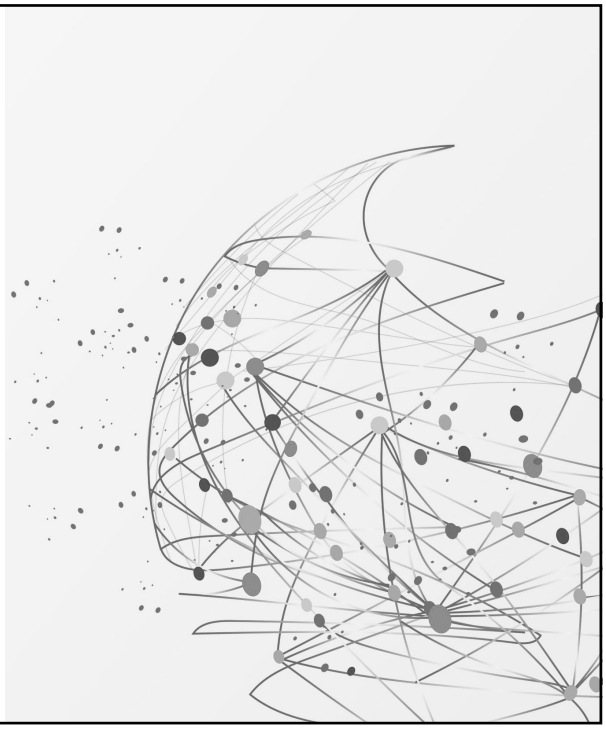
13 - Git & GitHub

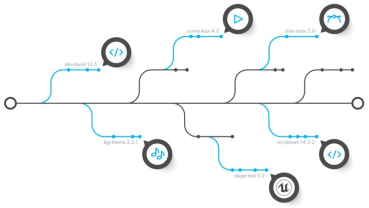
Python Programming and Analytics



1

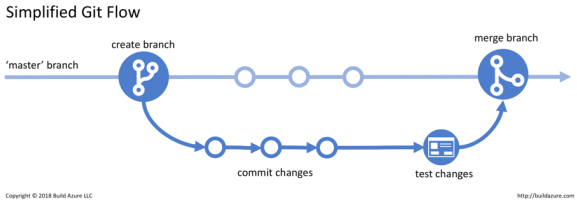
CHAPTER CONTENT
章節內容

- 版本控制 Version control
- Git & GitHub
- Benefit of Git & GitHub



2

版本控制
VERSION CONTROL



- 版本控制是一種管理計算機文件變化的系統。
- 它允許用戶在不同時間點對文件進行修改、保存和恢復。
- 版本控制系統能記錄每一次的變更，並且能回溯到任意一次變更的歷史版本。
- Version control is a system for managing changes to computer files.
- It allows users to modify, save, and restore files at different points in time.
- A version control system can record every change and can go back to the previous version of any change.

3

GIT & GITHUB

- **Git** is an open source distributed version control system that tracks versions of files. It is often used to control source code by programmers collaboratively developing software. 是一個開源分散式版本控制系統，用於跟蹤檔的版本。它通常由程式師協作開發軟體來控制原始程式碼。

GitHub is a widely-used Free-to-use cloud Storage platform with version control and many other essential features that specifically helps developers to manage and deploy their projects on GitHub. 是一個廣泛使用的免費雲存儲平臺，具有版本控制和許多其他基本功能，專門幫助開發人員在 GitHub 上管理和部署他們的專案。

4

BENEFITS OF USING GIT

1. **History Tracking**: Git allows you to track every change made in your project, including: who made the change and when it was made. Git 允許您跟蹤專案中所做的每項更改，包括：誰進行了更改以及何時進行了更改。
2. **Collaboration**: Multiple developers can be able work on the same project at the same time, and Git efficiently manages the merging of changes in code. 多個開發人員可以同時處理同一個專案，並且 Git 可以有效地管理代碼中更改的合併
3. **Branching and Merging**: Git enables developers to create branches to work on new features or bug fixes and later merge them back into the main codebase. Git 使開發人員能夠創建分支來處理新功能或錯誤修復，然後將它們合併回主代碼庫
4. **Offline Work**: Git works offline, which means you can commit changes and work on your project even without an internet connection. Git 可以離線工作，這意味著即使沒有互聯網連接，您也可以提交更改並處理您的專案

