

01：导语

01：导语

语言起源

python的语法特点

为什么喜欢python

python 能做什么

怎么对待python

课程内容

pythonic

高性能与优化

最后

python的未来前景

语言起源

python 最近这几年非常流行的语言,最近流行是没问题的,但不是新兴的语言,可以追溯到上世纪90年代初...

随着大数据与人工智能的流行, python逐步的走向我们的视线,真正意义上的变成主流语言

国内使用python的产品, 如豆瓣, 知乎

我们生活的时代, 编程语言是非常多的, 每一种语言都有它技术的特点以及适用的场景, 而python就像空气一样无孔不入, 可以帮助大家解决大大小小的各种问题。

那么python现在流行到什么程度呢/根据TIOBE语言热度排行榜,python高居第4位.已经超越了C# .net仅次于JAVA C C++,有些同学说目前才第4位,还没到第1位的程度

但大家要知道的是,

- 第一:python的上升速度最快的/
- 第二:在这个世界上是有成千上百种语言的能到第4位已经是非常不简单的.

python的语法特点

- 是一门编程语言是从多编程语言当中的一种
- 语法简洁优雅 编写的程序容易阅读

- 跨平台，可以在windows linux macos
- 易于学习，站在非计算专业的角度来讲，如果把编程语言当做解决问题的工具来说，python确实相对于C++，JAVA 等语言易于学习和掌握
- 丰富的标准库与第三方库，比如电子邮件等
- python是面向对象的语言

缺点

任何一种语言都不是完美的 有优点自然会有缺点,那么我们这一节课

慢 是python是它的缺点

那么我们说一种语言快或者是慢,肯定是有会有一个比较对象的,在这里我们说python慢 是相对哪种语言来说的呢,主要呢 是相较于 C C++ JAVA语言来说

这里简单的介绍一个,

编译型语言

C, C++ 预编译成机器语言

首先呢编译型语言啊,它呢在执行这个程序之前,会有一个预告编译的过程通常来说的话呢它会把我们写的源代码编译成相应环境的机器码,那么这个机器码是很接近于底层的,所以它的执行效率是很高的.

解释型语言

javascript, python

解释型语言是不会有预先编译的过程的,也就是说每一次去执行这个程序的时候,都会去 解释这样的一个源代码,所以相应的执行效率是要比编译型语言远要低的.

好那么编译型语言与解释语言到底哪个更好呢,其实这个也是我们以后要讲到一个问题吧,

对于语言来说呐,并没有什么好与坏,编译型语言有编译型语言适用的场景,解释型语言也是,那么这两种语言的特点呢,比如说我们要做一个偏底层 系统级级别 对运行效率要求比较高的 我们一 网页开发 上层应用的话呢 解释型语言是比较适用的.

其它 中间型语言

JAVA C#

在各种编程语言当中语言是人类人为界定的,区分语言分类不重要。重要的是根据自己的需求选择适合自己的语言。

存在的问题.

既然编译型语言执行效率更高,那么为什么还会有解释型语言的存在

- 通常我们说到的效率,这个放到我们具体的研发写代码里来讲的话 它是可以分成两类的,一类是程序的执行效率与而另一类是项目的开发效率,这两个效率通常是鱼与熊掌不能兼得,越是越高级的语言,像python PHP JAV 这样的一些语言它们呢 本质上来讲都是做了大量的封装的,为什么要有封装呢这就是为了我们程序员在写代码的时候 开发效率更高,你要去实现某一个功能的时候,你不需要每一个子功能都亲自去实现,你可以调用你所使用的语言中已经封装好的功能类库,那么正是大量的封装方便了我们的开发效率,那么相应的运行效率就会偏低,所以说语言都它适用的场景,如果说我们一味追求运行效率的话呢,那我认为选择汇编或者C应该来说,相对其它语言来说的话是要快很多的,但是我们想一想,如果我们现在用汇编或者C来开发一个WEB程序或者一个简单的应用,你觉得你的开发效率高吗? 我们现在开发一些简单的应用程序时都是借用某些语言的框架去开发的,不会从用汇编或者C语言从底层一点点重复去做一些别人已经开发的事情的。这道理就像假如给你一个任务要去开拓一片荒地,你第一时间只会想哪种现有的工具可以更高效率地帮助到你完成任务,而不是想到要重新发明制造一个开荒工具。这现有的工具在编译的世界它就是标准库, 第三方类库等。

