

# *Head First Open Source*

**Open Source culture and Open Source community**

Event Hosted By  
XinYuan Wang  
Microsoft Learn Student Ambassador



# 什么是开源

「开源」一词对应英文 Open Source，最初起源于软件开发领域，因此也称为「开放源代码」，对应的软件则称为**开源软件**（Open Source Software）。



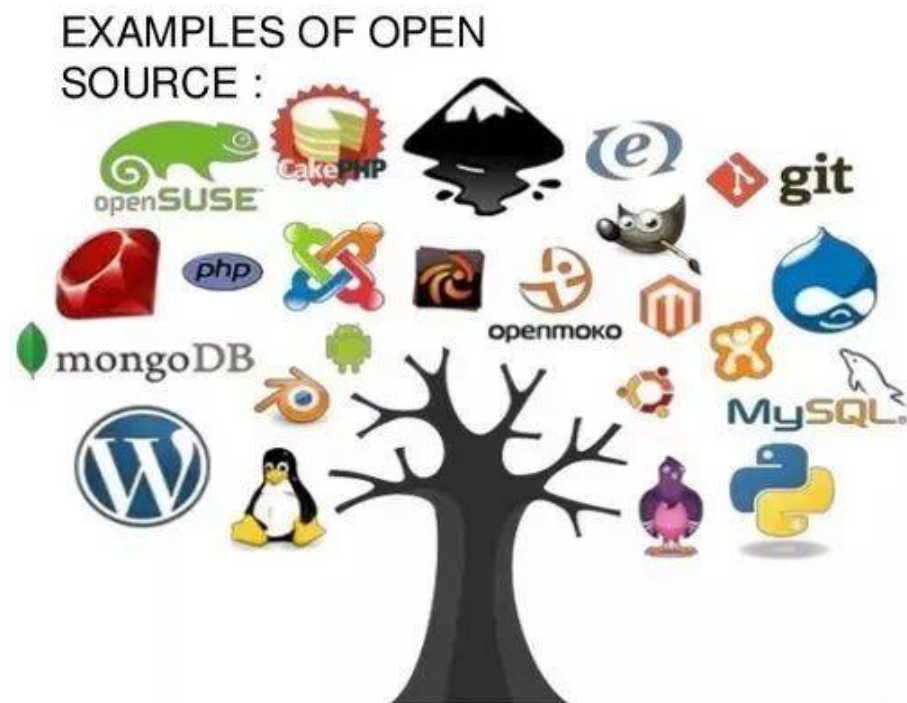
# 什么是开源

按照 [OSI 组织 \(opens new window\)](#) (Open Source Initiative Association) 的 [OSD 定义 \(opens new window\)](#), 除了公开源代码, 开源软件的发行条款还必须符合以下十个条件:

序号	条款	简单说明
1	Free Redistribution	允许自由地再发布软件
2	Source Code	程序必须包含所有源代码
3	Derived Works	可以修改和派生新的软件
4	Integrity of The Author's Source Code	发布时保持软件源代码的完整性
5	No Discrimination Against Persons or Groups	不得歧视任何个人或团体
6	No Discrimination Against Fields of Endeavor	不得歧视任何应用领域 (例如商业)
7	Distribution of License	许可证的发布具有延续性
8	License Must Not Be Specific to a Product	许可证不能针对于某一个产品
9	License Must Not Restrict Other Software	许可证不能限制其他软件
10	License Must Be Technology-Neutral	许可证必须是技术中立的

