Coding Bootcamp

Git

Git File Management

Learning Objectives

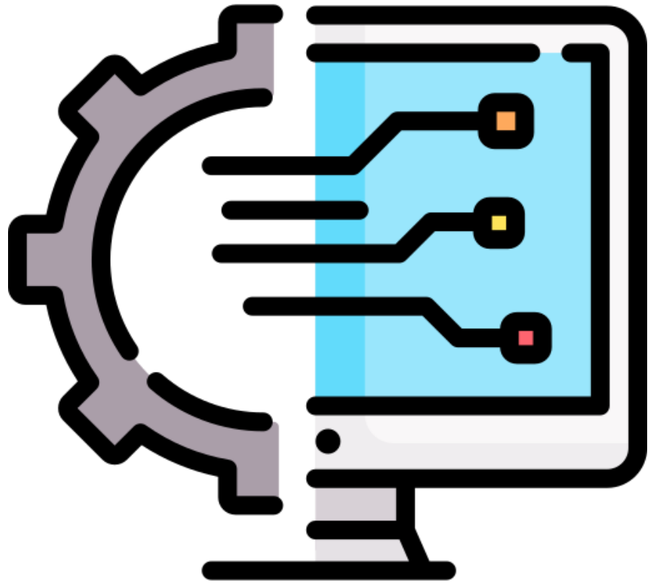
By the end of this lesson, you will be able to:

- Apply Git configuration settings to a new project, including setting up user information and configuring default behaviors
- Create and clone a GitHub repository to understand the fundamentals of repository management
- Execute basic Git commands such as add, commit, push, pull, and status to manage project files
- Differentiate between Git fetch and Git pull for retrieving updates from a remote repository and synchronizing local branches with the latest changes

Configuring Git

Configuring Git

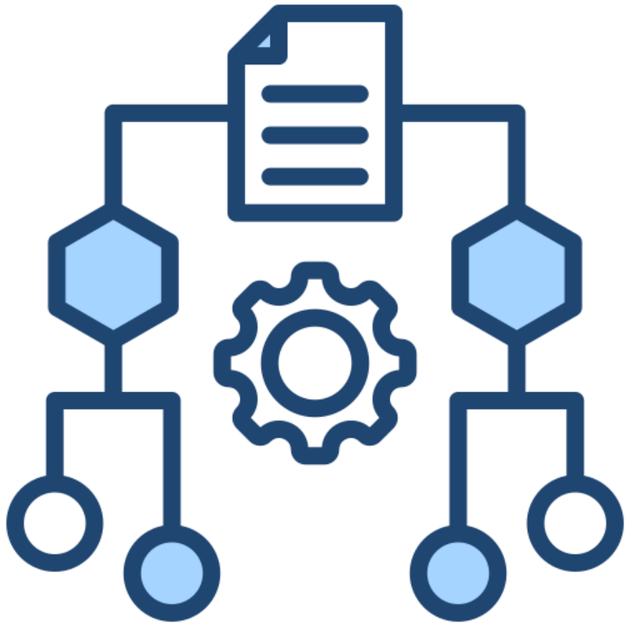
Git configuration refers to the process of setting up various options and preferences for how Git operates on your system and within your repositories.



The Git configuration command is used to set Git configuration values globally or locally within a project.

Configuring Git

The **git config** command accepts parameters to indicate which configuration level to operate.



Syntax:

```
git config [<option>] <key> <value>

Options indicate configuration level that are:
--system
--global
--local
```

The **git config** command writes to the local level, by default, if any configuration level is not specified.

Configuring Git

Global-level configuration settings apply to the current user across all repositories on the system. These settings are stored in the user's home directory, typically in the file **~/.gitconfig**.

These settings are user-specific which provide a customizable and consistent experience for each user, ensuring that their preferences are applied uniformly across all repositories.

Configuring Git

The username can be globally configured so that all commits across all repositories are attributed to the same user.

Syntax:

```
git config --global user.name "Your Name"
```

Example:

```
git config --global user.name
“John”
```

This avoids the need to configure the username for each repository individually.

Configuring Git

The user email ID can be globally configured to ensure it appears in the commit history for all repositories, enhancing traceability and communication.

