ISC 2024 CS Lab Guide and Description

Lab Description

Students in the lab will be using the programming language Python to create a **TV Show information system** that supplies the user with information about their requested input TV show title. The given information should include the show's:

- Title
- Genre
- Published Date
- Brief Description
- Number of Seasons

Beginner students will be provided with a **predefined dictionary with keys** of TV shows and values of categories which they will manipulate to get the outlined output.

Intermediate students will use a **public API** to learn and practice API skills and be able to create their own dictionary/organization system for the information to be retrieved by the user.

Template files: https://github.com/ISC-Edmonton/2024

The first hour will be reserved for a lecture on the concepts that students will need to complete the lab. This should be a review if the students read the background package. If that is the case, they may begin planning for the lab early. The remaining two hours will be open for the students to work on their lab and seek help from TA's and directors if needed.

Lab Instructions & Marking Guide

Download the respective folders containing the lab files which you will be using to create your program. The lab will be broken down into parts with increasing difficulty. Even if you are unable to complete **all** the parts, you will get marks for implementing each task listed under the parts.

Watch the expected output videos for your labs to see what your end product should look like. It's important that you match the format of the expected output – partial marks will be given when applicable.

- Mark Breakdowns for the labs are provided in each section's rubric. Make sure to look through them to ensure you don't lose marks for missing things!

Bonus:

- The beginner section has a bonus part with tasks listed underneath. These bonus tasks will earn you a maximum of **6 bonus points** to add to your total CS section marks (lab + question package)
    - The bonus task in the lab will not be included in the video for the expected output – you are given the freedom to implement and format this part in whatever way you'd like!

You MUST submit your lab as a **.zip file**. Locate the instructions for downloading the zip file under **What is Repl.it?** and upload your submissions to the Google Form: **Upload Link**

Tips & Bonus Points

⇒ Comments! Write comments (block and line comments) in your code to score some free marks
⇒ Implement error handling for malformed queries (e.g. typos, incorrect parameters, lowercase/uppercase etc…)
⇒ We will accept pseudocode for part marks if you can't figure out how to finish the code

Beginner

Instructions

In the provided file, ISC24_Beginner.py you've been provided with a template code to help you get started. Look over the program and understand what has been written for you. You will find a nested dictionary containing the TV Shows' information for you to use to complete the lab. Watch the expected output video to see what your program must look like for full marks.

Your final program must include the following:

Part I

- A printed "menu" of the TV Show titles the user can enter to find out information of
- An input prompt which asks the user to enter a TV Show of their choice from the provided menu
- A second prompt which asks the user what specific information from the show they would like to find:
    - Genre
    - Published Date
    - Brief Description
    - Number of Seasons
- An output which tells the user its requested information in the following format: "Here is the result for *[TV Show Name] - [Information Requested]*: [*Information*]"

Part II

- Add another feature which allows users to search by genre
    - This will list the titles with the matching specified genre
    - Note: **You do not** need to include Search by 'Title', 'Brief Description', 'Published Date', 'Number of Seasons' or 'All'
- Your input prompt will specify whether the user wishes to search for a specific TV Show's information (implemented in Part I) or to list the titles from a specific genre
- An output which tells the user the requested information

Bonus (3 Marks Each, total 6 Bonus Marks)

- **Add an 'All' Option:** In the first part, implement an option for 'All' which displays all four pieces of information: Genre, Published Date, Brief Description, Number of Seasons
- **Add Sorting Options:** Enhance the program to allow the user to sort the list of TV shows based on certain criteria such as title, genre, or number of seasons. After selecting a genre or choosing 'All', provide an option for the user to specify how they want the results to be sorted.

Expected Output

Part 1 - Demo

Part 2 - Demo

CLICK HERE FOR BEGINNER RUBRIC

Intermediate

This lab is reconstructed from a CMPUT 174 Lab

Instructions

In the provided file, ISC24_Intermediate.py you've been provided with a template code including some function definitions to help you get started. Look over the program and understand what has been written for you.

You will be using TV Maze, a REST API, to fetch the data for your TV Show Information program! Briefly read over and familiarize yourself with the API by going through its documentation. You will specifically need the following end points:

- Show search
- Show seasons
- Episode main information

The program uses the API to search for a show with a title matching the user's input. Then, it allows the user to select the specific show, displays all of the show's seasons, allows the user to select a specific season, and lists the episodes in that season. Watch the expected output video to see what your program must look like for full marks. For Part III, you are to implement the options with your own creative decision making (ie. no strict display rules / output video), so try to show us what you can make!