5

BENEFITS OF USING GITHUB

1. It makes it easy to contribute to your open-source projects: Using GitHub is free if your project is open source and includes a wiki and issue tracker that makes it easy to include more in-depth documentation and get feedback about your project. 如果您的專案是開源的，並且包含wiki和問題跟蹤器，則可以免費使用 GitHub，從而輕鬆包含更深入的文件並獲取有關項目的反饋。

2. Documentation: By using GitHub, you make it easier to get excellent documentation.

3. Showcase your work: Today, when searching for recruits for their project, most companies look into GitHub profiles. 在為他們的項目尋找新員工時，大多數公司都會查看 GitHub 個人資料。

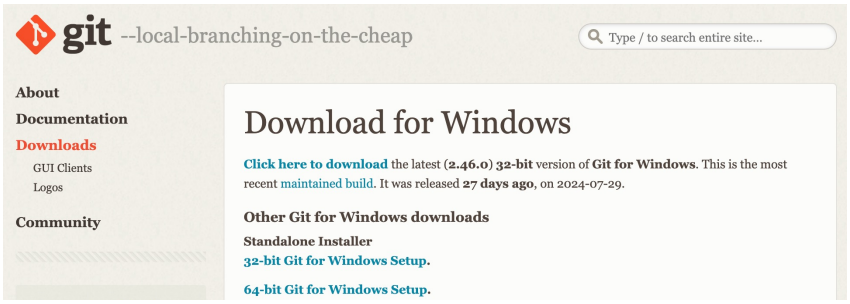
4. Track changes in your code across versions: Keep track of revisions—who changed what, when, and where those files are stored. GitHub takes care of this problem by keeping track of all the changes that have been pushed to the repository. 跟蹤修訂這些檔案存儲的內容、時間和位置

5. GitHub is a repository: it allows your work to get out there in front of the public. Moreover, GitHub is one of the largest coding communities around the globe. 使您的作品可以展示在公眾前。此外，GitHub 是全球最大的編碼社區之一。

6

GIT INSTALLATION

- If you use Mac or Linux, your git is pre-installed with OS. 如果您使用 Mac 或 Linux，則您的OS已預裝git。
- If you use Window, you may need to install Git from 如果您使用 Window，則可能需要從 https://git-scm.com/download/win



7

GIT SETUP

Configuring user information used across all local repositories 配置在所有本地儲存庫中使用的用戶資訊

```
git config --global user.name "[firstname lastname]"
```

set a name that is identifiable for credit when review version history

```
git config --global user.email "[valid-email]"
```

set an email address that will be associated with each history marker

```
git config --global color.ui auto
```

set automatic command line coloring for Git for easy reviewing

8

GIT SETUP & INIT

Configuring user information, initializing and cloning repositories.

配置使用者資訊、初始化和抄錄儲存庫。

`git init`
initialize an existing directory as a Git repository
`git clone [url]`
retrieve an entire repository from a hosted location via URL

9

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

使用快照和 Git 暫存區域

`git status`
show modified files in working directory, staged for your next commit
`git add [file]` git add . git add myfile.py
add a file as it looks now to your next commit (stage)
`git reset [file]`
unstage a file while retaining the changes in working directory
`git diff`
diff of what is changed but not staged
`git diff --staged`
diff of what is staged but not yet committed
`git commit -m "[descriptive message]"` git commit
commit your staged content as a new commit snapshot

10

BRANCH & MERGE

- Isolating work in branches, changing context, and integrating changes
- 在分支中隔離工作、更改上下文和集成更改，創造不同的代碼版本

git branch
list your branches. a * will appear next to the currently active branch
git branch [branch-name]
create a new branch at the current commit
git checkout
switch to another branch and check it out into your working directory
git merge [branch]
merge the specified branch's history into the current one
git log
show all commits in the current branch's history

