
```

文件式内容

1.5Datetime.py








# 1.5 程序的基本编写方法



## IPO

-  I: Input 输入，程序的输入
-  P: Process 处理，程序的主要逻辑
-  O: Output 输出，程序的输出



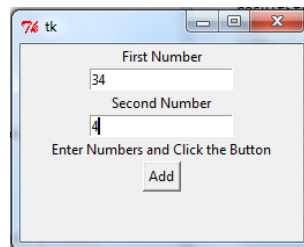
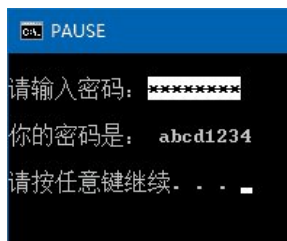


# 1.5 程序的基本编写方法



## Input 输入

- 程序的输入
  - 文件输入、网络输入、控制台输入、交互界面输入、内部参数输入等
- 输入是一个程序的开始



```
python argparse_test.py -t 1  
python argparse_test.py -t=1  
python argparse_test.py --thread 1  
python argparse_test.py --thread=1
```





# 1.5 程序的基本编写方法



## Process 处理

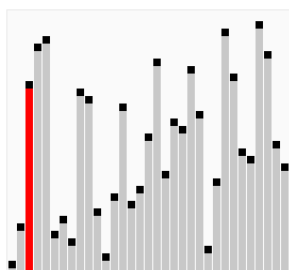
### 处理

程序对输入数据进行计算产生输出结果的过程

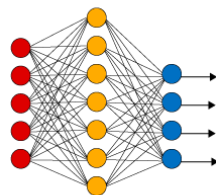
### 处理方法

统称为算法，它是程序最重要的部分

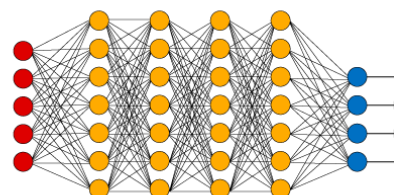
算法是一个程序的灵魂



Simple Neural Network



Deep Learning Neural Network



● Input Layer ● Hidden Layer ● Output Layer





# 1.5 程序的基本编写方法

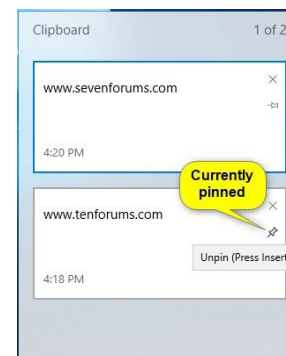
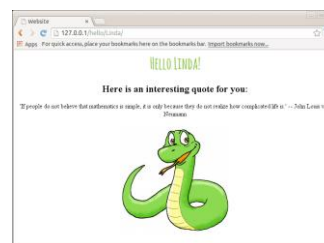
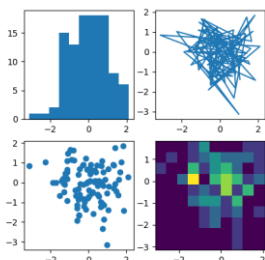
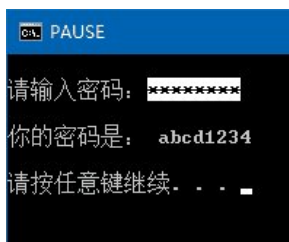


## Output 输出

### 程序的输出

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

输出是程序展示运算结果的方式





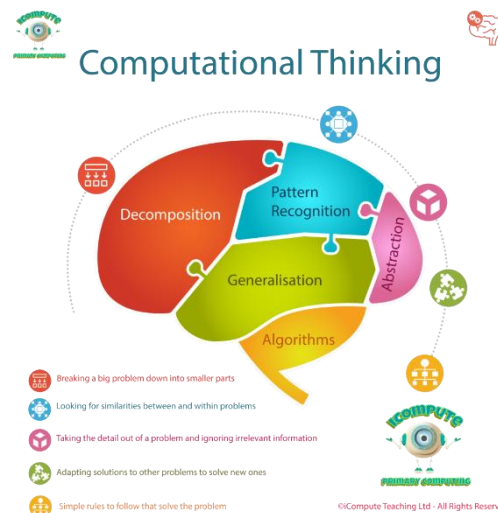
# 1.5 程序的基本编写方法



## ❧ 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





# 1.5 程序的基本编写方法



## 编程解决问题的步骤

- 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**
- 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**

**3个精简步骤：**确定IPO→编写程序→调试程序





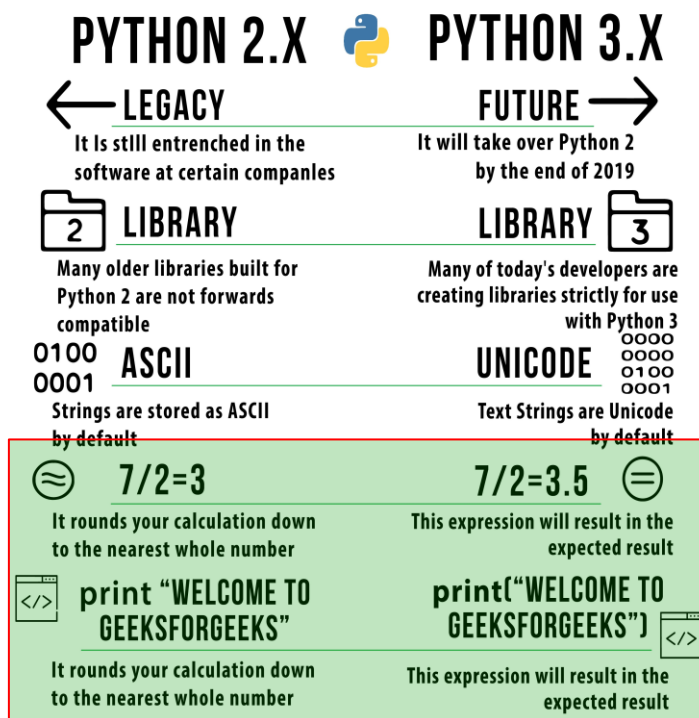