Your final program must include the following:

Part I

- **Title Search:** A prompt which asks the user to input a title they wish to search for
- **Show Display:** A printed result of shows from the API, formatted to the expected output
- **Show Selection:** A prompt which asks the user to select a show from the list of shows given (since there may be more than one show with that title)

Part II

- **Season Display:** After selecting a show, all of its seasons should be printed in the correct format
- **Season Selection:** A prompt which asks the user to select a season from the previously printed list of seasons
- **Episode Display:** A printed result of all the episodes in the selected season including the episode's title, number, and rating

Part III

- **Show the Show's Main Info:** Implement an option which displays the show's main information using the endpoint, "Show main information"
- **Search by Genre:** Enhance the program to allow the user to search for TV shows based on a specific genre. Modify the prompt to ask the user if they want to search by title or

genre. If the user chooses to search by genre, prompt them to input the genre and then display a list of shows matching that genre.

- **Errors Handling:** Implement error handling in the program to handle cases where the user input doesn't match any shows in the API or if there are any network-related errors during API requests. Provide clear error messages and give the user the option to retry or exit gracefully.

Bonus (2 Marks Each, total 6 Bonus Marks)

- Episode Synopsis: Implement an option to display a brief synopsis of each episode alongside its title, number, and rating when listing episodes for a selected season. Ensure the synopsis is concise and relevant to enhance the user's understanding of each episode.
- Customized Episode Sorting: Enhance the program to allow the user to specify how episodes should be sorted within a season. Provide options such as sorting by episode number, title, or rating. Ensure that the selected sorting method is accurately applied when displaying episodes.
- User Profiles and Favorites: Implement user profiles that allow users to save their favorite shows and episodes. Provide options to create, edit, and delete profiles, as well as add or remove shows from favorites. Ensure that user profiles persist between sessions to enhance user experience.

