

Tool Chain Introduction Table of Contents

1. Git introduction
2. Github introduction
3. VSCode / Cursor
4. Markdown
5. Warp for VSCode
6. Github pages and Mkdocs

Git Intxroduction

- What is Git?
 - i. Git is a free and open-source source control management (SCM) system.
 - ii. It is a distributed version control system for tracking changes in source code.
 - iii. Git's primary function is to track changes to files over time, allowing users to revert to previous versions and compare differences.
- Why Use Git?
 - i. Manage changes to files: Git allows you to see who changed what and compare different versions.
 - ii. Revert to previous versions: If you make a mistake, you can jump back to a previous version.
 - iii. Experiment safely: Use branches to work on new features or fixes without affecting the main code.
 - iv. Track project history: Git keeps a record of all changes made, creating a project history.

- Key Git Concepts
 - i. Repository (Repo): A storage location for source code, documentation, and other files with all its history. Represented by the hidden .git folder.
 - ii. Commit: A snapshot of a repository at a specific point in time, creating a record of changes made.
 - iii. Branch: A parallel version of a project's code, allowing developers to work on features or fixes without affecting the main code.
 - iv. Staging: A transitional area where changes are prepared to be included in the next commit.
 - v. Working Directory: The area where you make edits to your files.

Git installation

- Install Git from [here](#)
- Check if Git is installed by running `git --version` in the terminal

Git commands

- git clone: Clone a repository from a remote source.

```
git clone https://github.com/beets3d/FutureMakers.git
```

- git pull: Pull changes from a remote repository to your local repository.

```
git pull origin main
```

Github Introduction

- What is GitHub?
 - i. GitHub is a web-based platform for version control and collaboration using Git.
 - ii. It provides cloud hosting for Git repositories.
 - iii. It offers tools for project management, issue tracking, and code review.
- Key GitHub Features
 - i. Remote Repositories: GitHub hosts repositories that can be accessed by multiple users.
 - ii. Collaboration: GitHub facilitates collaboration through features like issues, pull requests, and project management tools.
 - iii. Issues: Used to track feature requests, bug reports, or questions related to a project.
 - iv. Pull Requests: A request to merge changes from one branch into another, often used in collaborative workflows.

- Connecting Local Git to GitHub

- i. Create a Repository on GitHub: Set up a new repository on GitHub.
- ii. Add Remote: Connect your local repo to a remote GitHub repo with the command `git remote add origin <github_repo_URL>`.
- iii. Set Main Branch: Set the target branch to main using `git branch -M main`.
- iv. Push to GitHub: Push your local repo to the remote with the command `git push -u origin main`. `git push --all` will push all branches to GitHub.

- Working with GitHub

- i. Making Changes: Edit files directly on GitHub or make changes locally and push them.
- ii. Downloading Changes: Use `git fetch` to download all the remote history or `git pull` to download all remote history and merge it with local files.
- iii. Collaboration: Use issues and pull requests to manage changes and collaborate with others.
- iv. Releases: Create releases to mark specific versions of your project.

Actions

1. Register a Github account
2. Git clone the FutureMakers repository to your local machine

VSCode / Cursor

- VSCode
 - i. VSCode is a code editor developed by Microsoft. It is a powerful and versatile tool that supports multiple programming languages and has a wide range of extensions for enhancing productivity.
 - ii. Download VSCode from [here](#)
- Cursor
 - i. Cursor is an AI-powered code editor developed by Cursor AI. It uses AI to assist with code writing, debugging, and refactoring. Cursor integrates with GitHub, allowing for seamless collaboration and version control.
 - ii. Download Cursor from [here](#)

Clone the FutureMakers repository in Cursor

1. Install Cursor
2. Register an cursor account
3. Git clone the FutureMakers repository
4. Open the repository in Cursor

Markdown

- Markdown is a lightweight markup language with plain text formatting syntax. It is designed to be easy to read, write, and format.
- Markdown syntax: <https://www.markdownguide.org/basic-syntax/>
- Examples:

Syntax	Example
<code># Heading</code>	# Heading
<code>**bold text**</code>	bold text
<code>*italic text*</code>	<i>italic text</i>
<code>[Link](URL)</code>	Link
<code>`code`</code>	<code>code</code>
<code>![Beets Logo](https://beets3d.github.io/FutureMakers/assets/beets-logo-square.avif)</code>	

Warp for VSCode / Cursor

- Marpit /mɑ:rpɪt/ is the skinny framework for creating slide deck from Markdown. It can transform Markdown and CSS theme(s) to slide deck composed of static HTML and CSS and create a web page convertible into slide PDF by printing.
- We can use Marpit in VSCode or Cursor to create slide deck from Markdown with a excellent editing experience. It also helps us to generate slides from Markdown in our own website.
- Install Marpit in VSCode or Cursor: <https://marketplace.visualstudio.com/items?itemName=marp-team.marp-vscode>

Github pages and Mkdocs

- Github pages is a static website hosting service that allows us to host our own website for free.
- We can use MKdocs to build our website and host it on Github pages.
- Install MKdocs by running `pip install mkdocs` in the terminal
- Run `mkdocs serve` to serve the website
- Run `mkdocs build` to build the website. The built website will be in the `site` folder.

Note:

- We need to install the requirements.txt file by running `pip install -r requirements.txt` in the terminal.
- If you are using python 3.10 or above, you should make your virtual environment