Syntax:

```
git config --global user.email
"your.email@example.com"
```

Example:

```
git config --global user.name
“John”
```

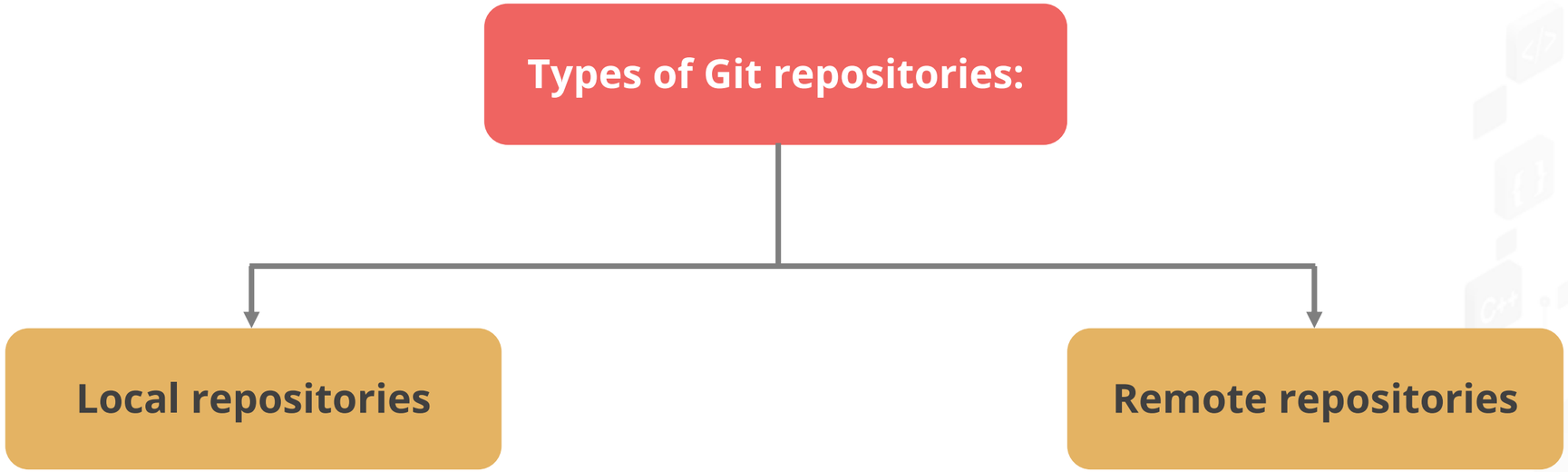
TECHNOLOGY

Creating Repositories in Git

simplilearn

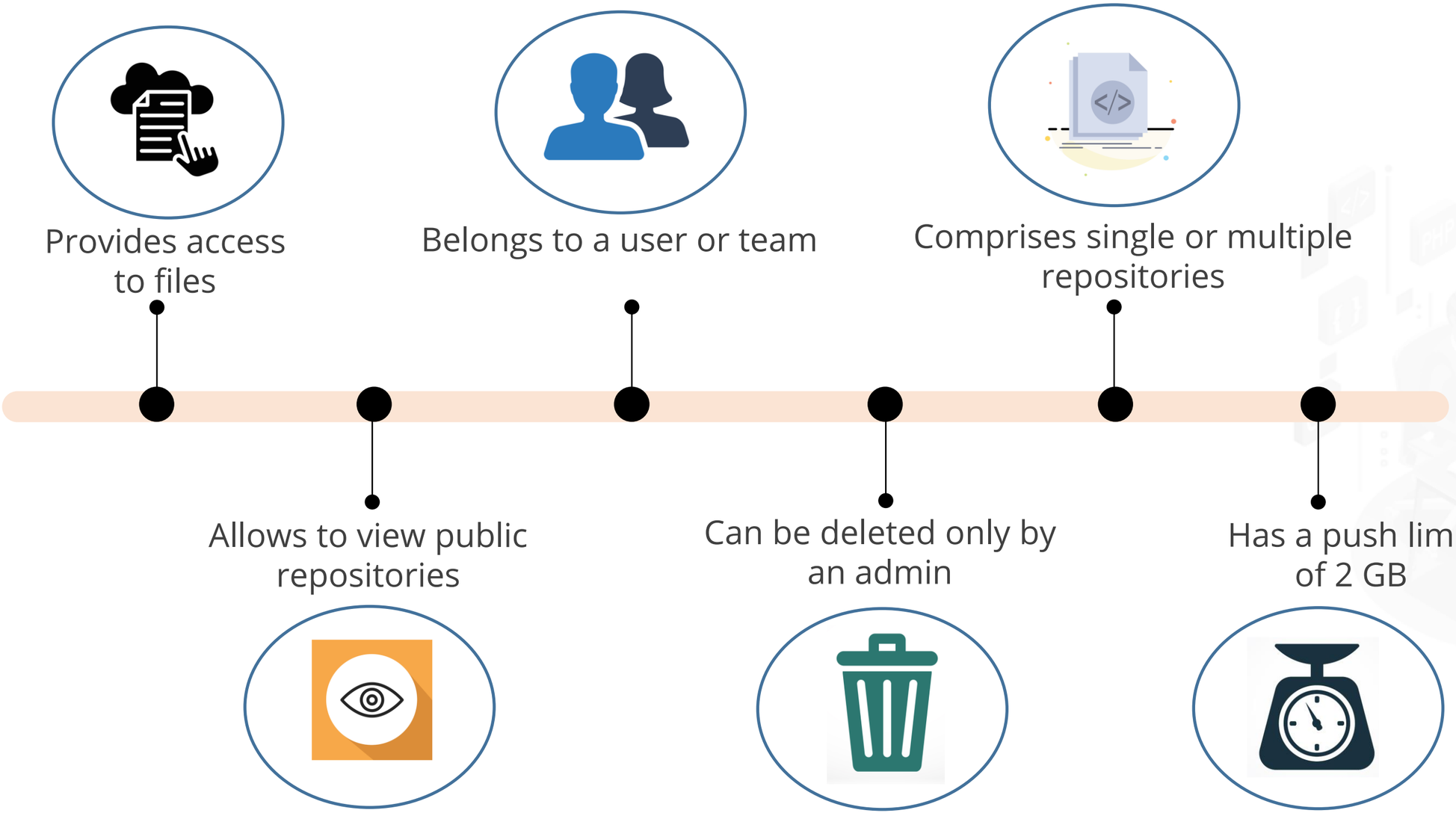
Repositories in Git