# 1.6 Python语言的版本更迭

## Python 2.x与Python 3.x

2010年, Python 2.x最后一个版本2.7

2008年, Python 3.x第一个主版本3.0

## 版本差异



Python 2.7  
将于2020年1月1  
日终止支持。

Python 3.x  
不完全向下兼容  
Python 2.x程序!



# 1.6 Python语言的版本更迭



## 🐍 版本选择建议

- 🐍 若无特殊情况，**请学习Python 3.x版本**
- 🐍 特殊需要可以使用Anaconda虚拟环境管理功能搭建Python 2.x开发环境





# 本章要点



- ✚ 程序设计语言的执行方式包括编译和解释两种。
- ✚ Python语言是一个语法简洁、跨平台、可扩展的开源通用脚本语言。
- ✚ IDLE是轻量级Python语言开发环境，支持交互式 and 批量式两种编程方式。
- ✚ 程序的基本编写方法为IPO方法。
- ✚ Python 3.x版本是Python语言的现在和未来。



# 程序练习题



## 1.1 字符串拼接

如：[某人]想去[某地]看看。

接受用户输入的2个字符串，将它们组合后输出。

## 1.2 整数序列求和

用户输入一个正整数N，计算从1到N相加之和。

## 1.3 九九乘法表输出

工整打印输出常用的九九乘法表，格式不限。

1×1=1									
1×2=2	2×2=4								
1×3=3	2×3=6	3×3=9							
1×4=4	2×4=8	3×4=12	4×4=16						
1×5=5	2×5=10	3×5=15	4×5=20	5×5=25					
1×6=6	2×6=12	3×6=18	4×6=24	5×6=30	6×6=36				
1×7=7	2×7=14	3×7=21	4×7=28	5×7=35	6×7=42	7×7=49			
1×8=8	2×8=16	3×8=24	4×8=32	5×8=40	6×8=48	7×8=56	8×8=64		
1×9=9	2×9=18	3×9=27	4×9=36	5×9=45	6×9=54	7×9=63	8×9=72	9×9=81	

## 1.4 计算 $1!+2!+3!+\dots+10!$

## 1.5 五角星的绘制

绘制一个红色的五角星。



作业的代码文件打包(.zip)发送到邮箱[python\\_xxmu@163.com](mailto:python_xxmu@163.com)



# 课程材料地址



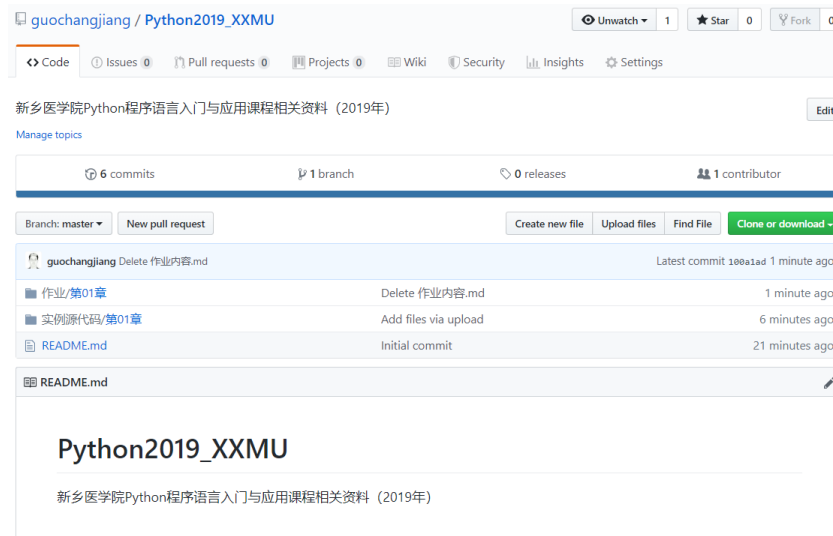
## GitHub



 <https://github.com>

 一个面向开源及私有软件项目的托管平台

 可以十分轻易地找到海量的开源代码



[https://github.com/guochangjiang/Python2019\\_XXMU](https://github.com/guochangjiang/Python2019_XXMU)  
包含实例代码、课程作业、作业代码等课程资料



脚本文件一般用 .py 后缀

单行注释

导入其它代码模块

注意! Python最好也最个性的语法:  
使用缩进来代替其它语句块声明;  
一般建议每个层级用4个空格来缩进.

变量得先实例化  
才可进一步计算

