

UI编写方式名称	界面语言	界面文件格式	界面绘制器	是否支持高DPI拉伸	后端算法编程语言
Win32 API			GDI		C/C++
MFC			GDI		C/C++
Qt	Qt窗体定义语言	.ui	DirectX/OpenGL	√	C/C++
PyQt	Qt窗体定义语言	.ui	DirectX/OpenGL	√	Python
wxWidgets			GDI		C/C++
wxPython			GDI		Python
Kivy	kv代码	.kivy	OpenGL	√	Python
WinForm			GDI		C#
WPF	XAML语言	.xaml	DirectX	√	C#
网页	HTML语言 + CSS样式表	.html	Chromium=>DirectX/OpenGL	√	JavaScript (默认)
			MiniBlink=>DirectX/OpenGL		Python
			Firefox=>DirectX/OpenGL		C/C++
			IE=>DirectX/OpenGL		C#
			Opera=>DirectX/OpenGL		Java
			Electron=>DirectX/OpenGL		Lua
			Sciter=>DirectX/OpenGL		PHP/ASP/ASP.NET
			还有任何你自己写的界面渲染器 只要你可以画出HTML图形		你自己想用的任何语言 甚至是你自己发明的语言 只要和界面绘制器绑定
备注	拥有界面语言的UI框架 创作页面更容易 没有界面语言：100-500行 有界面语言：50行	.ui .xaml .html都是用<>的， 很类似，都是标记语言家族 .kivy有些特殊， 是类似Python的界面语言， 对用Python的人比较友好	用GDI绘制的：10年以前的技术， 用于在低配置、无显卡环境流畅画图， 那时候没有显卡，所以不能做动画， XP相比Win7就少了很多动画 直到现在GDI都不能做动画， 现在连CPU都自带核心显卡，时代变了， GDI已经完成了历史使命，所以被抛弃	只要是GDI的都不能支持 高DPI拉伸， 所以现在用高分屏 总会有一些应用是模糊的， 他们全都是GDI写的	要分清后端编程语言和渲染语言， 比如网页，界面是HTML写的， 但却可以调用Python、C++，甚至汇编， 方法就是使用不同的界面绘制器。
备注			DirectX和OpenGL是绘制渲染界新宠， 能调用显卡，速度超快， 能画动画，能画3D图形， 浏览器渲染HTML就都用的DirectX或OpenGL， 两者差别：DirectX只能用于Windows， OpenGL跨平台，支持Windows、Linux、 Mac、Android、iOS、甚至STM32， 但OpenGL在Windows上略差于DirectX 同一个界面绘制器可能会 同时支持OpenGL和DirectX， 以便支持最多机型		HTML默认以JavaScript作为 后端编程语言，但这不代表HTML 只能搭配JavaScript， 我的方法是修改浏览器， 使浏览器能和C++和Python沟通， 从而实现HTML配合C++或Python编程 顺带一提，现在微信电脑版、 网易云音乐、迅雷都是通过修改浏览器 来实现联合开发的，毕竟HTML是 目前开发效率最高的界面语言。
备注					再顺带一提，我之前说网页慢， 是指后端JavaScript慢， 前端HTML绘图快得很，毕竟用的是 DirectX或者OpenGL。 你完全可以用HTML搭配 C++来开发，速度超神！ 再再顺带一提，JavaScript速度真不慢， 和Python处于同一个数量级， 还比Python稍微快一些， 编程语言执行效率： C/C++ >> C#/Java >> Python/JavaScript 再再再顺带一提， JavaScript和Java没有任何关系， 当年JavaScript是仿照Scheme设计的， 可当时Java大火，为了蹭热度， 起名JavaScript

可以使用的IDE	商用是否付费	是否广泛使用	是否开源	背后是谁撑腰
Visual Studio/VS Code/CLion/Dev-C++/CodeBlocks		否，没人直接用Win32与UI，都是被别的UI库封装		微软（已抛弃此框架）
Visual Studio/VS Code/CLion/Dev-C++/CodeBlocks		否，10年前很火，现在冷门。现在MFC=MeiFanChi没饭吃		微软（已抛弃此框架）
Visual Studio/VS Code/CLion/Dev-C++/CodeBlocks	√	是，目前很热门	√	Qt Company
Visual Studio/VS Code/PyCharm/Jupyter Notebook	√	是，目前很热门	√	Qt Company
Visual Studio/VS Code/CLion/Dev-C++/CodeBlocks		否，冷门	√	开源社区
Visual Studio/VS Code/PyCharm/Jupyter Notebook		否，冷门	√	开源社区
Visual Studio/VS Code/PyCharm/Jupyter Notebook		国外很热门，很火的技术，但不支持中文显示	√	开源社区
Visual Studio/VS Code		否，10年前很火，是当年C#程序员标配，现在没人用		微软（已抛弃此框架）
Visual Studio/VS Code		是，目前很热门		微软
Visual Studio/VS Code/WebStorm/Dreamweaver		一直很热门	√	开源社区 W3C Organization Google FaackBeek Meta Microsoft (最近开始支持) (难得有竞争对手为同一个技术站台)
Visual Studio/VS Code/PyCharm/Jupyter Notebook				
Visual Studio/VS Code/CLion/Dev-C++/CodeBlocks				
Visual Studio/VS Code				
Visual Studio/VS Code/IntelliJ IDEA				
Visual Studio/VS Code				
你自己设计的IDE				
IDE只与后端编程语言有关， 和前端界面框架没有关系。 e.g. 可以用Visual Studio开发网页	商用付费很重要， 没注意到License就商用 程序做出来可能会被 被起诉。 推荐学一个不需要商用付费 的语言，比如Qt就万万不要	广泛使用，而且热门挺重要的， 查资料会方便	开源框架有个好处： 在撑腰的人抛弃之后， 如果用的人多，这些开发者 可能会另起炉灶， 继续维护	如果是公司撑腰，有被抛弃的风险， 再加上大多数公司从来不开源， 抛弃后即使开发者众多， 也无法志愿维护。e.g. WinForm
			e.g. 开发者不满PyQt 商用收费，就自己搞了一个 PySide，兼容PyQt而且 商用免费 再 e.g. Dev-C++被原公司 BloodSheild抛弃， 国内大佬使用源码 继续更新，最近4年一直更新不间断	顺带一提，微软每过几年就出新技术， 旧技术往往会被抛弃，且不开源。 所以行业内程序员对微软的框架 有一种不信任感，直到现在， 虽然技术上WPF很厉害， 但大家不放心用，毕竟是微软出的。
				而如果是开源社区，虽然不至于被抛弃 （社区是众多开发者构成的， 没有人有权利终止一个项目）， 但也不可能不温不火没人用。 e.g. wxWidgets 如果是开源 + 公司撑腰， 那就很棒。e.g. 网页