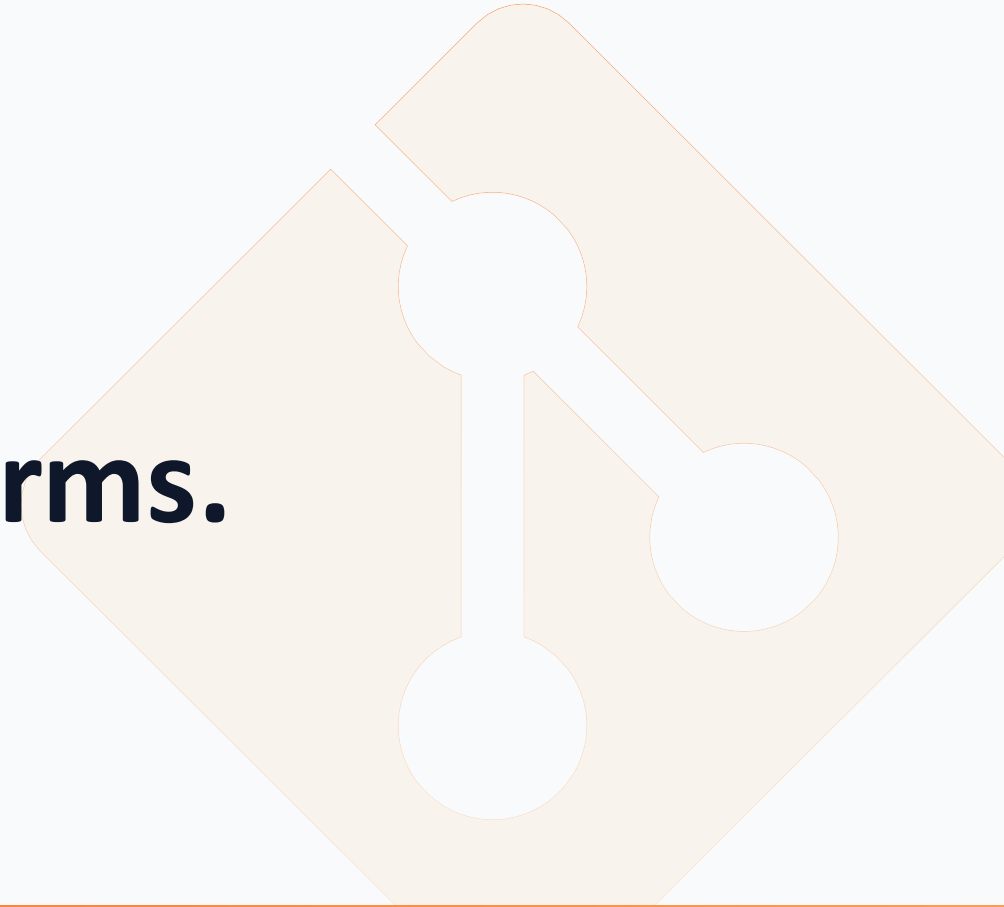


Introduction to Git and Source Code Hosting Platforms.

> 1st CPU Tech Forum _

Baicen LIU

Computer Psycho Union, UNNC.

