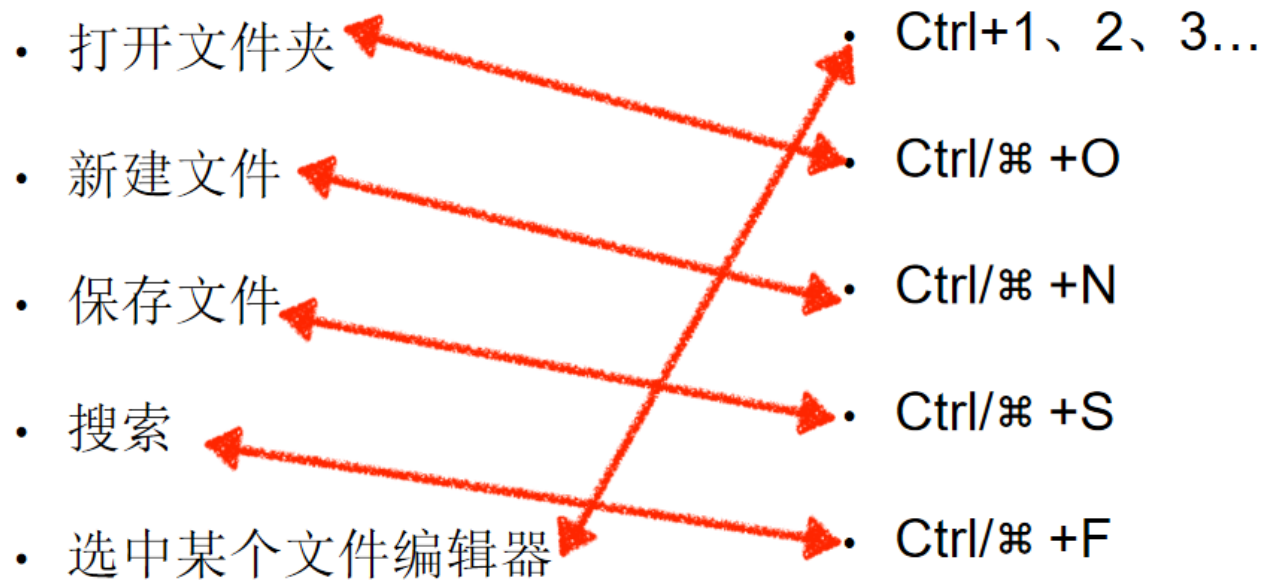
工欲善其事必先利其器

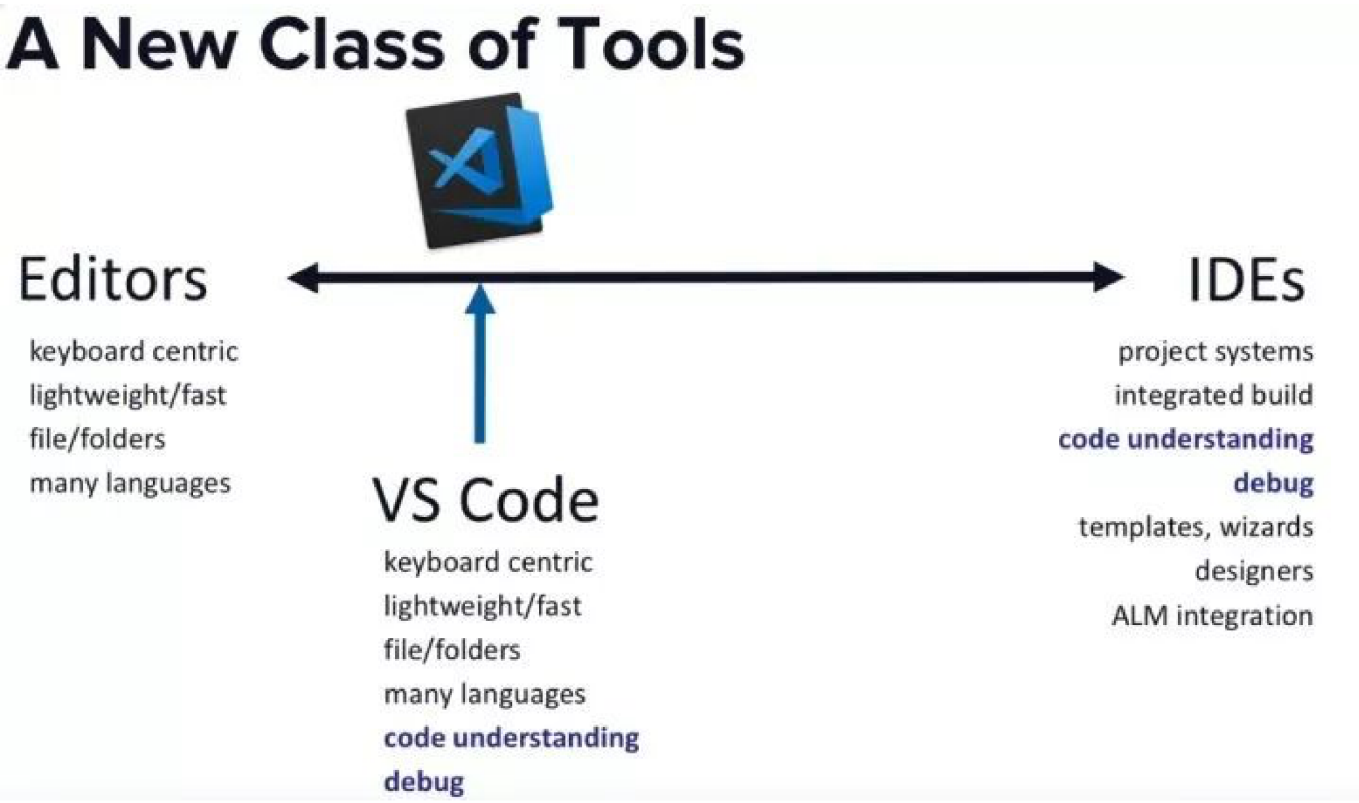
VScode

五大主菜单分别是 文件夹，搜索，git, 调试，插件。

文件夹 ctrl+shift+E 搜索 ctrl+shift+F git ctrl+shift+G 调试 ctrl+shift+D 插件 ctrl+shift+X 查看错误和警告 ctrl+shift+M 查找并运行所有命令 ctrl+shift+P 切换集成终端 ctrl+'



打开文件夹ctrl+o
新建文件ctrl+n 关闭文件ctrl+w ctrl+shift+A代码块注释和取消注释



编辑器+代码理解+版本控制+远程开发+调试

稳定性
VScode是个Node.js环境，是单线程的，任何代码崩了都是灾难性的后果。
VScode把插件都放在单独的进程中。

一致的用户体验
UI界面渲染与业务逻辑隔离。 所有用户的操作被转为各种请求发送给插件进程，插件能做的就是响应这些请求，插件只能专注于业务逻辑处理，不能影响界面元素如何被渲染。

代码理解与调试
Language Server Protocal(LSP) 多语言支持的基础 1.节制的设计 2.合理的抽象 3.周全的细节
LSP最重要的概念是动作和位置。 抽象成请求和回复

remote development(VSCRD)
可以在远程环境开一个VS Code工作区，然后用本地的VSCode连上去工作。

git

git是分布式版本控制系统
本地repo中的branch与一个或多个远程repo中的branch存在跟踪关系
场景一：git本地版本库
git init
git status 查看当前工作区状态
git add [FILES] 文件添加到暂存区
git commit -m "something"
git log
git reset --hard HEAD^^/commit-id 回退
git reflog 当前HEAD之后的提交记录，便于回到未来

场景二：git远程版本库
一个单人项目，要么在本地提交代码到仓库，要么通过web页面更新远程仓库，并且两种不会同时发生。

场景三：团队项目中的分叉合并

两种合并方法
1.快进式合并

2.非快进式合并 (强制产生一次新的提交)
git merge --no-ff mybranch

克隆或同步最新代码到本地存储库

git clone https://something.git
git pull

为自己工作创建一个分支，该分支只负责单一功能模块或代码模块

git checkout -b mybranch
git branch

该分支上完成某单一功能模块开发工作

git add FILES
git commit -m "aaa"