A Git repository is a storage space where your project files are stored, and their complete revision history is tracked by Git.



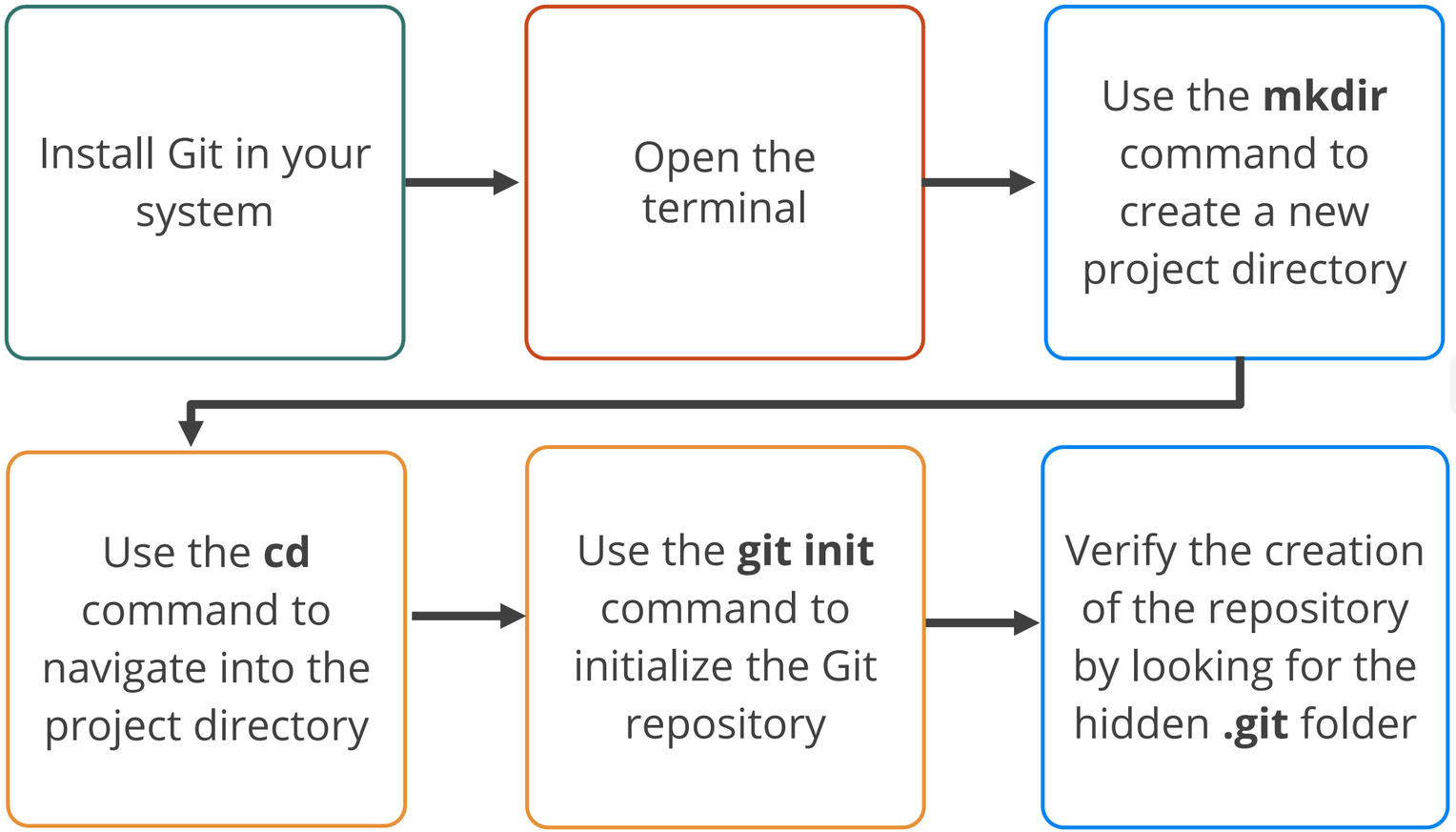
It contains all the data, including commits, branches, tags, and remotes, required to manage the version history of a project.