Expected Output

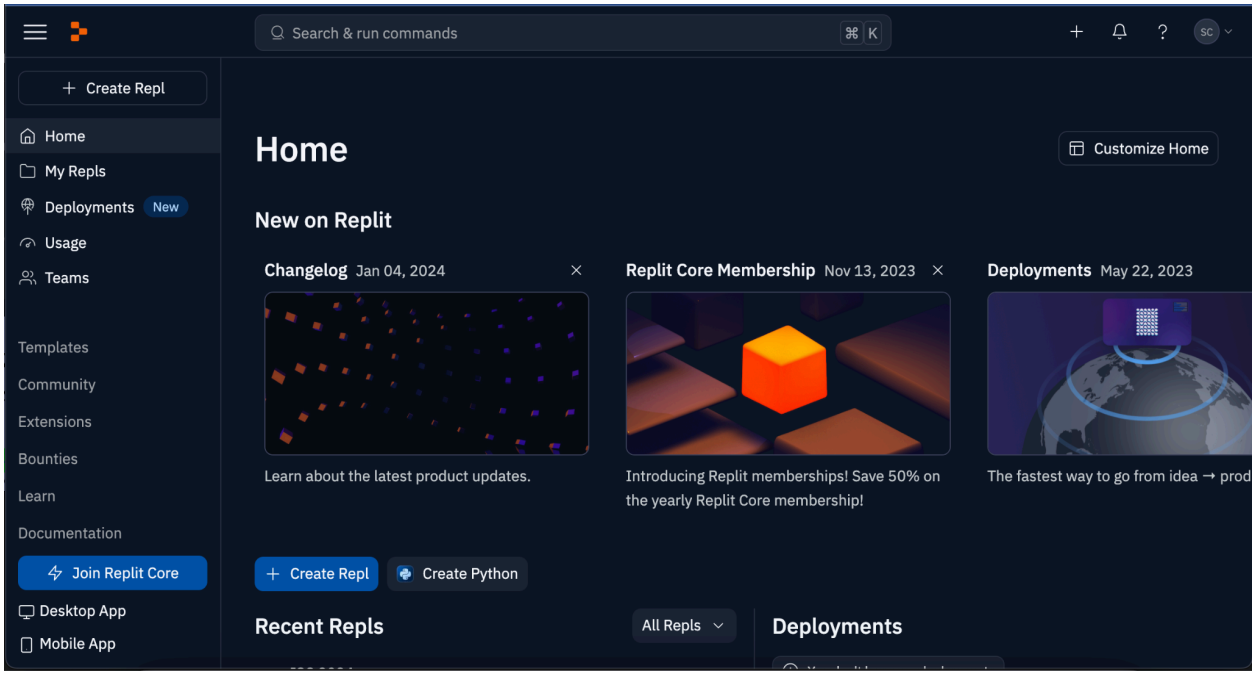
Part 1 & 2 Demo

CLICK HERE FOR INTERMEDIATE RUBRIC

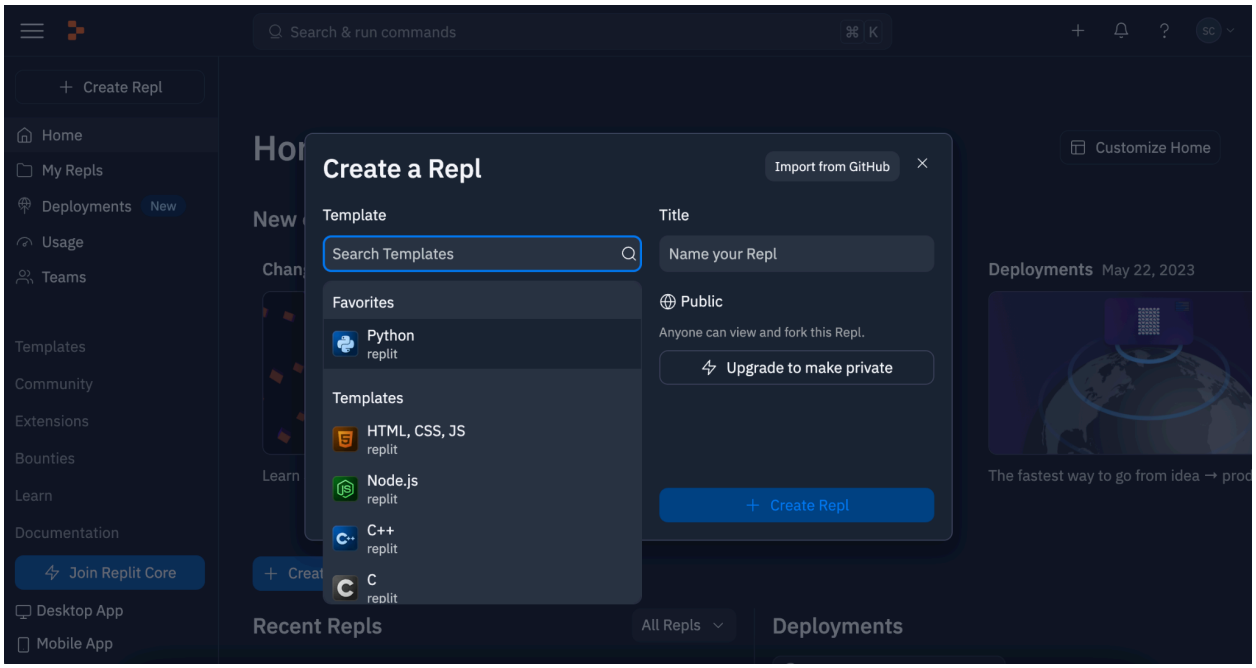
What is Repl.it?

To make our project today and be able to collaborate in a team, we will be using an online tool called Repl.it to design our program. Here is a quick overview of Repl.it:

