<https://blog.csdn.net/rl529014/article/details/51344881>

Qt 是一个跨平台的C++应用程序框架，支持Windows、Linux、Mac OS X、Android、iOS、Windows Phone、嵌入式系统等。也就是说，Qt 可以同时支持桌面应用程序开发、嵌入式开发和移动开发，覆盖了现有的所有主流平台。你只需要编写一次代码，发布到不同平台前重新编译即可。  
  
Qt 不仅仅是一个GUI库，它除了可以创建漂亮的界面，还有很多其他组件，例如，你不再需要研究STL，不再需要C++的<string>头文件，不再需要去找解析XML、连接数据库、Socket 的各种第三方库，这些 Qt 都已经内置了。  
  
Qt 是应用程序开发的一站式解决方案！  
  
Qt 虽然庞大，封装较深，但它的速度并不慢，虽不及MFC，但比Java、C#要快。Qt 程序最终会编译成本地代码，而不是依托虚拟机。  
  
**Qt 命运多舛，总共经历了三家公司：**

* Qt 由奇趣科技公司（Trolltech）的两位创始人于1990年着手开发，1995年发布Qt  1.0。
* 2008年，奇趣科技被诺基亚公司收购，Qt 也因此成为诺基亚旗下的编程语言工具。
* 在智能手机的冲击下诺基亚大厦轰然倒塌，2012年将 Qt 转让给Digia公司。

Digia 是一家总部位于芬兰的IT业务供应商，每天向全球数以百万的人提供通讯技术的及时解决方案，业务涉及电信、工业、贸易、金融等，客户包括诺基亚、索尼爱立信、惠普、甲骨文等全球一流企业。 超过1200名研发精英分别位于芬兰、瑞典、爱沙尼亚、俄罗斯、中国等地为客户提供端到端的服务。

**Qt 可以做什么：**

使用 Qt 开发的程序非常多。1997年，Qt 被用来开发 Linux 桌面环境 KDE，大获成功，使 Qt 成为 Linux 环境下开发 C++ GUI 程序的事实标准。Linux 也是嵌入式的主力军，广泛应用于消费类电子、工业控制、军工电子、电信/网络/通讯、航空航天、汽车电子、医疗设备、仪器仪表等相关行业，这些地方都有 Qt 的影子。  
  
  
在 Windows 下，GUI 解决方案比较多，基于C++的有 Qt、MFC、WTL、wxWidgets、DirectUI、Htmlayout，基于C#的有 WinForm、WPF，基于Java的有AWT、Swing，基于Pascal的有Delphi，还有国内初露头角的 aardio；如果你有Web开发经验，也可以基于Webkit或Chromium将网页转换为桌面程序。  
  
  
**总之，没有哪一种方案能够独霸Windows，用 Qt 来开发 Windows 桌面程序有以下优点：**

* 简单易学：Qt 封装的很好，几行代码就可以开发出一个简单的客户端，而 MFC 封装简陋，还需要了解 Windows API，普遍反映难学。
* 资料丰富：资料丰富能够成倍降低学习成本，否则你只能去看源码，关于 DirectUI、Htmlayout、aardio 的资料就很少。
* 漂亮的界面：Qt 很容易做出漂亮的界面和炫酷的动画，而 MFC、WTL、wxWidgets 比较麻烦。
* 独立安装：Qt 程序最终会编译为本地代码，不需要其他库的支撑，而 Java 要安装虚拟机，C#要安装 .NET Framework。
* 跨平台：如果你的程序需要运行在多个平台下，同时又希望降低开发成本，Qt 几乎是必备的。

**下面的程序都使用 Qt 开发：**

WPS、YY语音、Skype、豆瓣电台、虾米音乐、淘宝助理、千牛、暴雪的战网客户端、VirtualBox、Opera、咪咕音乐、Google地图、Adobe Photoshop Album 等。

但是，由于Android本身支持Java，iOS本身支持Objective-C和Swift，所以Qt 在移动端的贡献寥寥无几。

**总起来说，Qt 主要用于桌面程序开发和嵌入式开发。**

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/rl529014/article/details/51344881