Characteristics of Git Repository



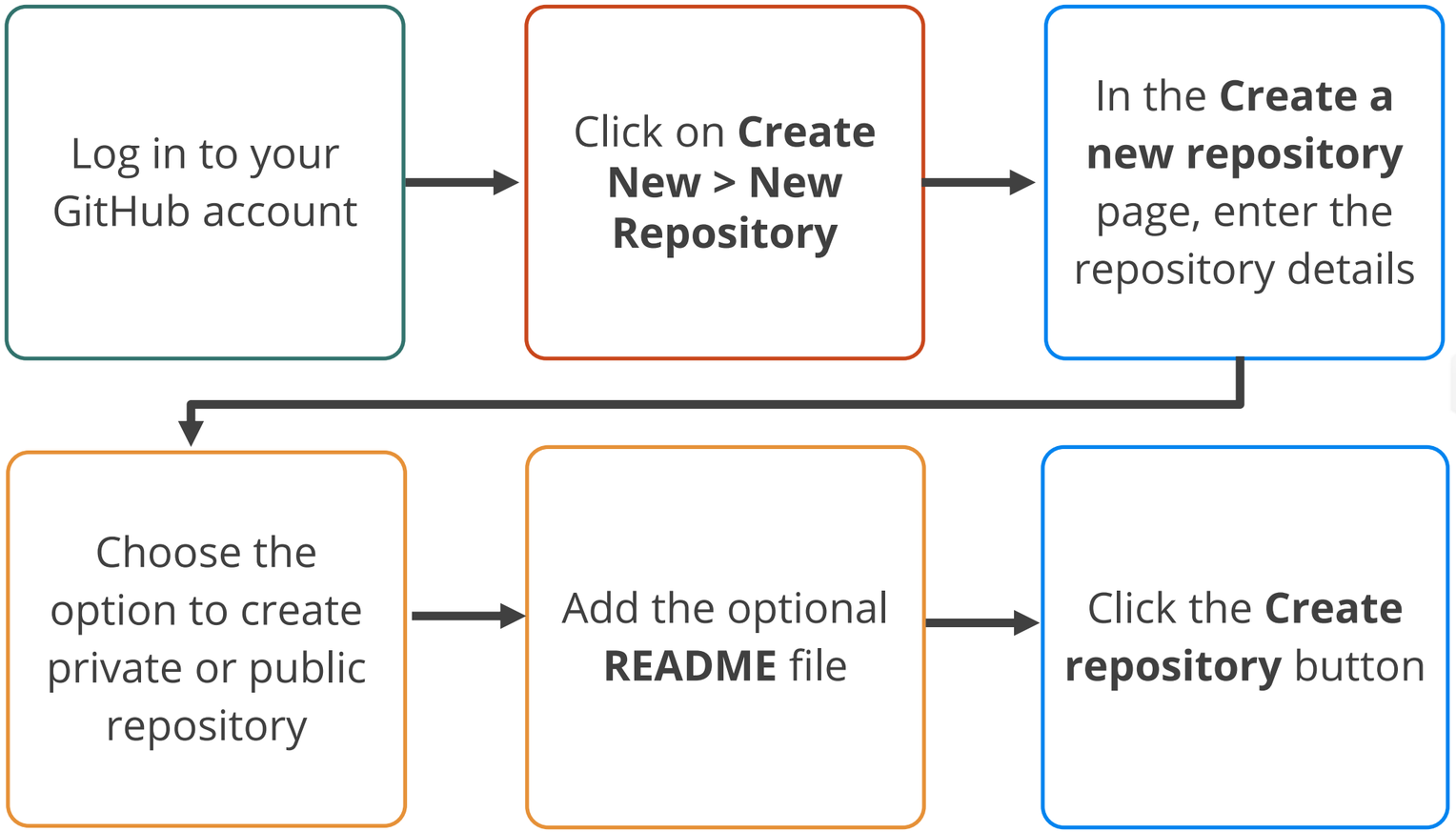
Creating a Local Git Repository

The following are the steps to create a local Git repository:



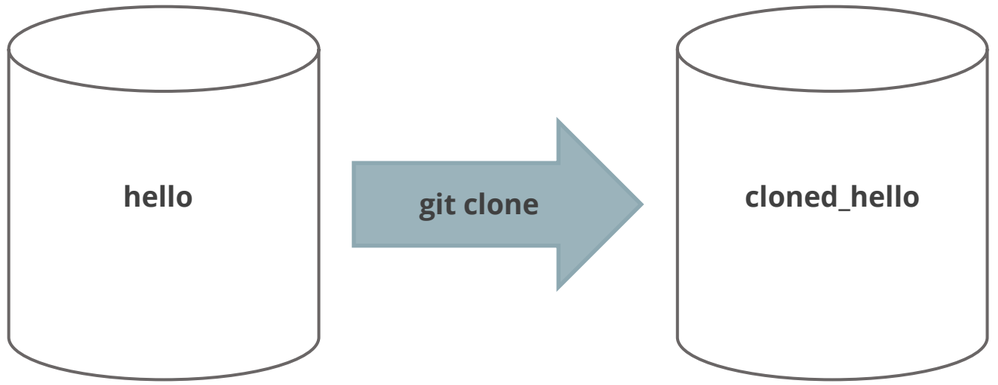
Creating a Remote GitHub Repository

Following are the steps to create a remote GitHub repository:



Cloning a Remote Git Repository

It is the process of creating a copy of an existing repository from a remote server onto the local machine to get a local repository.



Original repository

Copy of original repository

This local repository is not linked with the main remote repository; hence, it is not synchronized with the main repository until done manually.

Creating and Cloning a GitHub Repository

Duration: 10 Min.

Problem Statement:

You have been assigned a task to create and clone a GitHub repository for understanding the fundamentals of repository management and version control using Git and GitHub.

Outcome:

By completing this task, you will be able to create and clone a GitHub repository, gaining an understanding of the fundamentals of repository management and version control using Git and GitHub.

Note: Refer to the demo document for detailed steps:
01_Creating_and_Cloning_a_GitHub_Repository

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Create a new GitHub repository
2. Clone the GitHub repository

Common Git Commands

Common Git Commands

The **git init** command is used to create a new Git repository or convert a folder into a Git repository. It sets up the necessary Git metadata in the directory, enabling version control for the project.

Example:

```
cd /path/to/your/project
git init
```

The execution of this command will create a hidden **.git** directory within the project directory version controlling.

Common Git Commands

The **git clone** command is used to create a copy of an existing remote Git repository.
It downloads the repository and its entire history, creating a local version of the project.

Syntax:

```
git clone <remote_repo_URL>
```

Example:

```
cd /path/to/desired/directory
git clone
https://github.com/username/repository.g
it
cd repository
```

In the above example, the sequence of commands navigates to the desired directory, clones the specified GitHub repository into that directory, and then changes into the newly created repository directory.

