SDEV 1001

Programming Fundamentals

Introduction to Programming - 2

A LEADING POLYTECHNIC COMMITTED TO YOUR SUCCESS

Expectations - What I expect from you

- No Late Assignments

- No Cheating

- Be a good classmate

- Don’t waste your time

- Show up to class

Agenda

On the right is what we will cover today. Note: this is a slides version of the example that we're going to use in class.

What is Version Control?
Why is Version Control Important?
Useful Git Command Reference
Step 1: Check if Git is Installed
Step 2: Configure Git
Step 3: Initialize a Repository
Step 4: Add and Commit Files
Step 5: Make and Track Changes
Step 6: Connect to a Remote Repository
Step 7: Verify Changes on GitHub

What is Version Control?

Version control is a system that records changes to files over time.

It allows you to recall specific versions later, making it easier to track changes, collaborate with others, and manage project history.

Why is Version Control Important?

Version control enables multiple people to work on a project at the same time.
It allows us to keep track of changes, go back to previous versions, and see who made what changes and when.

Useful Git Command Reference

- `git --version` : Check your git version
- `git init` : Initialize a repository
- `git status` : See the current status of your repo
- `git add <file>` : Add files to the staging area
- `git commit -m "message"` : Commit staged changes
- `git log --graph` : View commit history as a graph

Step 1: Check if Git is Installed

Open a terminal and type:

```
git --version
```

If git is not installed, download it from GitHub Desktop or Git Guides.

Step 2: Configure Git

Set your name and email for commits:

```
git config --global user.name "Your Name"
git config --global user.email "your@email.com"
```

Check your configuration with:

```
git config --global --list
```

Step 3: Initialize a Repository

Navigate to your project folder and run:

```
git init
```

Check the status with:

```
git status
```

You should see untracked files listed.

Step 4: Add and Commit Files

Add all files to the staging area:

```
git add .
```

Commit your changes:

```
git commit -m "Initial commit"
```

Check the commit log with:

```
git log --graph
```

Step 5: Make and Track Changes

Edit your file (e.g., add a company name), then check status:

```
git status
```

See the difference with:

```
git diff
```

Add and commit your changes as before.

Step 6: Connect to a Remote Repository

Create a new repository on GitHub. Add the remote URL to your local repo:

```
git remote add origin <remote-url>
git remote -v
```

Push your changes and set the upstream branch:

```
git push --set-upstream origin master
```

Step 7: Verify Changes on GitHub

Go to your GitHub repository in the browser. Check that your files and commits are present. Click the "commits" button to view the commit log.

Example

Let’s do a full example together using this knowledge.