CS50's Introduction to Programming with Python

OpenCourseWare

Donate (https://cs50.harvard.edu/donate)

David J. Malan (https://cs.harvard.edu/malan/)
malan@harvard.edu
(https://www.facebook.com/dmalan) (https://github.com/dmalan)
(https://www.instagram.com/davidjmalan/) (https://www.linkedin.com/in/malan/)
(https://www.reddit.com/user/davidjmalan)
(https://www.threads.net/@davidjmalan) (https://twitter.com/davidjmalan)

Final Project

Once you have solved each of the course's problem sets, it's time to implement your final project, a Python program of your very own! The design and implementation of your project is entirely up to you, albeit subject to these requirements:

- Your project must be implemented in Python.
- Your project must have a `main` function and three or more additional functions. At least three of those additional functions must be accompanied by tests that can be executed with `pytest`.
  - Your `main` function must be in a file called `project.py`, which should be in the "root" (i.e., top-level folder) of your project.
  - Your 3 required custom functions other than `main` must also be in `project.py` and defined at the same indentation level as `main` (i.e., not nested under any classes or functions).
  - Your test functions must be in a file called `test_project.py`, which should also be in the "root" of your project. Be sure they have the same name as your custom functions, prepended with `test_` (`test_custom_function`, for example, where `custom_function` is a function you've implemented in `project.py`).
  - You are welcome to implement additional classes and functions as you see fit beyond the minimum requirement.
- Implementing your project should entail more time and effort than is required by each of the course's problem sets.

- Any `pip`-installable libraries that your project requires must be listed, one per line, in a file called `requirements.txt` in the root of your project.

▶ **Example Project Structures**

You are welcome, but not required, to collaborate with one or two classmates on your project. (You might want to collaborate with Live Share (https://code.visualstudio.com/learn/collaboration/live-share)!) But a two- or three-person should entail twice or thrice the time and effort required by a one-person project.

Note that CS50's staff audits submissions to CS50P including this final project. Students found to be in violation of **the Academic Honesty policy** will be removed from the course and deemed ineligible for a certificate. Students who have already completed CS50P, if found to be in violation, will have their CS50 Certificate (and edX Certificate, if applicable) revoked.

When to Do It

By Wednesday, January 1, 2025 at 12:59 PM GMT+8 (https://time.cs50.io/20241231T235900-0500).

Getting Started

Creating an entire project may seem daunting. Here are some questions that you should think about as you start:

- What will your software do? What features will it have? How will it be executed?
- What new skills will you need to acquire? What topics will you need to research?
- If working with one or two classmates, who will do what?
- In the world of software, most everything takes longer to implement than you expect. And so it's not uncommon to accomplish less in a fixed amount of time than you hope. What might you consider to be a good outcome for your project? A better outcome? The best outcome?

Consider making goal milestones to keep you on track.

How to Submit

You must complete all three steps!

Step 1 of 3

Create a short video (that's no more than 3 minutes in length) in which you present your project to the world, as with slides, screenshots, voiceover, and/or live action. Your video should somehow include your project's title, your name, your city and country, and any other details that you'd like to convey to viewers. See howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen (https://www.howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen/) for tips on how to make a "screencast," though you're welcome to use an actual camera. Upload your video to YouTube (or, if blocked in your country, a similar site) and take note of its URL; it's fine to flag it as "unlisted," but don't flag it as "private."

Submit this form (https://forms.cs50.io/5e2dd8e8-3c8b-4eb2-b77d-085836253f26)!

Step 2 of 3

Create a `README.md` text file (named exactly that!) in your `~/project` folder that explains your project. This file should include your Project title, the URL of your video (created in step 1 above) and a description of your project. You may use the below as a template.

```
# YOUR PROJECT TITLE
#### Video Demo:  <URL HERE>
#### Description:
TODO
```

If unfamiliar with Markdown syntax, you might find GitHub's Basic Writing and Formatting Syntax (https://docs.github.com/en/free-pro-team@latest/github/writing-on-github/basic-writing-and-formatting-syntax) helpful. If you are using the CS50 Codespace and are prompted to "Open in CS50 Lab", you can simply press `cancel` to open in the Editor. You can also preview your `.md` file by clicking the 'preview' icon as explained here: Markdown Preview in vscode (https://code.visualstudio.com/docs/languages/markdown#_markdown-preview). Standard software project `README`s can often run into the thousands or tens of thousands of words in length; yours need not be that long, but should at least be several hundred words that describe things in detail!

> Your `README.md` file should be minimally multiple paragraphs in length, and should explain what your project is, what each of the files you wrote for the project contains and does, and if you debated certain design choices, explaining why you made them. Ensure you allocate sufficient time and energy to writing a `README.md` that documents your project thoroughly. Be proud of it! If it is too short, the system will reject it.

Execute the `submit50` command below from within your `~/project` directory (or from whichever directory contains `README.md` file and your project's code, which must also be submitted). If your project does not meet all the requirements above, it may be rejected, so be

sure you have satisfied all of the bullet points atop this specification and written a thorough `README`:

```
submit50 cs50/problems/2022/python/project
```

▼ **Trouble Submitting?**

If you encounter issues because your project is too large, try to ZIP all of the contents of that directory (except for `README.md`) and then submit that instead. If still too large, try removing certain configuration files, reducing the size of your submission below 100MB, or try to upload directly using GitHub's web interface (https://docs.github.com/en/free-pro-team@latest/github/managing-files-in-a-repository/adding-a-file-to-a-repository) by visiting github.com/me50/USERNAME (https://github.com/me50/USERNAME) (where `USERNAME` is your own GitHub username) and manually dragging and dropping folders, ensuring that when uploading you are doing so to your `cs50/problems/2022/python/project` branch, otherwise the system will not be able to check it!

Step 3 of 3

That's it! Your project should be graded within a few minutes. Be sure to visit your gradebook at cs50.me/cs50p (https://cs50.me/cs50p) a few minutes after you submit. It's only by loading your Gradebook that the system can check to see whether you have completed the course, and that is also what triggers the (instant) generation of your free CS50 Certificate and the (within 30 days) generation of the Verified Certificate from edX, if you've completed all of the other assignments.

This was CS50P!