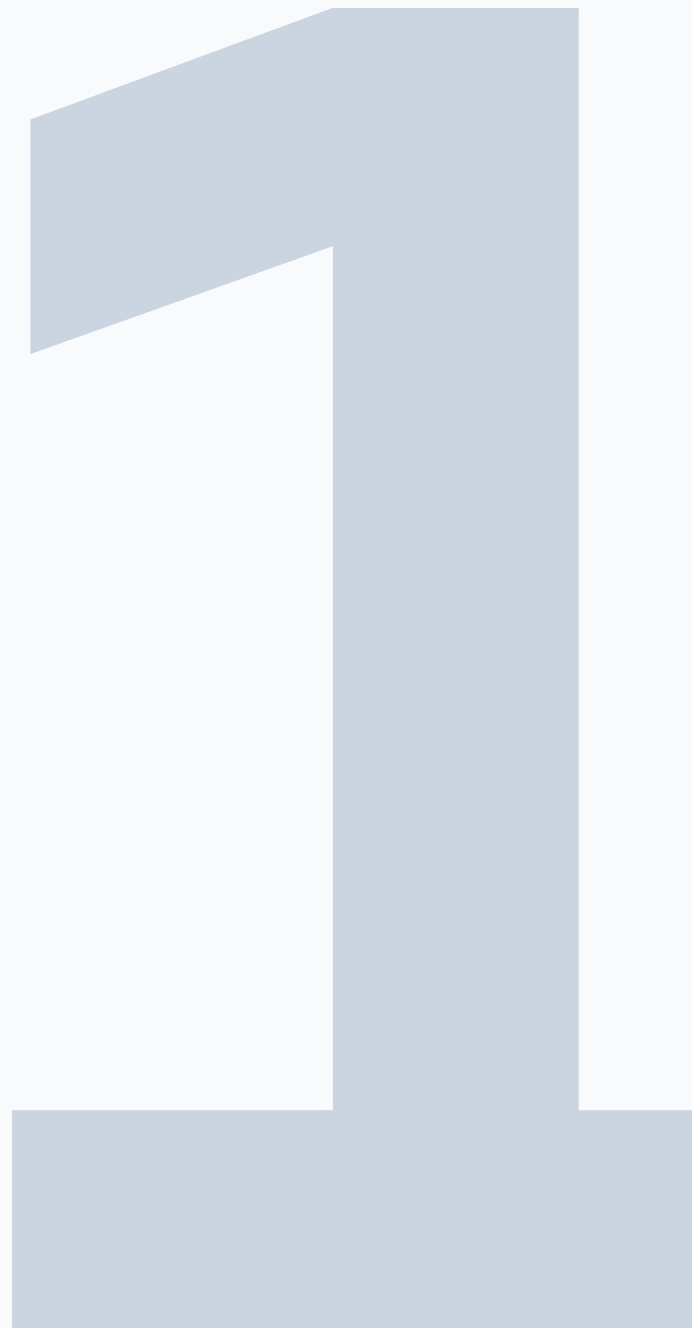


Index

- > **Introduction**
- > **Basic**
- > **Command**
- > **Source-code-host Platform**
- > **Synergy**

Introduction

介绍



Introduction.

> What is Version Control

Version Control(版本控制), 一种在开发的过程中用于管理我们对文件、目录或工程等内容的修改历史, 方便查看更改历史记录, 备份以便恢复以前的版本的软件工程技术。

简单来说就是用于**管理多人协同开发项目的技术**。

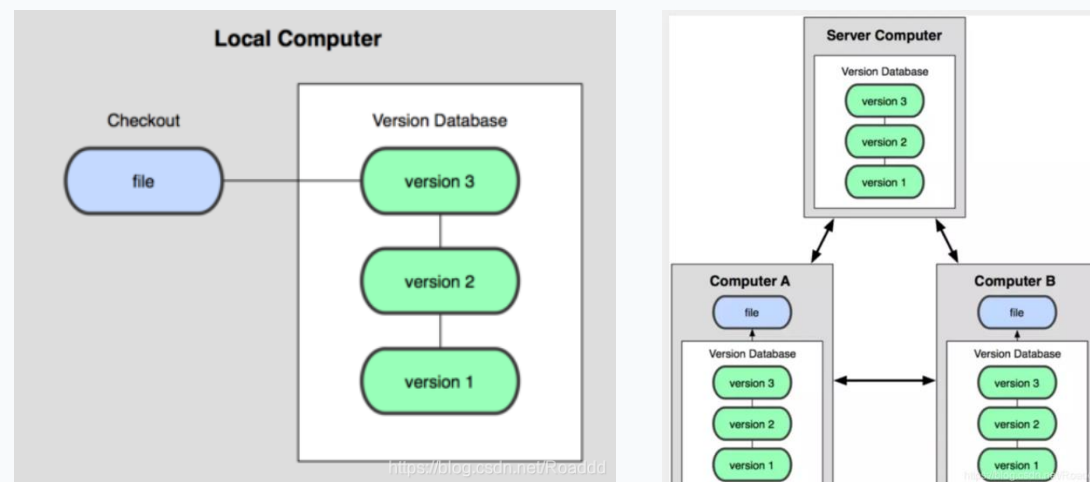
Importance in Development

协同性

可追溯性

并行开发

Local VCS vs Distributed(分布式) VCS



Introduction.