单行的语句块, 其实可以不换行的,  
但是, 建议清晰起见, 规范点:  
- 另起一行  
- 缩进一级

函数声明,  
注意使用冒号结束声明

多行注释的内容不用遵守当前缩进  
只要开始的''' 缩进正确就成!

每级语法块不用}之类的括号引领!  
直接回车+4空格  
(当然, 要在当前缩进基础上)

coffeeghost-q-in-py.py x

中文用户一定得先用这行来声明编码, 同时文件本身也得存储成UTF-8编码!

```
1 # -*- coding: utf-8 -*-
2 # Quick Python Script Explanation for Programmers
3 # 给程序员们的超快速Python脚本解说
4 import os
5
6 def main():
7     print 'Hello World!'
8     print "这是Alice\的问候."
9     print '这是Bob\的问候.'
10
11     foo(5, 10)
12
13     print '=' * 10
14     print '这将直接执行'+os.getcwd()
15
16     counter = 0
17     counter += 1
18
19     food = ['苹果', '杏子', '李子', '梨']
20     for i in food:
21         print '俺就爱整只:' + i
22
23     print '数到10'
24     for i in range(10):
25         print i
26
27
28 def foo(param1, secondParam):
29     res = param1 + secondParam
30     print '%s 加 %s 等于 %s' % (param1, secondParam, res)
31     if res < 50:
32         print '这个'
33     elif (res >= 50) and ((param1 == 42) or (secondParam == 24)):
34         print '那个'
35     else:
36         print '嗯...'
37     return res
38
39 '''这是多
40 行注释.....'''
41
42 if __name__ == '__main__':
43     main()
```

模块名, 其实导入了 os.py

函数名"main"在这儿并不是必须的, 调用在这段脚本的最后部分;

声明单行字符串, 使用双/单引号都成,  
注意对字符串中的引号进行逃逸处理!

函数调用, 声明在后述代码;

字符串可乘, 等于: '====='

调用了os 模块中的函数

连接字符串

内置的列表类型对象, 其实可以包含不同类型数据,  
甚至可以包含其它列表对象;

在循环中, i 指代了列表中按顺序的每个"food"

range()内置函数, 返回类似  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

的数字列表, 注意 for in 循环语句使用冒号结束声明!

