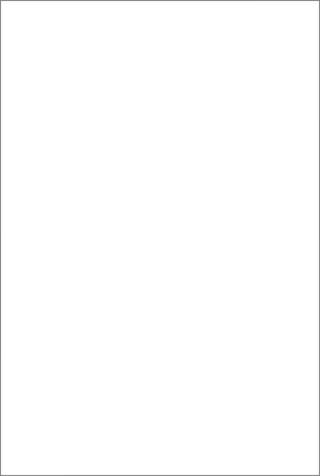
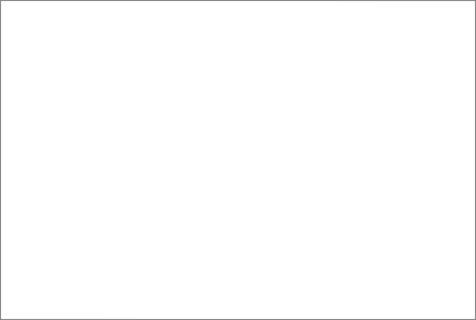
在这一课的内容中，我们将简单介绍Python语言的发展历史，Python语言的应用领域，以及Python2 和Python3的区别等等。从而让大家对Python有一个基本的认识。

**Python**的前世今生

如今，当我们看到Python这个词的时候，首先想到的已经不再是下面这种令人敬畏的生物。



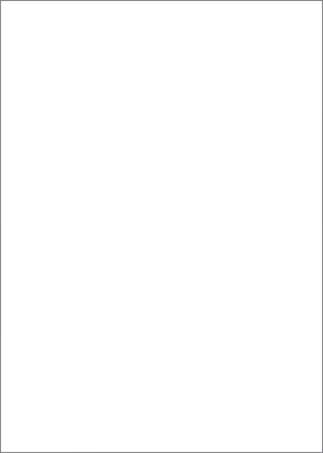
而在三十年前，情况恰恰相反。

Python语言孕育于上个世纪80年代，而其具体的实现则始于1989年的12月。Python语言的作者是Guido van Rossum，就是下面这个典型的IT Geek。

作为Python语言之父，Guido van Rossum于1956年1月31日出生于荷兰，并在1982年获得了阿姆斯特丹大学的数学与计算机科学硕士学位。在Python社区，Van Rossum被称为“终身的仁慈独裁者”（BDFL），也就是说他将持续参与Python的开发过程，并在需要的时候做出决定。从2005年到2012年，Van Rossum被Google雇佣， 在那里他有一半的时间用于开发Python语言。从2013年开始至今，Van Rossum成为Dropbox的雇员。如今，他生活在美国加州的Belmont。

关于Python语言的诞生，Van Rossum在1996年写道，

“大概在1989年12月的时候，我想给自己的圣诞节🎄假期找一个好玩的编程项目来打发时间。而在这段时间里面，办公室关门了，我只有一个家用电脑可以用。于是，我决定为构思已久的新的脚本语言写一个解释器。这个脚本语言将是ABC语言（另一种非系统语言，大家可以google或者维基搜索）的继承者，可以很好的为Unix/C黑客所用。出于某种玩世不恭的心态，我为这个项目取名Python，因为我是Monty Python’s Flying Circus（巨蟒剧团之飞翔的马戏团）的忠实粉丝。



这里科普一下，Monty Python是上世纪六七十年代非常出名的英国电视喜剧片，于1969年10月5日在BBC开播，总共播出了4季共45集，并在随后几十年内继续影响着喜剧的发展，其对于喜剧的影响力不亚于披头士乐队对引用的影响。

看到Python的诞生，我不禁想到，现在的00后10后里面会不会因为看了某部电影或者玩了某款游戏然后发明一门新的编程语言呢？

2000年的10月16日，Python 2.0版本被正式发布，其中包含了多种重要的新特性，包括垃圾回收机制和对Unicode的支持。在2.0版本发布后，Python语言的开发流程变得更加透明，也受到了开发者社区的广泛关注。

Python 3.0版本（最初被称为Python 3000，或者py3k）在2008年的12月3日发布。它是Python语言的一个全新分支，而且不兼容之前的版本。不过尽管如此，Python 3的很多特性也在随后被赋予了之后发布的Python 2.6.x和2.7.x版本。

最初，Python 2.7的“终结日”被设置为2015年，不过考虑到大量已有项目仍在使用Python 2.x版本，这一时间被推迟到了2020年。

2017年1月的时候，Google宣布将开发一款从Python 2.7到Go语言的转译器，这一举动也被视为对Python 2.7大限将至的一种回应。

Python是一种多范式编程语言，这么说有点太抽象了。简单来说，它支持了高级编程语言的多种特性，比如面向对象的编程和结构化编程，还支持函数式编程和面向切面的编程，以及元编程等等。此外，Python通过扩展还可以支持逻辑编程和契约式设计等等高大上的概念。