> What is **Git**

Git 是一个开源的分布式版本控制系统(DVCS)，用于追踪文件更改和帮助多人协作开发。
它被设计成能够快速、高效地处理从很小到非常大的项目。

Basics

基础



Basics of Git

> Installation

For Windows:

前往Git官网download page中下载安装器安装:

<https://git-scm.com/downloads>(推荐)

或使用Win-get进行安装(不推荐)

For macOS:

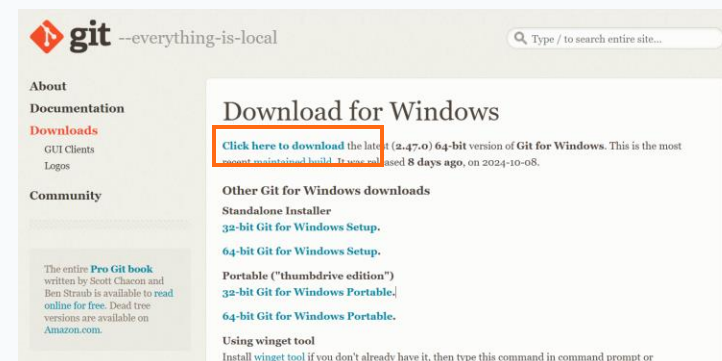
推荐使用brew进行安装:

若你还未在你的电脑上安装过Brew, 请在终端输入Brew安装命令:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

使用Brew安装Git命令:

```
brew install git
```



Basics of Git

> Check Installation

在Terminal / CMD中输入 `git -v` 若回复当前版本号即为正确安装了Git。

> Terminal?

请在你的电脑上找到这两个图标



macOS Terminal



Windows 11 CMD

还没找到??

For macOS, 按住Command + Space键调出Spotlight.

在弹出的搜索框中输入 “终端” / “Terminal”

For Windows, 在开始菜单中搜索 “命令提示符”

Basics of Git

> Key Concept

- **Repository(仓库):**
What is a repository (local vs. remote)?
- **Commit(提交):**
How commits represent snapshots of your project.
- **Branch(分支):**
What is branching in Git? Why it is important?
- **Merge(合并):**
Merging changes from different branches.
- **Staging Area(暂存区):**
How to add files to staging before committing.
- **HEAD:**
The current state of your working directory.

Command

命令



Command of git

> Initialize a repository

```
BASH  
git init
```

初始化Git仓库

Command of git

> Cloning a repository

BASH

```
git clone <repository-url>
```

在本地没有仓库的时候，克隆一份远程仓库到本地。请区别于pull。
对于需要远程获取的仓库，不需要 git init 来初始化

Command of git

> Checking the status

```
BASH  
git status
```

Command of git

> Adding files to the staging area

BASH

`git add <file-name>` //添加某一文件到暂存区

`git add .` //添加目录下所有有更改的文件到暂存区

Command of git

> Committing changes

BASH

```
git commit -m "Your commit message"
```

在团队协作时，请合理的编写commit message，这样可以提高团队协作效率

Command of git

> Branching

BASH

```
git branch <branch-name>
```

```
git checkout <branch-name> //切换到分支
```

```
git checkout -b <new-branch-name> //新建并切换到新分支
```

> Merging branches

BASH

```
git merge <branch-name> //合并分支
```


Command of git

> Pushing(推送) changes to a remote repository

BASH

```
git push origin <branch-name>
```

在进行Push操作前，请务必先拉取(Pull)仓库，检查仓库的最新状态。
若产生版本冲突，处理本地与云端的版本冲突后，再Push!