字符串的格式化输出基本类似C语言的

判定式也基本和C语言的相同

逻辑运算符, 不使用 && 和 ||, 使用直观的E文单词

这都是合法注释

一般在脚本最后调用主函数 main(); 而且使用 内置的运行脚本名来判定;  
当且仅当我们直接运行当前脚本时, \_\_name\_\_ 才为 \_\_main\_\_  
这样当脚本被当作模块进行 import 导入时, 并不运行 main()  
所以, 一般这里是进行测试代码安置的...

在MS中很好的支持UTF-8的编辑器不多,  
跨平台又支持Py特性的更少;  
好在我们有 Limodou 贡献的 UliPad  
这一编辑器本身就是Py实现的!  
(基于wxPython)

关于

UliPad 4.0

作者: Limodou (limodou@gmail.com)

如果你有任何问题请与我联系。

[The UliPad project homepage](#)

[The UliPad maillist](#)

[The UliPad Snippets Site](#)

[My Blog](#)

[Contact me](#)

确定

用冒号来结束判断句,  
在 if elif else 行最后







# 下周课程



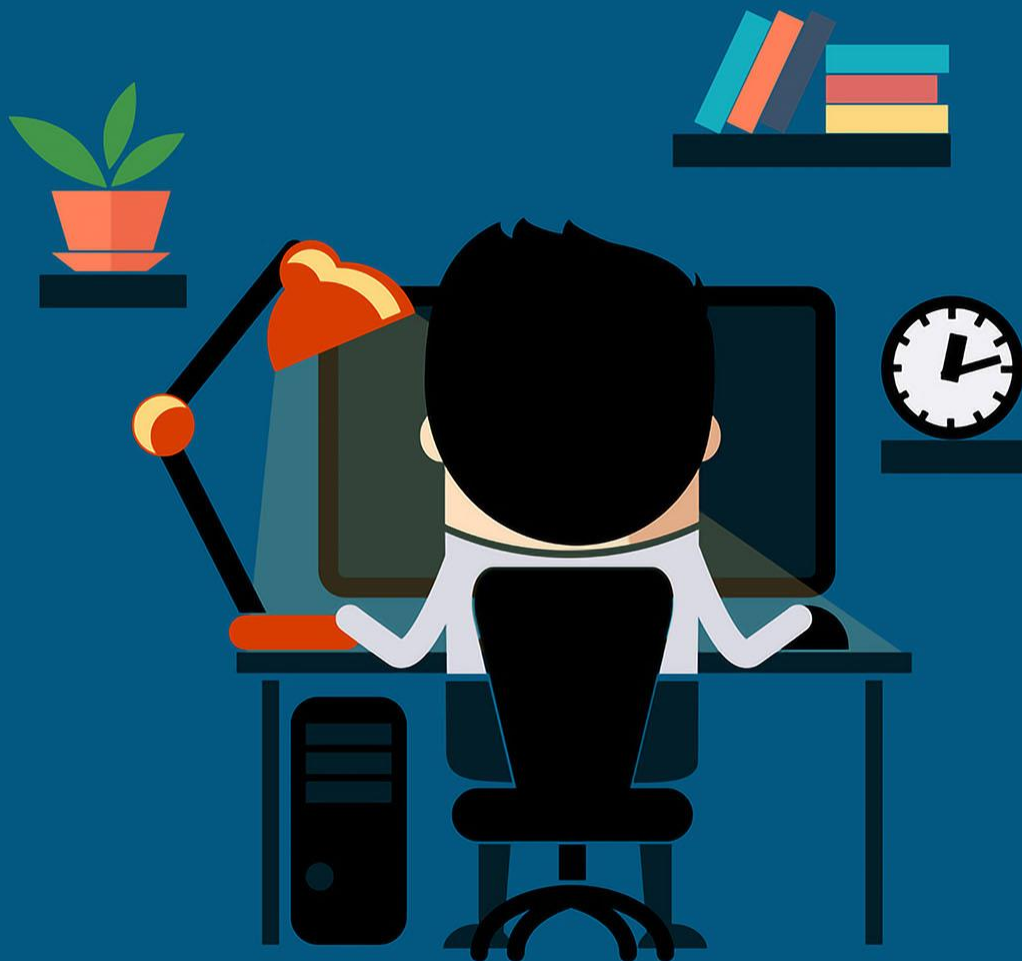
## 第2章 Python程序实例解析

 Python语言基本语法

 实例1：温度转换

 实例2：Python蟒蛇绘制

编程辣么好，还等什么？开始学习吧！



Programing is an Art