看了上面这段话，你很可能没有任何感觉。没关系，这只是个开始而已。在随后的学习过程中，我们将会把这些概念通过具体的实例让大家来逐渐熟悉和了解。

而现在，就像《盗梦空间》中的入梦一样，只是在你的脑中洒下一颗种子。

在Zen of Python（Python之道）中，对Python语言的编程哲学做了以下的概括：

1.优雅美丽比丑陋好

2.直白比含蓄好

3.简单比复杂好

4.复杂比难以理解好

5.可读性很重要

虽然Python语言并没有完全按照这种思想来设计，但却具有高度的可扩展性。

**Python**语言的应用领域

如今，Python语言的应用领域已经非常广阔，除了最近所有人都在关注的AI深度学习，Python在诸多领域都有着广泛的应用。

1.网络应用开发

Python可以通过多种方式来进行网络开发：

（1）使用类似[Django](http://www.djangoproject.com/)和[Pyramid](http://www.pylonsproject.org/)之类的框架

（2）使用类似[Flask](http://flask.pocoo.org/)和[Bottle](http://bottlepy.org/)的微框架

（3）使用类似[Plone](http://www.plone.org/)和[django CMS](https://www.django-cms.org/)的高级内容管理系统

Python的标准库还支持多种互联网协议：

（1）[HTML and XML](http://docs.python.org/library/markup)

（2）[JSON](http://docs.python.org/library/json.html)

（3）Email

（4）支持FTP,IMAP和其它的互联网协议

（5）简单易用的[socket interface](http://docs.python.org/howto/sockets.html)接口

而丰富的第三方类库提供了更为强大的功能：

（1）[Requests](https://pypi.python.org/pypi/requests/),一个非常强大的HTTP 客户端库

（2）[BeautifulSoup](http://www.crummy.com/software/BeautifulSoup/)，一种HTML 解析器，可以处理各种HTML内容

（3）[Feedparser](https://pypi.python.org/pypi/feedparser)，可以解析RSS/Atom feeds

（4） [Paramiko](https://pypi.python.org/pypi/paramiko/)，可以实现SSH2协议

（5）[Twisted Python](http://twistedmatrix.com/),一种用于异步网络编程的框架。

2.科学与数据统计分析

这也是目前Python受到广泛关注的重要应用领域。

（1）[SciPy](http://scipy.org/)中包含了数学、科学和工程的多个包。

（2）[Pandas](http://pandas.pydata.org/)用于数据分析和建模

3.桌面GUI开发

Python中主要使用[Tk](http://wiki.python.org/moin/TkInter) GUI库来实现用户界面。

4.软件开发

对于软件开发来说，Python通常作为一种辅助语言，而非主要开发语言。

5.商业应用

Python语言也可以用来开发ERP和电子商务系统。

（1）[Odoo](https://www.odoo.com/)

（2）[Tryton](http://www.tryton.org/)

关于使用Python开发的应用，可以参考这里：<https://wiki.python.org/moin/Applications>

可以看到，在Python的所有应用领域中，网络应用以及科学与数据统计分析是最为主流的应用，当然如今在AI时代，Python更是开始大量用于深度学习开发，和C++语言一起成为AI研究与应用开发的两大主流编程语言。

**Python 2 vs Python 3**

大家现在也知道了，Python有两个主要的分支版本，分别是Python 2（如今的版本号是2.7.14)和Python 3(最新的版本号是3.6.3）。

使用Python 2的好处在于，有相当多的第三方库目前只支持Python 2，而且很多已开发的项目都是基于Python 2的。

然而现在离2020年只剩下两年多的时间了，继续使用Python 2似乎有一定的风险。



而使用Python 3的好处和坏处跟Python 2恰恰相反。

作为未来的主流分支，Python 3无疑将获得更好的支持，但目前仍有部分第三方库不支持Python 3。

那么，何去何从呢？

如果是3年前问这个问题，那么还要纠结一番。不过在2017年底再问这个问题，答案是明显的。

1.大多数的软件包都已经支持Python 3

2.Django推荐使用Python 3,而且大量的软件都已经停止支持Python 2

3.Python 2.7将在2020年终结，如今已经只有两年多了

4.Python 3正在不断发展完善，并加入更多诱人的特性。

因此，对于新手开发者来说，个人建议就不要有任何的纠结，直接上Python 3就是。

好了，本课的内容到此为止。

在下一课的内容中，我们将学习如何配置和安装Python3及其运行环境，以及我们在本系列教程中所需要的第三方工具