> Pulling(拉取) changes from a remote repository

BASH

```
git pull
```

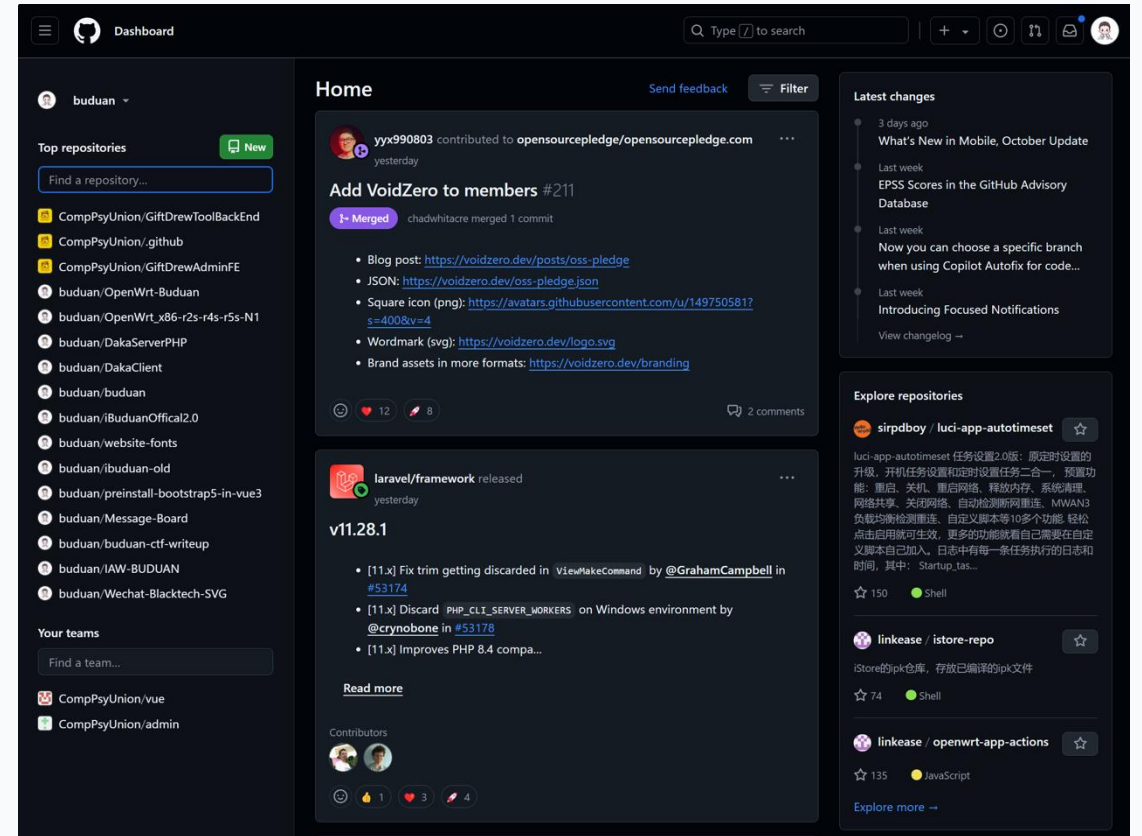
Source-code-host Platform

> Introduction to GitHub

GitHub 是一个软件代码托管平台。

GitHub 是一个存放软件代码的网站。

> Setting up a GitHub Repository



Source-code-host Platform

> Working with GitHub

将本地仓库链接到远程仓库

BASH

```
git remote add origin <repository-url> //添加远程仓库
```

```
git push -u origin master //推送到master分支
```

Source-code-host Platform

> **Advanced Git & GitHub Features**

Hands-on Practice

Exercise 1:

创建一个Git仓库，并提交(commit)第一个更改

Exercise 2:

创建一个分支(branch)，并更改新的分支，然后将它合并回主分支(main branch)。

Exercise 3:

在Github网站上创建一个仓库，并推送(push)本地的更改。

Further Exercise:

Contribute to an open-source project by forking and submitting a pull request.

Synergy

协作

Recommend Reading Article:

<https://developer.aliyun.com/article/1491733>



Synergy

> Use branches effectively

在团队协作中，通常需要采用合理的**分支策略**，建立清晰的**分支模型**。

master

通常是项目中主分支的命名

Feature Branch 功能分支

为每一个功能创建分支，并在开发完成时合并到主分支。

feature/login-page

Git Flow

一种广泛使用的分支模型，具有用于发布、用于持续开发和功能/修补程序分支。

master

develop

XX_fix

基于主干的开发过程

为每一次修改创建分支，并在修改处开发完成时合并到主分支。

master

a

b

c

此外，团队之间还需要保持分支命名的一致性。

Synergy

> Descriptive Commit Messages

在团队协作中，清晰的Commit Messages 往往可以告诉你的同伴你对文件做了什么。这样也会在处理冲突时更方便。

fix :

update :

create :

Synergy

> Handling Merge Conflicts

对于本地拉取时候的冲突，建议与相应的小伙伴通过微信/QQ/钉钉/飞书/企业微信/Teams/HiLink/邮件等方式进行合理沟通后进行修改。

对于云端冲突，建议在团队项目的时候指定一个人来处理冲突，保证“所有的冲突都由ta来处理”。

如果冲突处理不合理，可能会导致仓库中代码混乱。