Common Git Commands

The **git add** command stage changes in the working directory for the next commit. It adds files to the staging area, preparing them to be included in the next commit.

Syntax:

```
git add [options] <filepattern>...
```

Example:

```
git add file1.txt
```

In the above example, the changes in the file named **file1.txt** are getting staged for the next commit.

Common Git Commands

The **git add** command can be used with the following options:

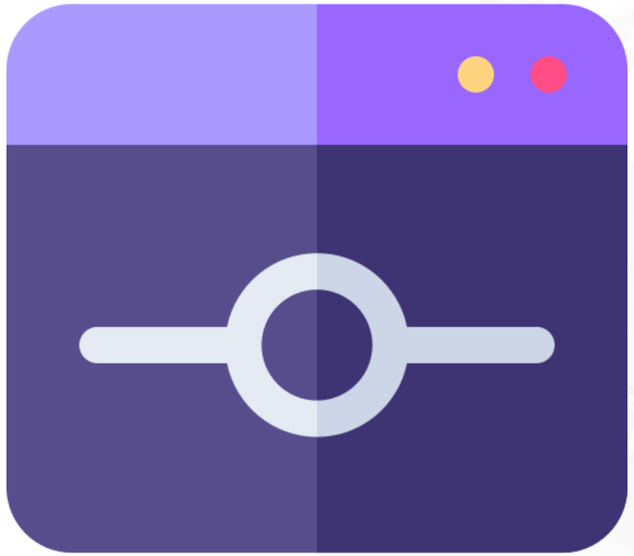
Scenarios	Syntax
Add all changes	git add .
Stage a specific file	git add <filename>
Add a specific directory	git add directory/
Open an interactive prompt to stage changes selectively	git add -i
Stage parts of files	git add -p
Stage changes to tracked files	git add -u
Provide detailed output of the staging process	git add -v

Common Git Commands

The **git commit** command is used to save changes to the local repository. The files are moved from the staging area to a local repository using the **commit** command.

It is always executed after **git add .** command.

It takes a snapshot of the changes made to a Git repository.



Common Git Commands

Different ways of using the Git commit command:

git commit

This command will open the text editor, prompting a commit message.

```
git commit
```

git commit -m <message>

This is a shorthand way of git commit with a message argument.

```
git commit –m “Initial Commit”
```

Common Git Commands

Different ways of using the Git commit command:

git commit --amend

This command will modify the last commit instead of creating a new commit; staged changes will be added to the previous or last commit.

```
git commit –amend –m “New Changes”
```

Common Git Commands

The **git remote** command helps manage the set of remote repositories that are being tracked with the local repository.

These remote repositories are typically located on a server or another machine, allowing you to collaborate with others.

Common Git Commands

The **git remote add** command helps to add a new remote to the local repository.

Syntax:

```
git remote add <remote-name> <repo-url>
```

Example:

```
git remote add origin
https://github.com/<username>/<repo-name>.git
```

Common Git Commands

The **git remote remove** command will remove an existing remote repository from the local repository.

Syntax:

```
git remote remove <repository_name>
```

Example:

```
git remote remove origin
```

Common Git Commands

The **git remote set-url** command is used to change the URL of an existing remote repository in your local Git configuration.

Syntax:

```
git remote set-url <name> <new-url>
```

Example:

```
git remote set-url origin
https://github.com/newuser/newrepo.git
```