# 什么是开源

开源软件是一种“技术和立场中立”的使用“许可证约束”的开放源代码的软件。



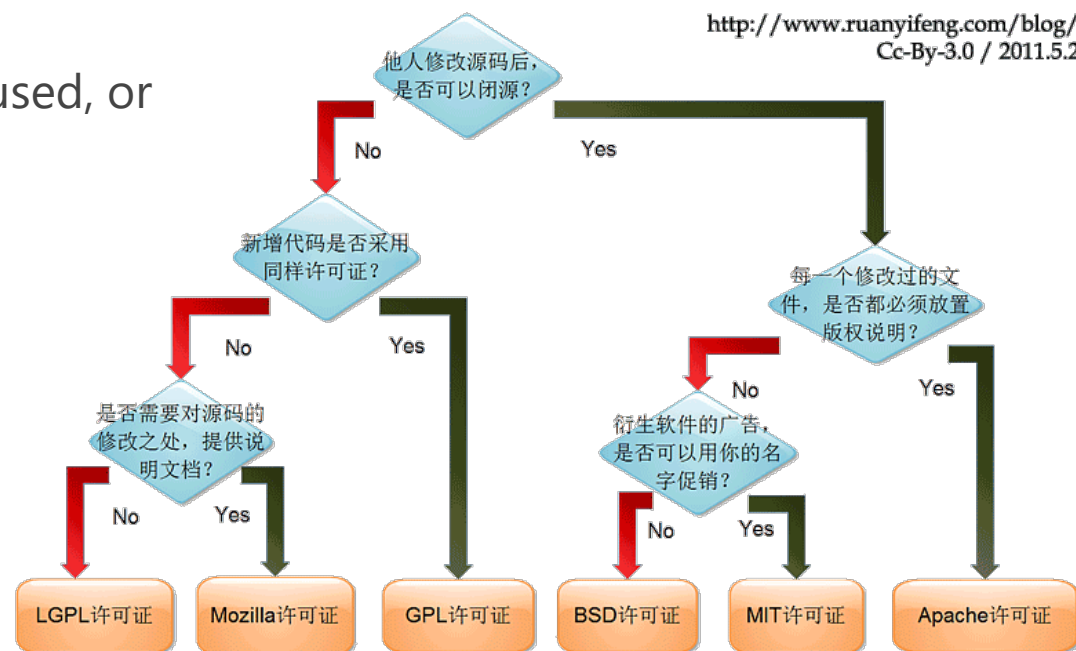
# 什么是开源协议

开源许可协议主要分为宽松许可协议（Apache、BSD、MIT 等）和严格许可协议（GPL、GPL v3、LGPL、Mozilla 等）两大类。

## Popular Licenses

The following OSI-approved licenses are popular, widely used, or have strong communities:

- [Apache License 2.0](#)
- [BSD 3-Clause "New" or "Revised" license](#)
- [BSD 2-Clause "Simplified" or "FreeBSD" license](#)
- [GNU General Public License \(GPL\)](#)
- [GNU Library or "Lesser" General Public License \(LGPL\)](#)
- [MIT license](#)
- [Mozilla Public License 2.0](#)
- [Common Development and Distribution License](#)
- [Eclipse Public License version 2.0](#)



# 什么是开源协议

宽松式许可证 (permissive license) 是最基本的类型，对用户几乎没有限制。用户可以修改代码后闭源。

它有三个基本特点。

(1) 没有使用限制

用户可以使用代码做任何想做的事情。

(2) 没有担保

用户自担风险。

(3) 披露要求 (notice requirement)

用户必须披露原始作者/贡献者。

如 Apache、BSD、MIT

严格式许可证 (copyleft license) 修改后的 Copyleft 代码不得闭源。

如 GPL、GPL v3、LGPL、Mozilla



# 什么是开源软件托管平台

以 Github 为例

GitHub是一个面向[开源](#)及私有[软件](#)项目的托管平台，因为只支持Git作为唯一的版本库格式进行托管，故名GitHub。

GitHub于2008年4月10日正式上线，除了[Git](#)代码仓库托管及基本的Web管理界面以外，还提供了订阅、讨论组、文本渲染、在线文件编辑器、协作图谱（报表）、代码片段分享（Gist）等功能。目前，其注册用户已经超过350万，托管版本数量也是非常之多，其中不乏知名开源项目[Ruby](#) on Rails、[jQuery](#)、[python](#)等。





2008年4月10日，GitHub正式上线。

2014年1月23日，联合创始人汤姆·普雷斯顿-维尔纳（Tom Preston-Werner）从另一位联合创始人克里斯·万斯特拉斯（Chris Wanstrath）手中接过总裁职位，后者也将接过普雷斯顿-维尔纳留下的CEO位置。

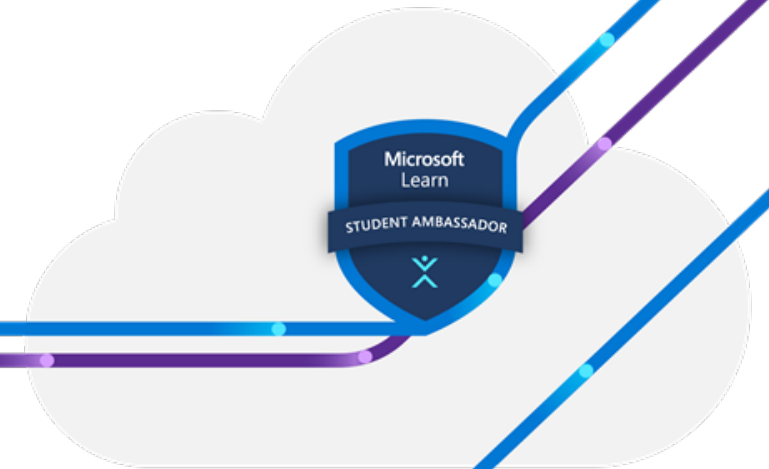
2018年6月4日晚，微软宣布，通过75亿美元的股票交易收购GitHub。<sup>[2]</sup> 10月26日，微软以75亿美元收购GitHub交易已完成。10月29日，微软开发者服务副总裁奈特·弗里德曼（Nat Friedman）将成为GitHub的新一任CEO。<sup>[3]</sup>