•

为什么喜欢python

- 简洁 灵活 优雅 哲学

pyhton 之禅

simple is better tnan complex

简洁胜于复杂

Now is better than never,Although never is often better than right now

做也许好过不做,但不假思索就动手还不如不做

- 易于上手难于精通

要完成一个功能方法和 方式 是很多,但是要写出性能最高语法最简洁的代码来完成一个功能呢 这是一个比较难的一个点呐放在pthon里尤其如此 ,python可以让你通过快速的学习,马上就能实现一个简单的功能,但是呢你想写出一些优秀的代码还是有一定的难度,好 那么为什么 要说易于上手难于精通这样的一个特点呢,这就像大家去玩一个新的游戏一样,如果这个游戏太难,那大家就对这个游戏就没什么太多的兴趣了,因为你光要去了解这个游戏的机制呢都非常花时间,它很容易让人失去兴趣,但是一个优秀的游戏非常容易上手,但是呢你玩过一段时间之后,你会发现你要更高的追求的话,还要付出更多的努力,而python就是如此

- 动态脚本的特性,又有面向对象的特性

python 能做什么

- 爬虫

这一个应该是python最知名的一个应用场景了,我们知道呢现在的互联网数据是非常重要的,所以说呢以数据为中心的场景区5花8门,各种各样非常多可以这么说.爬虫在数据时代扮演着非常重要的角色以爬虫为中心的产品也是非常之多.下面举几个例子:比如大家最常见用得最多的搜索引擎它的核心就是这样的一个爬虫, 资讯新闻这样的一些聚核产品, 它们从本质上来讲的话呢 也是爬虫

- 科学计算

随着NumPy, SciPy, Matplotlib, Enthought librarys等众多程序库的开发, Python越来越适合于做科学计算、绘制高质量的2D和3D图像。和科学计算领域最流行的商业软件Matlab相比, Python是一门通用的程序设计语言, 比Matlab所采用的脚本语言的应用范围更广泛, 有更多的程序库的支持。虽然Matlab中的许多高级功能和toolbox目前还是无法替代的, 不过在日常的科研开发之中仍然有很多的工作是可以用Python代劳的。

- 大数据与数据分析 (spark)

大数据处理的时候, 可能会用到spark语言框架, 而python spark所支持的语言当中的其中一种;

- 自动化运维与自动化测试
- 机器学习:Tensor Flow

机器学习这个概念也是最近很火的一个概念,如,艾法狗,其实机器学习也是有很多框架,类库,可以选择不同的语言.如果我们不是专业的从事机器学习,只是说有兴趣的话 那么我们选择语言肯定要选择一种比较大众化,容易入门的语言,像python就是比较好的机器学习的入门语言,python也是谷歌主推的框架开发的首选语言

- 脱水语言:

如开发一个项目的时候,我们曾经用其它的语言编写了某些指令模块的话,再开发新的项目时,是没有必要重新编写这些指令功能模块的,因为已经存在直接调用即可,但是问题是可能存在一个问题这些功能指令模块可能是C++ JAVA编写的,此时python就能像脱水一样将些模块粘合在一起.

怎么对待python

当你遇到问题时,随手拿起python,,编写一个工具,这才是python真正的打开方式.

课程内容

- 基础语法

有很多同学认为基础语法比较简单, 顺便看一些资料就足够, 但是基础语法对于一种语言学习来说的话呢 是非常重要的, 而且也没有同学们所相像中那么简单的, 或许高深的知识点

在编程的初级阶段可能了解一些简单的基础语法就可以了, 但是随着编程的日积月累, 你越到后面呢你就会发现基础语法有多么的重要。我们有很多时候编程编到一定阶段的时候呢就会遇到一个瓶颈, 那么这样一个瓶颈的引起的因素可能有很多个原因, 但是其中一个非常重要的原因就是基础知识还不够扎实,

所以说

基础语法是任何语言的基础, 只有熟练掌握, 才能灵活运用语言写出高效优美简洁的代码。

pythonic

交换变量值的示例

高性能与优化

同样的一个功能, 可以有数个及至数十种写法, 但每种写法的性能与美观度是不同的, 选择性能最高又易于理解的写法才是最正确的。

最后

我们处于这样一个时代，框架太多类库太多技术太多，容易让人在学习语言的时候产生困惑，既然如此更应该抓住一门语言，把它学精通，当你精通一门语言的时候，很多时候其他语言也会变得触类旁通，你在学习其他语言的时候会发现它们有很多相同点。

回归语言的本质，享受语言本身的纯粹之美。

python的未来前景

当我们说到一门语言的前景时,主观性呢是非常强的,通常呢我们比较喜欢哪种语言时,潜意识是会认为这门语言很有发展前景的

- 用python做爬虫 做一些小工具,,随着数据时代的来临,越来越多开发者用python去做数据挖掘分析的项目 ,在未来的人工智能的兴起,python也必将会有更好的前景
- 另外现在计算机硬件性能很多时候都存在一些过剩的情况 ,不是说所有的项目都需要非常高性能的计算机,一旦我们的项目不是对性能要求很高的情况下,我们更关注的是项目的开发效率.