11

INSPECT & COMPARE

- Examining logs, diffs and object information
- 檢查日誌、差異和對象資訊

git log
show the commit history for the currently active branch
git log branchB..branchA
show the commits on branchA that are not on branchB
git log --follow [file]
show the commits that changed file, even across renames
git diff branchB...branchA
show the diff of what is in branchA that is not in branchB
git show [SHA]
show any object in Git in human-readable format

12

TRACKING PATH CHANGES

- Versioning file removes and path changes
- 對檔案進行版本控制、刪除和路徑更改

git rm [file]
delete the file from project and stage the removal for commit
git mv [existing-path] [new-path]
change an existing file path and stage the move
git log --stat -M
show all commit logs with indication of any paths that moved

13

IGNORING PATTERNS

- Preventing unintentional staging or commiting of files
- 防止意外暫存或提交檔

logs/ ***.notes** **pattern*/**
Save a file with desired patterns as .gitignore with either direct string matches or wildcard globs.
git config --global core.excludesfile [file]
system wide ignore pattern for all local repositories

14

SHARE & UPDATE

- Retrieving updates from another repository and updating local repos
- 從其他存儲庫檢索更新並更新本地存儲庫

`git remote add [alias] [url]`
add a git URL as an alias
`git fetch [alias]`
fetch down all the branches from that Git remote
`git merge [alias]/[branch]`
merge a remote branch into your current branch to bring it up to date
`git push [alias] [branch]`
Transmit local branch commits to the remote repository branch
`git pull`
fetch and merge any commits from the tracking remote branch

15

REWRITE HISTORY

- Rewriting branches, updating commits and clearing history
- 重寫分支、更新提交和清除歷史記錄

`git rebase [branch]`
apply any commits of current branch ahead of specified one
`git reset --hard [commit]`
clear staging area, rewrite working tree from specified commit

16

TEMPORARY COMMITS

- Temporarily store modified, tracked files in order to change branches

臨時存儲已修改的跟蹤檔，以便更改分支

git stash
Save modified and staged changes
git stash list
list stack-order of stashed file changes
git stash pop
write working from top of stash stack
git stash drop
discard the changes from top of stash stack

17

GITHUB ACCOUNT

- **1. Creating an account**
- To sign up for an account, navigate to https://github.com/ and follow the prompts.
- To keep your GitHub account secure you should use a strong and unique password. For more information, see "Creating a strong password."

18

GitHub

GITHUB ACCOUNT

- **Verifying your email address**
- To ensure you can use all the features in your GitHub plan, verify your email address after signing up for a new account. For more information, see "Verifying your email address."
- **Configuring two-factor authentication**
- Two-factor authentication, or 2FA, is an extra layer of security used when logging into websites or apps. We strongly urge you to configure 2FA for the safety of your account. For more information, see "About two-factor authentication."
- Optionally, after you have configured 2FA, add a passkey to your account to enable a secure, passwordless login. See "Managing your passkeys."

19

GitHub

USING GITHUB'S TOOLS AND PROCESSES

- GitHub's collaborative approach to development depends on publishing commits from your local repository to GitHub for other people to view, fetch, and update using Git. For more information about Git, see the "Git Handbook" guide. For more information about how Git is used on GitHub, see "GitHub flow."

20

GitHub

CREATING A REPOSITORY

- A repository is like a folder for your project. You can have any number of public and private repositories in your personal account. Repositories can contain folders and files, images, videos, spreadsheets, and data sets, as well as the revision history for all files in the repository. For more information, see "About repositories."

