## Creating and running the "Hello, World!" Program

First, a Python program that prints "Hello, World!" was created in PyCharm. The code is a single line:

print("Hello, World!")

This file was saved as

**helloworld.py**.

## Understanding the Git Error

After creating the file, the user went to the Git terminal and entered the

git status command. The output was a fatal error:

$ git status fatal: not a git repository (or any of the parent directories): .git

This error occurred because the

git init command, which is necessary to initialize a Git repository, was not run beforehand. The git status command can't function without a repository to check.

A screenshot of a computer

AI-generated content may be incorrect.

**What is a Git Working Directory?**

The **working directory** (or working tree) is the folder on your system where you have checked out and are actively working on files. It contains the actual files and subfolders that you can modify, add, or delete during development.

In this case, the working directory contains the **helloworld.py** file. The working directory is separate from the “.git” hidden directory, which holds all of Git's repository information.

**Initializing the Git Repository**

The first step is to fix the "fatal: not a git repository" error. To do this, you need to initialize a new Git repository in your working directory. You would use the

git init command in the Git terminal. This command creates a hidden .git folder, which is where Git stores all the information about your project's history.

A screen shot of a computer

AI-generated content may be incorrect.

**Adding and Committing the File**

Once the repository is initialized, the

helloworld.py file will be an untracked file in your working directory.

A screen shot of a computer

AI-generated content may be incorrect.

The next steps are:

A computer screen with a black screen and a black screen with text

AI-generated content may be incorrect.

1. **Add the file to the staging area**: Use the command git add helloworld.py. The staging area is a middle ground where you prepare files to be committed.

A computer screen with black and green text

AI-generated content may be incorrect.

helloworld.py is now moved to stagging directory from working directory.

To move all files and folders at one shot into tracked files, we need to use “**git add .**” command.

1. **Commit the file**: Use the command git commit -m "Initial commit of helloworld.py". This command permanently saves the staged file to the Git repository's history. The message in quotes describes the changes you've made.

A screen shot of a computer

AI-generated content may be incorrect.

**Note**: we have added hellowworld.py only into our local git repository.

If we want to add all the files and folders to git repository we need to first use git add . command first and then we may proceed to commit all files and folders.

After these steps, you can use git status again, and it will show that your working directory is clean, as the file has been successfully committed to the repository.

A computer screen shot of a program

AI-generated content may be incorrect.

**Understanding Line Ending Warnings which might seen during second commit**

The most common warning you might see is: warning: LF will be replaced by CRLF or vice-versa.

* **CRLF (Carriage Return Line Feed):** This is the standard line ending used by Windows.
* **LF (Line Feed):** This is the standard for macOS and Linux.

Git tries to handle these differences automatically, but sometimes it will issue a warning to let you know it's converting the line endings to maintain consistency across the repository. This warning is not a critical error and your commit will still go through. However, it can cause problems if your team is working on different operating systems and you don't have a consistent configuration.