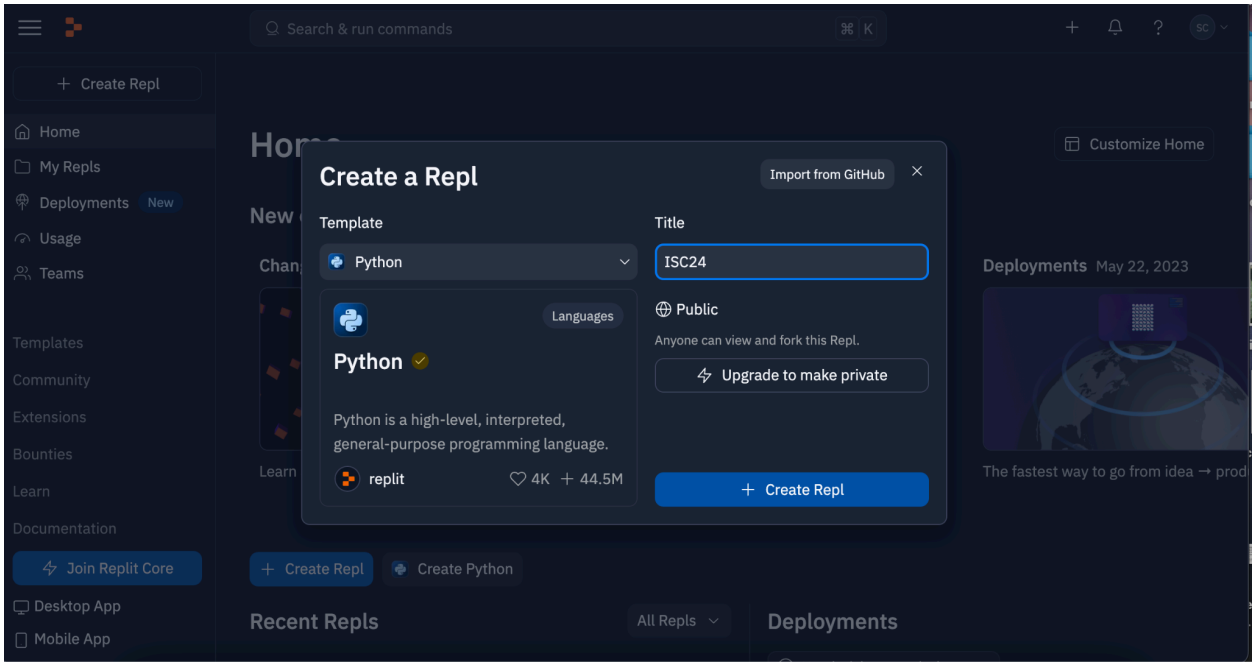
Click Here for a Video on Repl.it
Click Here to Sign up for Repl.it



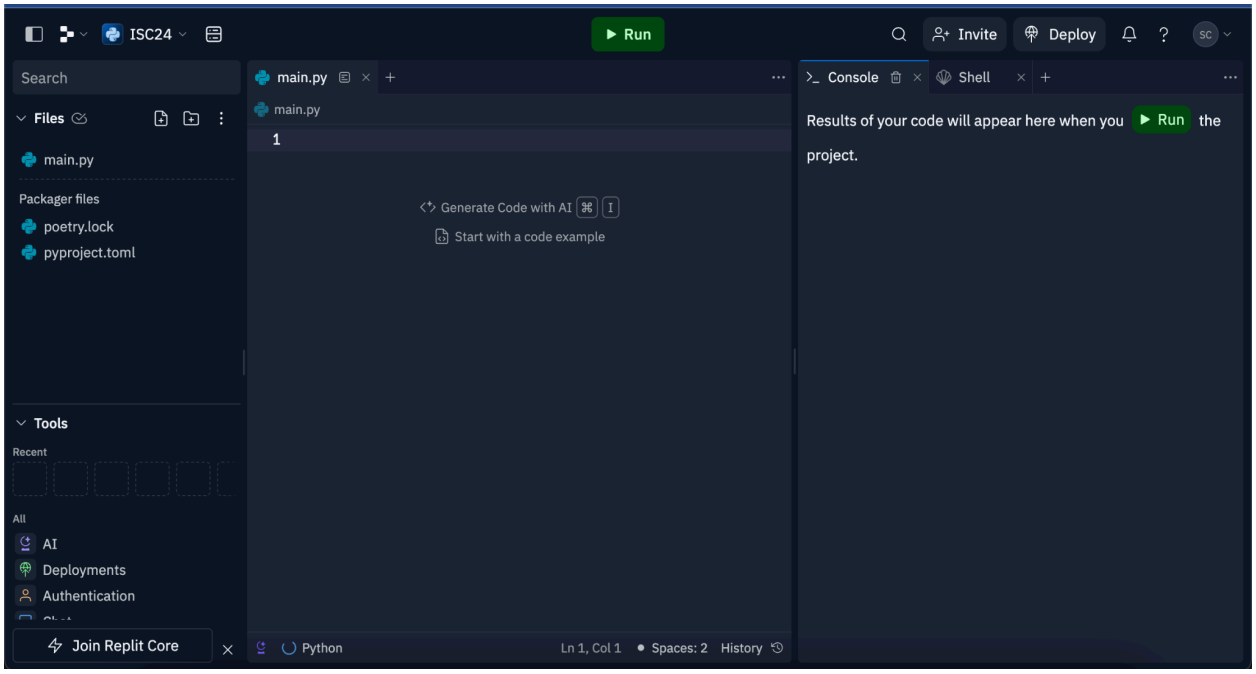
After signing up, you should be able to access your home page which looks like this.