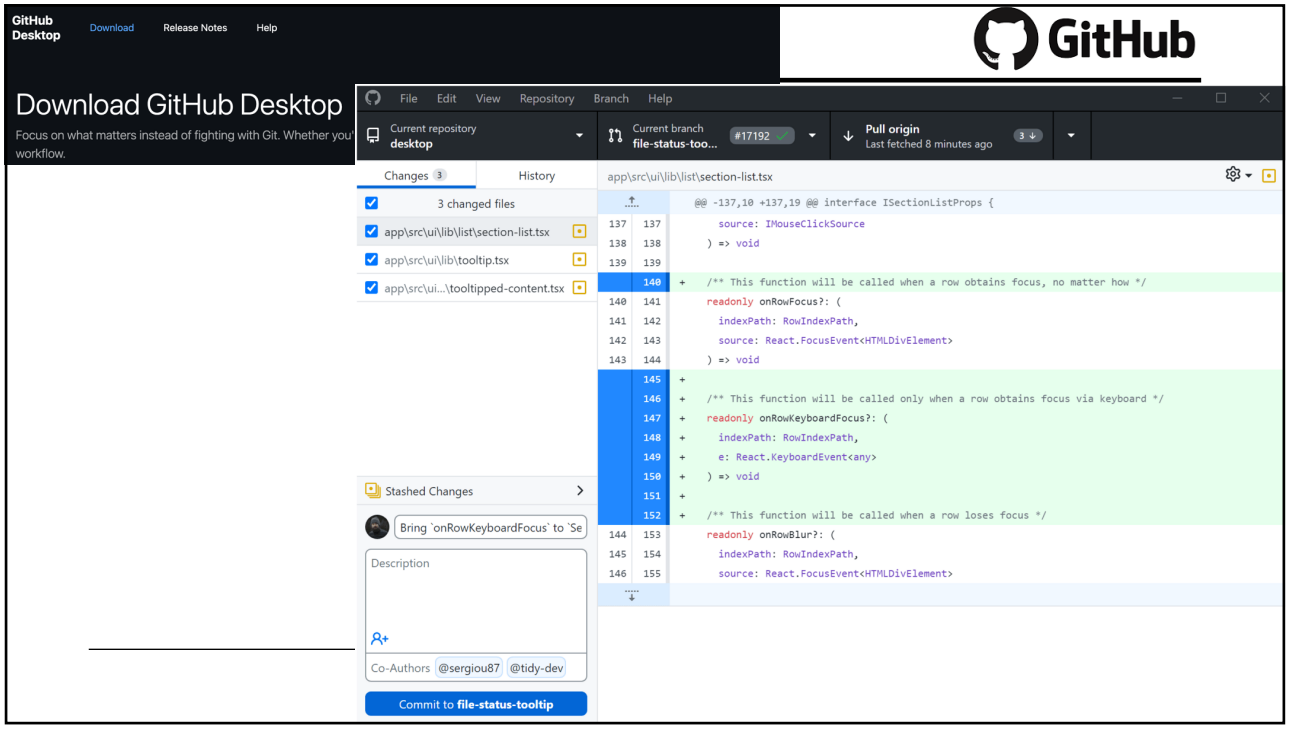
21

GitHub

CLONING A REPOSITORY

- You can clone an existing repository from GitHub to your local computer, making it easier to add or remove files, fix merge conflicts, or make complex commits. Cloning a repository pulls down a full copy of all the repository data that GitHub has at that point in time, including all versions of every file and folder for the project. For more information, see "Cloning a repository."

22


GitHub Desktop
Download
Release Notes
Help
Download GitHub Desktop
Focus on what matters instead of fighting with Git. Whether you'
workflow.
GitHub
File
Edit
View
Repository
Branch
Help
Current repository
desktop
Current branch
file-status-too...
#17192
Pull origin
Last fetched 8 minutes ago
Changes 3
History
3 changed files
app\src\ui\lib\list\section-list.tsx
app\src\ui\lib\tooltip.tsx
app\src\ui...\tooltipped-content.tsx
Stashed Changes
Bring `onRowKeyboardFocus` to `Se
Description
Co-Authors @sergiou87 @tidy-dev
Commit to file-status-tooltip
app\src\ui\lib\list\section-list.tsx
@@ -137,10 +137,19 @@ interface ISectionListProps {
source: IMouseClickSource
) => void
/** This function will be called when a row obtains focus, no matter how */
readonly onRowFocus?: (
indexPath: RowIndexPath,
source: React.FocusEvent<HTMLDivElement>
) => void
/** This function will be called only when a row obtains focus via keyboard */
readonly onRowKeyboardFocus?: (
indexPath: RowIndexPath,
e: React.KeyboardEvent<any>
) => void
/** This function will be called when a row loses focus */
readonly onRowBlur?: (
indexPath: RowIndexPath,
source: React.FocusEvent<HTMLDivElement>

23