2020年3月17日，Github宣布收购[npm](#)，GitHub现在已经保证npm将永远免费使用。



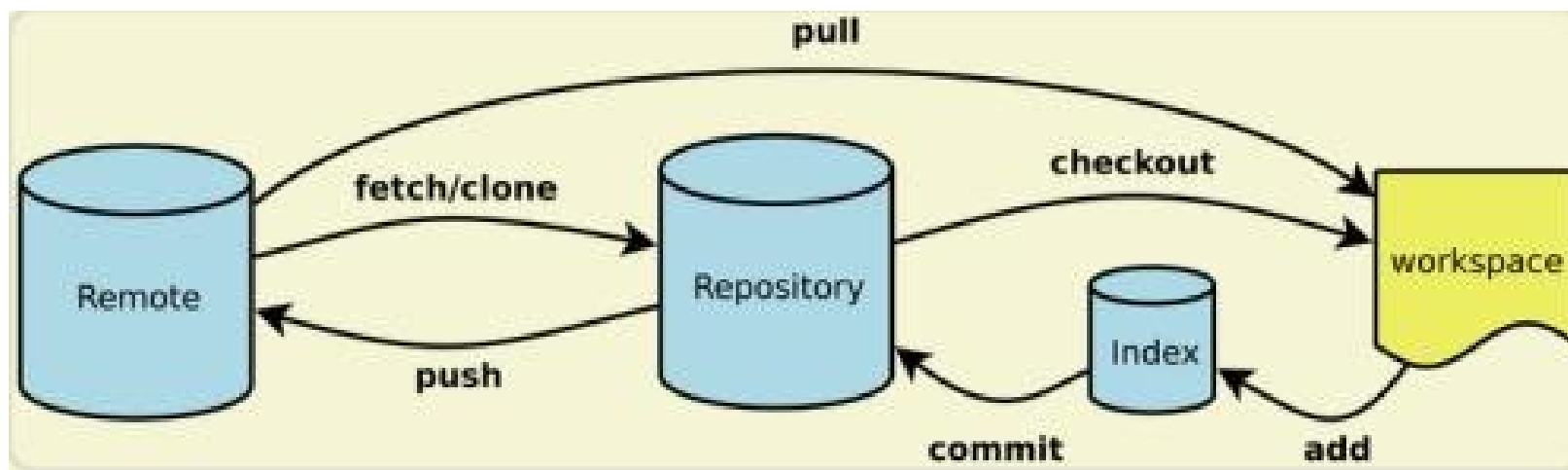


# Please Coding with me , Step by step



# Git 进行代码管理

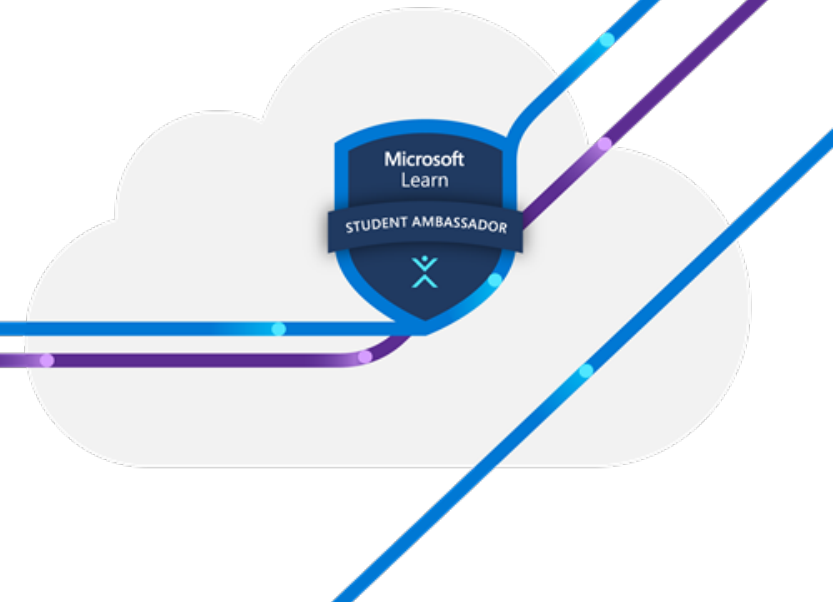
- 1、Git 管理代码，保证代码版本迭代连续性，即：向A分支merge或者push代码时，A分支代码必须是当前代码的上一个版本，不然会产生冲突。（换句话说：Git确保当前的本地的代码为最新）
- 2、Git有修改就有提交，就有新的代码版本，git管理维护的是修改。
- 3、Git分支存储的是代码副本。
- 4、push：实际上就是将本地分支合并到远端库分支；pull：实际就是将远端分支合并到本地分支。



# 常用的 Git 指令

- `git add .` # 添加所有修改文件到暂存区
- `git add [ path ]` # 添加文件
- `git commit -m "XXXX"` # 提交文件
- `git push` # 上传至远端
- `git pull` # 下载至本地
- `git init` # 初始化本地 git 仓库
- `git status` # 查看当前 git 仓库的状态
- `git log` # 查看 git 的提交信息
- `git branch` # 创建 git 分支
- etc

# Talk is cheap. Show me the code



# Thanks for your listening

Xinyuan Wang

