



Git

`git status` 🖱️ 看当前所在的分支

`git checkout xxx` 🖱️ 切换到 xxx 分支

`git checkout -` 🖱️ 切换到之前的分支

`git add .` 🖱️ 添加所有改动的文件到缓存区

`git commit -m "xxxx"` 🖱️ 将“暂存区”的改动**真正记录下来**，形成一个提交（commit），保存在本地仓库（这一步是说清楚提交的目的）

`git push` 🖱️ 把你本地的提交 push 到自己的远程仓库

```
git add index.html      # 第一步：添加到暂存区
git commit -m "更新主页内容" # 第二步：提交到本地仓库
git push origin main    # 第三步：推送到远程仓库
```

- 跳过 `git add`，就无法提交新改动
- 不 `push`，别人是看不到你做的改动的（远程仓库没更新）
- 每次 `push` 之前都要 `commit`，而 `commit` 之前要 `add`

IDE的快捷键

点 2 下 `shift` 键 🖱️ 在 idle 中全文搜索

`alt + enter` 🖱️ intelliJ会自动展示一些可优化的tips

- 万用键: Alt+Enter
- Search everywhere: 双击Shift
- 查看定义: Declaration
- 查看文件: Navigate – File
- 高级查找: Find in Path
- 快速生成: Generate
- 格式化: Reformat Code
- 优化导入语句: Optimize imports


- 导航: Navigate – Back/Forward
- 谁调用了这个方法: Call Hierarchy
- 所有的实现类: Implementation
- 文件大纲: File Structure
- 下一处错误: Next Highlighted Error
- 高级重命名: Rename
- 调试器快捷键: F5/F6/F7/F8

想知道快捷键是什么: 点击左上角的intelliJ IDEA👉setting👉keymap

Git相关操作

- 查找背锅侠：Annotate/Blame
- 查看当前文件的历史版本
- 高级筛选方式查找commit记录
- 显示差异：show diff
- Open in GitHub

▼ debug操作

图标	功能	说明
 Resume	继续执行程序直到下一个断点或程序结束	
 Step Over	执行当前行，然后停在下一行（不会进入函数内部）	每按一下就运行一行代码
 Step Into	进入方法内部进行调试	不会直接执行调用函数，而是进入所调用的函数的声明部分一步步执行
 Step Out	跳出当前方法，回到上层调用	