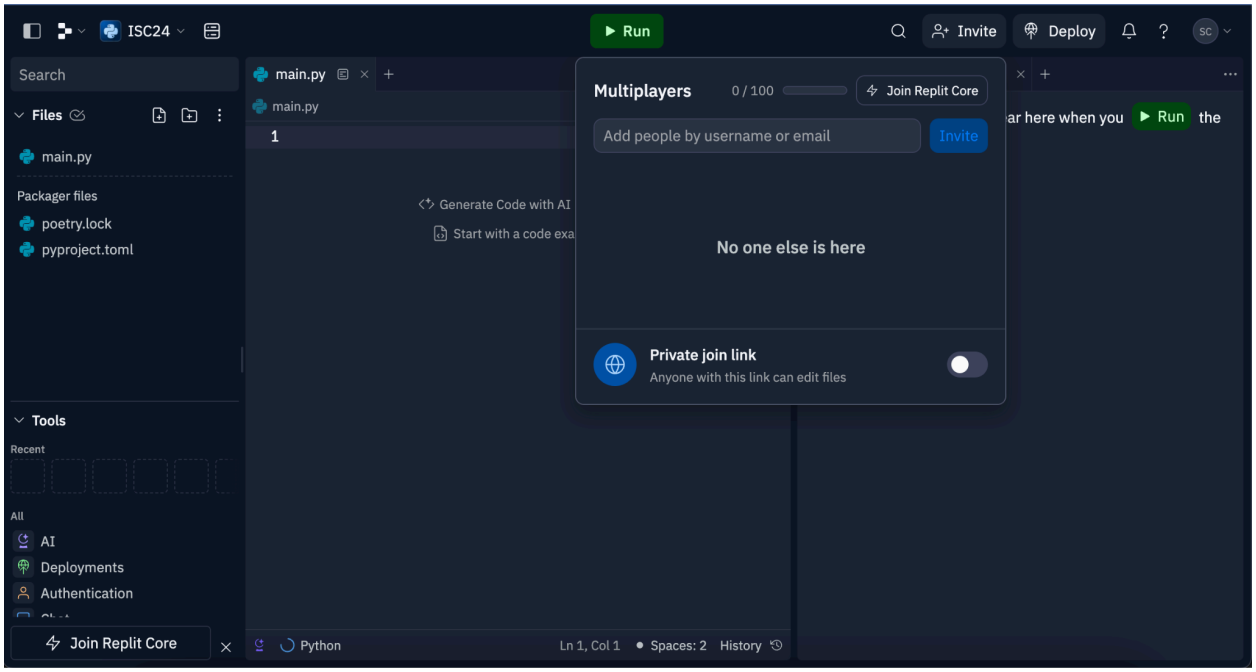
Click the '+ Create Repl' at the top left and fill out the following…



