# Python开发入门

### **本章学习目标**

* 了解Python的历史与特征
* 了解Python解释器
* 掌握PyCharm的安装

自从20世纪90年代初Python语言诞生至今，Python以其优美、清晰、简单的特性在全世界广泛流行，成为最主流的编程语言之一，它已被逐渐广泛应用于系统管理任务的处理和Web编程。

## Python语言的历史与特征

### **Python语言的起源**

Python的创始人为Guido van Rossum（见图1.1）。1989年圣诞节期间，在阿姆斯特丹，Guido为了打发圣诞节的无趣，决心开发一个新的脚本解释程序，做为ABC 语言的一种继承。之所以选中Python（大蟒蛇的意思）作为该编程语言的名字，是因为他是一个叫Monty Python的喜剧团体的爱好者。



1. Python之父

ABC是由Guido参加设计的一种教学语言。就Guido本人看来，ABC这种语言非常优美和强大，是专门为非专业程序员设计的。但是ABC语言并没有成功，究其原因，Guido认为是其非开标识放造成的。Guido决心在Python中避免这一错误。同时，他还想实现在ABC中闪现过但未曾实现的东西。

就这样，Python在Guido手中诞生了。可以说，Python是从ABC发展起来，主要受到了Modula-3（另一种相当优美且强大的语言，为小型团体所设计的）的影响。并且结合了Unix shell和C的习惯。

### **Python语言的发展**

1991年，第一个Python编译器诞生。它是用C语言实现的，并能够调用C语言的库文件。从一出生，Python已经具有了：类，函数，异常处理，包含表和词典在内的核心数据类型，以及模块为基础的拓展系统。

Python语法很多来自C，但又受到ABC语言的强烈影响。来自ABC语言的一些规定知道今天还富有争议，比如强制缩进。但这些语法规定让Python容易读。另一方面，Python聪明的选择服从一些惯例，比如回归等号赋值。Guido认为，如果“常识”上确立的东西，没有必要过度纠结。

Python从一开始就特别在意可拓展性，Python可以在多个层次上拓展。从高层上，你可以直接引入.py文件。在底层，你可以直接引用C语言的库。Python程序员可以快速的使用Python写.py文件作为拓展模块。但当性能是考虑的重要因素时，Python程序员可以深入底层，写C程序，编译为.so文件引入到Python中使用。Python就好像是使用钢构建房一样，先规定好大的框架。而程序员可以在此框架下相当自由的拓展或更改。

最初的Python完全由Guido本人开发，Python得到Guido同事的欢迎。他们迅速的反馈使用意见，并参与到Python的改进中。Guido和一些同事构成Python的核心团队。他们将自己大部分的业余时间用于hack Python。随后，Python拓展到研究所之外。Python将许多机器层面上的细节隐藏，交给编译器处理，并凸显出逻辑层面的编程思考。Python程序员可以花更多的时间用于思考程序的逻辑，而不是具体的实现细节。这一特征吸引了广大程序员，Python开始流行。

我们不得不暂停我们的Python时间，转而看一看瞬息万变的计算机行业。1990年代初，个人计算机开始进入普通家庭。

### **Python语言标准**

### **Python语言特性**

## Python解释器

### **CPython**

### **IPython**

### **PyPy**

### **Jython**

### **IronPython**