Common Git Commands

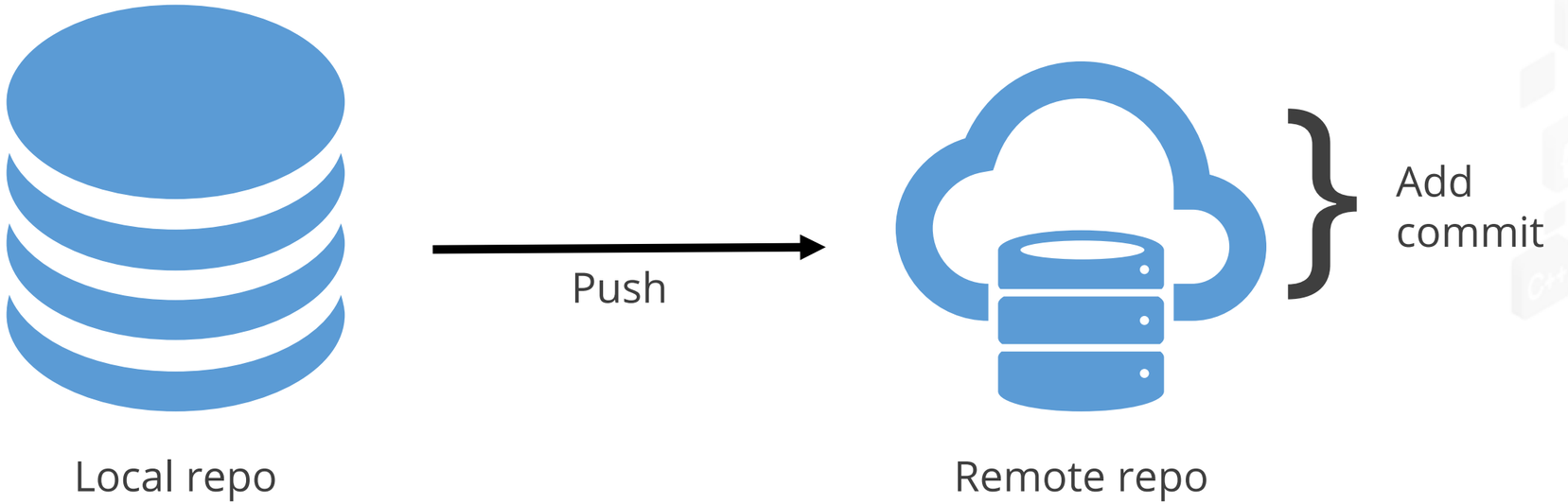
The **git status** command displays the state of the working directory and the staging area.
It shows which changes have been staged, which have not, and which files are not being tracked by Git.

Example output of executing git status command:

```
no changes added to commit (use git add and/or git commit –a)

vacas: ~/learnLua (waqas) $ git status

On branch waqas
Changes not staged for commit:
      (use “git add <file>...” to update what will be committed)
      (use “git checkout -- <file>...” to discard changes in working directory)

                                    modified:    src/garbage.lua
Untracked  files:
      (use “git add <file>...” to include in what will be committed)

            test/
            test1/

no changes added to commit (use “git add” and/or “git commit –a”)
```

Common Git Commands

Pushing in Git refers to the process of copying material from a local repository to a remote repository.



Common Git Commands

The **git push** command is used to upload local repository content to a remote repository.

Syntax

```
git push -u <remote—name> <remote-branch>
```

Example

```
git push -u origin master
```


It transfers commits from your local repository to a remote repository, updating the remote repository with your changes.

Common Git Commands

The **git pull** command fetches changes from a remote repository and merges them into your current branch.

