What is a Portfolio Project?

This project is a little different from other Codecademy projects you've encountered. In this project, you will bring together what you have learned in previous lessons to build a project in your own development environment and publish it to the web!

Create your own files outside of the Codecademy platform

Write your own code with less guided instructions

Use common project management processes to track your progress

Welcome to the portfolio project in CS 101: Introduction to Programming!

In this portfolio project, you will research, brainstorm, and build a basic terminal program of your choice for your friends and family to play with. After you finish building the program, you will create a blog post to share the program on a publication of your choice!

Project Ideas:

- Blackjack
- Tic-Tac-Toe
- Connect Four
- Battleship
- Minesweeper
- Who Wants to Be a Millionaire?
- Tarot Reading
- NYC MetroCard Price Calculator
- Mortgage Calculator
- Today's Horoscope
- Your Very Own Text-Based Adventure

Project Objectives:

- Build a terminal program using Python
- Add at least one interactive feature using `input()`
- Use Git version control
- Use the command line and file navigation
- Write a technical blog post on the project

Prerequisites:

- Python
- Git/GitHub
- Command Line and File Navigation

Think of an Idea

Take some time to think of an idea for your project.
💡

Some questions you can ask yourself is:

- Who do you want to build something for? Yourself? Your friends? Your family? Your co-workers?
- What are your/their hobbies?
- Do you want to build something more fun or something more useful?

Hint

If you are having trouble thinking of one, take a look at the example projects!

Project Brainstorming

Now that you have an idea, visualize the end result:

- What does your program do?
- How will it work in a terminal?
- Is it one player or two players?

Make sure that it satisfies all of the project objectives.

Make a timeline for yourself and avoid the temptation to build things that aren't required. Setting firm boundaries and deadlines will keep you on track and prevent scope creep.

The following tasks will help you identify natural breakpoints.

Hint

Properly scoping your project will greatly benefit you; scoping creates structure while requiring you to think through your entire project before you begin. You should start with stating the goals for your project, then gathering the data, and considering the analytical steps required. A proper project scope can be a great road map for your project, but keep in mind that some down-stream tasks may become dead ends which will require adjustment to the scope.

Setting up your GitHub Repository

Create a new GitHub repository for this project and add a Python file inside.

Don't forget to name your file!

Hint

Your Python program should have a name, something like **name.py**.

If you need more guidance, review the:

- GitHub Desktop article
- Git Cheatsheet

Version Control

Set up Git tracking in your directory and make sure to add and commit changes as you make them.

Hint

Set up Git tracking with

```
git init PROJECTNAME
```

Stage changes with:

```
git add FILENAME
```

Commit changes with:

```
git commit -m "MESSAGE"
```

Write Your Program

This is the bulk portion of your project.

Start with creating the variables that you need and then text prompts that will let the user understand what the program is.

Hint

Constantly run the program every few lines to make sure you knock out the bugs as you go.

Refactoring

Now that you have a working program, try to go back to it see what you can do better?

- Can you add some functions to make the code more byte-sized?
- Can you add a class & object somewhere?

Blog Post

Once you've completed the program, you're ready to create your blog post.

Hint

Create a blog post using Medium, Dev.to, or some other blogging platform.

Your blog post should include the following:

- A compelling title about your program
- An introduction sharing the background info (the "why")
- An image or a GIF of your program
- An accompanying paragraph describing your Python code
- A link to your code on GitHub.
- A conclusion