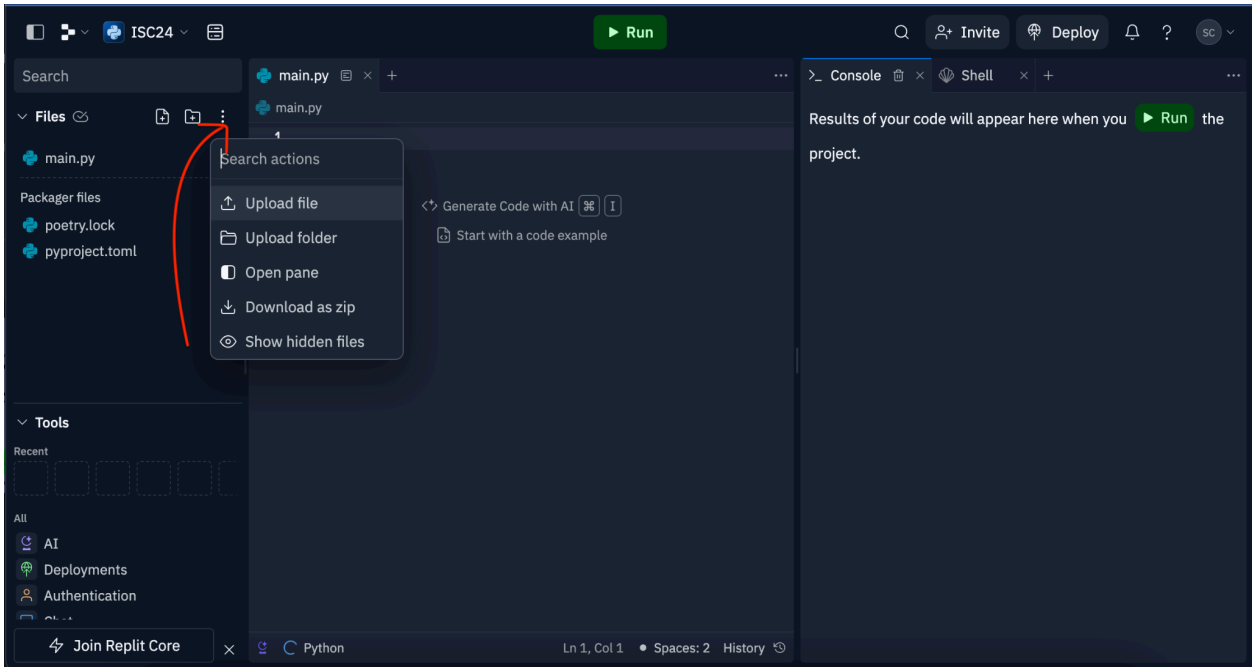
Template: Python
Title: ISC24



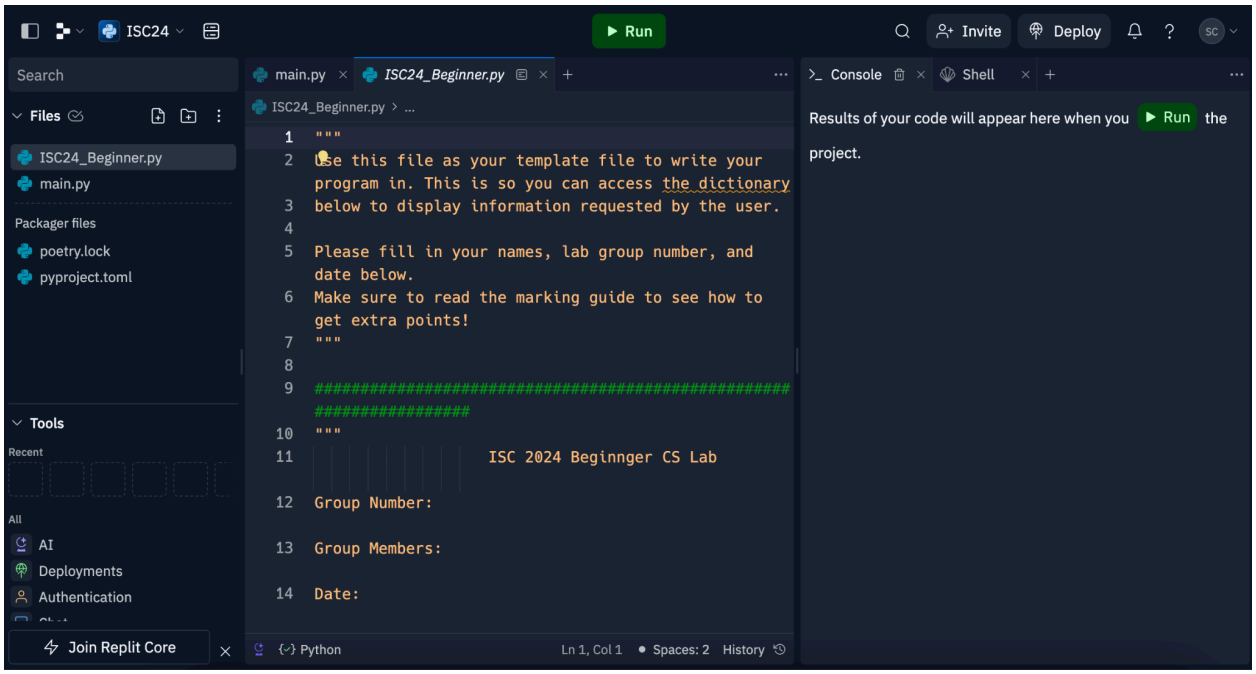
This is what your program/text editor should look like when you first start. You will type your program in the main/middle panel that says ‘main.py’ at the top. You will see the output of your code in the right panel that says ‘Console’. The left panel is where you will see your files and other options.