Syntax

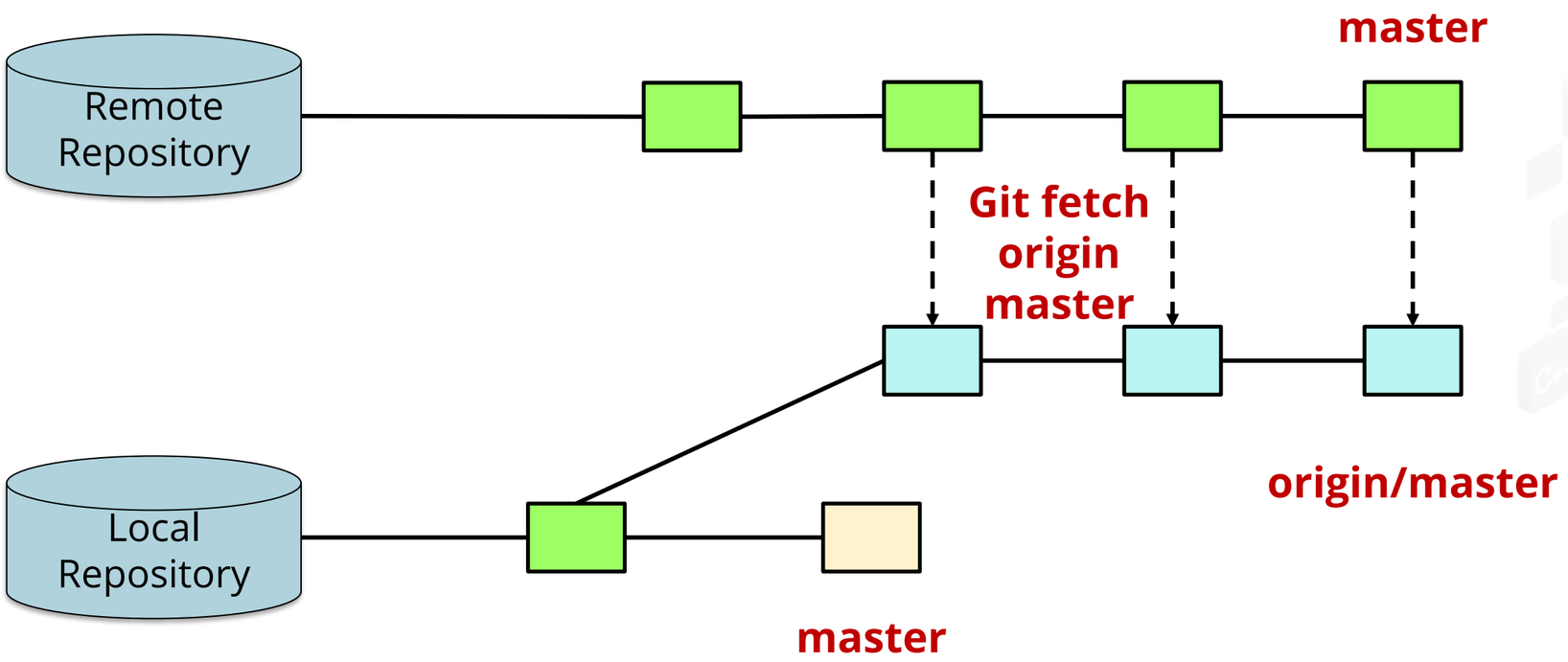
```
git pull <remote> <branch>
```

Example

```
git pull origin main
```

Common Git Commands

The **git fetch** command downloads commits, files, and references from a remote repository into your local repository.



It updates your remote-tracking branches with the latest changes from the remote repository without merging them into your current branch.

Common Git Commands

git fetch	git pull
Downloads commits, files, and refs from a remote repository	Downloads updates from a remote repository and merges them into your current branch
Does not merge changes into your current branch	Merges fetched changes into the current branch automatically
Used to review changes before merging	Used to update the local branch with the latest changes from the remote branch
Conflicts can be resolved manually after reviewing changes	Conflicts must be resolved immediately after the pull
Allows inspection of fetched changes before merging	Merges changes without separate inspection step

Executing Basic Git Commands

Duration: 10 Min.

Problem Statement:

You have been assigned a task to demonstrate the execution of basic Git commands for synchronizing code or files with GitHub, covering repository initialization, adding files, and committing changes.

Outcome:

By completing this task, you will be able to demonstrate the execution of basic Git commands for synchronizing code or files with GitHub, including repository initialization, adding files, and committing changes.

Note: Refer to the demo document for detailed steps:
02_Executing_Basic_Git_Commands

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Execute Git commands

Pushing Code to GitHub

Duration: 10 Min.

Problem Statement:

You have been assigned a task to showcase the process of pushing local code changes to a GitHub repository, ensuring version control and collaboration.

Outcome:

By completing this task, you will be able to showcase the process of pushing local code changes to a GitHub repository, ensuring effective version control and collaboration.

Note: Refer to the demo document for detailed steps:
03_Pushing_Code_to_GitHub

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed:

1. Push the code to GitHub

Key Takeaways

- Git configuration refers to the process of setting up various options and preferences for how Git operates on your system and within your repositories.
- The global level Git configurations are stored in the user's home directory, typically in the file **~/.gitconfig**.
- A Git repository is a storage space where your project's files and their complete revision history are tracked by Git.
- Cloning is the process of creating a copy of an existing repository from a remote server onto the local machine.
- The **git init** command is used to create a new Git repository or convert a folder into a Git repository.

Key Takeaways

- The **git add** command stage changes in the working directory for the next commit.
- The **git commit** command is used to save changes to the local repository.
- The **git remote** command helps manage the set of remote repositories that are being tracked with the local repository.
- The **git status** command displays the state of the working directory and the staging area.
- The **git pull** command fetches changes from a remote repository and merges them into your current branch.
- The **git push** command is used to upload local repository content to a remote repository.

TECHNOLOGY

Thank You

simplilearn