切换回master分支，将远程origin/master同步最新到本地，合并mybranch到master分支，Push到远程origin/master

git checkout master
git pull
git merge --no-ff mybranch
git push

场景四：git rebase
git rebase -i [startpoint][endpoint]

场景五：Fork+Pull request

1. 先fork别人的仓库，相当于拷贝一份
2. 做一些代码贡献
3. 发起pull request给原仓库
4. 原仓库所有者review pull request,如果没有问题，就会merge到原仓库中

Vim

命令模式 输入模式 底线模式

正则表达式十步通关

基本的字符串搜索

VS Code 跨文件搜索(ctrl+shift+F) 文件内搜索(ctrl+F)
区分大小写 是否全字匹配

同时搜索多个字符串

使用|符号

匹配字符串时大小写问题

可以直接设置

通配符基础

.任意一个字符
?一个或0个 +一个或多个 *0，1，或者更多
3到5次的a，a{3,5}h
仅3次的a，a{3}h

匹配具有多种可能性字符集

用方括号[and]来定义一组希望匹配的*字符集*。如匹配 bag big bog 可以创建表达式/b[aiu]g

字符-
[a-e] [0-5]

字符^ ~ 否定字符集
[^aeiou]排除元音的所有字符

\w ~ 所有字母数字[A-Za-z0-9_] **包括下划线_**
\W 与 \w 互补

\d 所有数字 [0-9] \D 与 \d 互补

懒惰匹配和贪婪匹配

greedy ~ 找到符合模式的最长可能部分
lazy ~ 找到符合模式的最小可能部分

正则表达式默认为greedy模式

可以使用?字符将greedy改为lazy懒惰匹配。如t[a-z]*?i

特殊位置和特殊字符

^字符串开头
$字符串末尾

Ricky is first and can be found
查找开头的Ricky为^Ricky 查找结尾的found为found$

\s搜索空格，包括了换行等与[\r\t\f\n\v]类似
\S 与 \s 互补

使用捕获组复用

用(and)定义捕获组，指定重复开始的位置使用\和数字（数字从1开始） 1 1 1 1 22 22 3 432 343 54 使用capture groups捕获组来匹配字符中连续出现三次数字
(\d+)\s\1\s\1$

字符串搜索替换

跨文件替换(ctrl+shift+H) 文件内替换（ctrl+H)

复用捕获组进行替换

保留搜索字符串中的某些字符串作为替换字符串的一部分，使用$访问替换字符串中的捕获组 如$1使用了第一个捕获组
将HTML标签中h改为大写 可用<h(\d)>和<H$1>