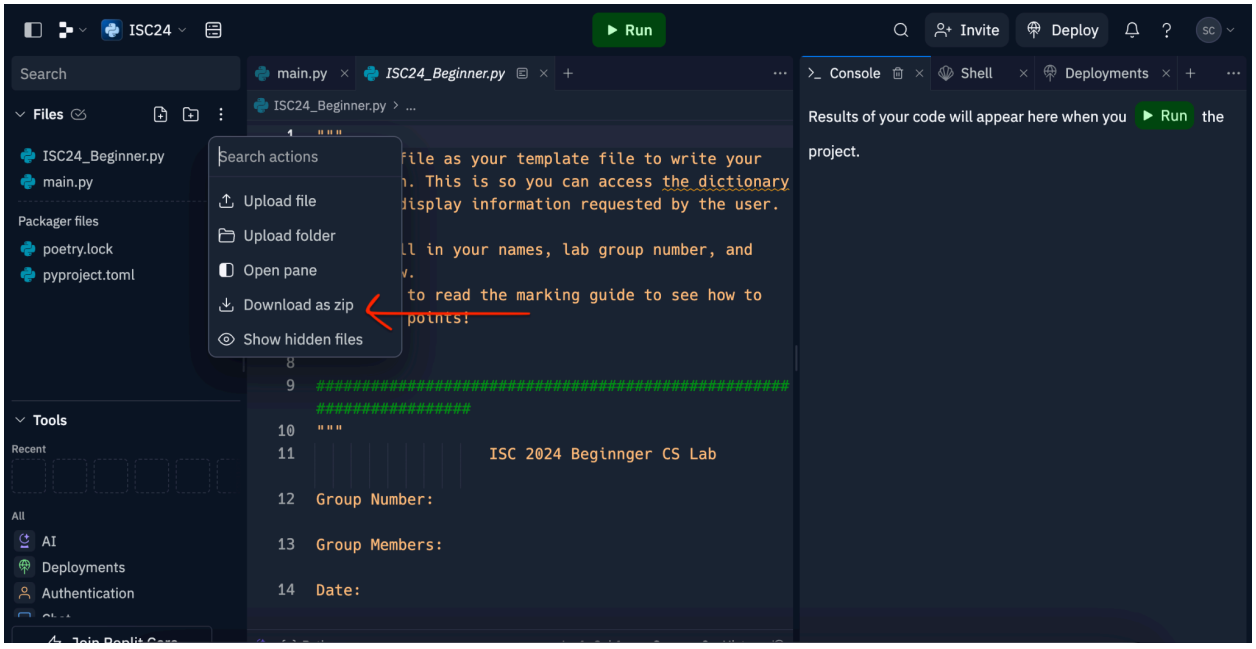
In the top right, click the ‘Invite’ button to add your teammates to your workspace. This is so you can collaborate with your team on the program. Think of it like a Google Docs programming platform. You will only need one file made for each team, but every member of your team should have a Repl.it profile for them to access the workspace.



Uploaded the provided lab template files by clicking the 3 dots and choosing the option ‘Upload file’



After uploading the file, you should see it appear in your left panel under ‘Files’.
Make sure to create & edit your file in ISC24_X.py (X = Beginner / Intermediate) and NOT the pre-loaded main.py.



After you’re done creating your program, click the 3 dots again and choose ‘Download as zip’

Submit your .zip files here!