



Minor Challenge Set #2

STEM Field: Software Engineering

Level: Senior

Challenge Name: Coding Games with Python

Project cost: 0 USD

Materials required:

- Computer with internet access
- (For Part B) A Google account to access Google Drive

Duration:

- The challenge takes approximately 3 hours to finish, however, the time guideline is an estimation only, and students and mentors can complete the tasks around their schedules
- Students can attempt the Minor Challenge – Software Engineering Intermediate project if they prefer learning the basics of Python programming first. Students can submit the Intermediate project regardless of their age category
- Students can attempt either part A or part B, or both

Introduction:

Python is a programming language that can be used to create programs, games and more.

In this project, you have the choice to develop a game (or two!) using Python. You can complete part A or part B, or attempt both if you have time and want to practice writing codes in Python.

In **part A (pages 3-8)**, we will be creating a well known game: rock, paper, scissors. Before we code the game, it is important to first think about what steps will be needed to accomplish our task. In this project, we will ask the user to choose between “Rock”, “Paper”, or “Scissors”. Then, the computer will randomly pick between the three options. The outcome of the game will be then determined. You will be guided on how to translate these steps into code below!

In **part B (pages 9-10)**, we will be using Colab to create a game in Python. Colab allows us to write code, test the code and save it onto Google Drive! In Colab, you can also add text in between the lines of code, which you will see in our sample file. We will be creating a text-based “Choose your own Adventure” game in Python. This game will take user input to decide what actions to take. Different actions will lead to different “rooms”, which will be defined by functions.

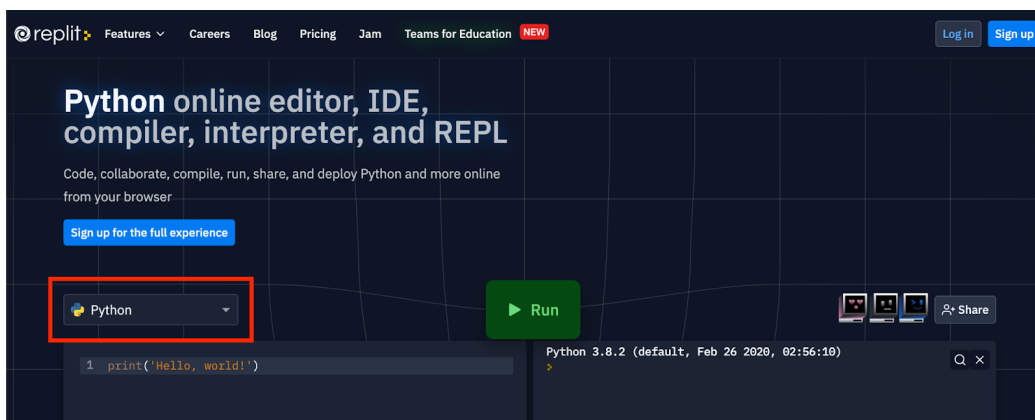
Here is a glossary of new words you may encounter in the instructions:

- **Comment:** Any words that are on the same line as the # but after the # are called “comments”. This means that these words are invisible to the computer when it compiles the code. They are essentially comments you include to explain your ideas behind your code.
- **Variable:** A name attached to a particular object.
- **While Loop:** The code inside of a while loop will continue to run WHILE the condition for a while loop is true.
- **Boolean:** A variable with two possible values: 0 (false) or 1 (true).

Instructions:

Part A: Rock, Paper, Scissors Game

- 1) We will write Python codes on an online programming environment called Replit. Navigate to <https://replit.com/languages/python3> on your computer. Make sure that the box in the red square in the image below is set to Python.



In steps 2-9, we will be describing what each line of code means. We will describe the line number so you can follow along. The line number is on the left hand side of the program.

```
1 #Import the random module to randomise the computer's
  choice
2 import random
3 #Asking for the user's input
4 user_action = input("Enter your choice (rock, paper,
  scissors): ")
```

You can find the full set of code in step 11. This set of code is not complete, and the computer cannot run this program now! You can copy and paste this set of code to Replit, then follow steps 2-9 to modify and complete the program.

- 2) (Lines 1-2) In this game, the computer's choice will be randomised. To do this, we need to import a module called random. Here is the first two lines of code to add to your program:

```
#Import the random module to randomise the computer's choice
import random
```

- 3) (Lines 3-4) Next we will ask the user for their input. In this case, their choices are "rock", "paper", or "scissors".

Write a comment to describe what you want to do in this step. An example is:

```
#Asking for the user's input
```

We recommend writing more descriptive comments to help other people better understand your code, and for your reference later on. Remember to put the symbol # in front of your comment!

We will assign the user's choice to a variable called user_action. Add the next line of code to your program:

```
user_action = input("Enter a choice (rock, paper, scissors): ")
```

- 4) (Lines 5-7) We will now create our first data structure, a list! In the same way that you might create a list of related items in your day-to-day life (for example, you may create a list of groceries), you can create a list of related items when you code. This allows you to have a single item (a list) which contains a bunch of related items you may want to access in the same context.

Here, we want the computer to choose one out of three items in a list: "rock", "paper", or "scissors". We will name the list as

possible_actions. Then we will use random.choice() to have the computer randomly select between the actions.

Replace [Insert your comment here] with your comment, then add the codes to your program.

```
#[Insert your comment here]
possible_actions = ["rock", "paper", "scissors"]
computer_action = random.choice(possible_actions)
```

- 5) (Lines 8-9) We want to see the choices the user and the computer made. We will print these choices using print.

Replace [Insert your comment here] with your comment. Then add the below code to your program.

```
#[Insert your comment here]
print(f"\nYou chose {user_action}, computer chose {computer_action}.\n")
```

- 6) (Lines 10-12) We will now try to determine the winner of the game, by using if, elif, and else statements. When using these statements, you are comparing the user's action to the computer's, and determine the winner. In the first case, the user's and the computer's actions are the same.

```
#If the user's and the computer's choices are the same, it is a tie
if user_action == computer_action:
    print(f"Both players selected {user_action}. It's a tie!")
```

- 7) (Lines 13-18) In the second case, if the computer's action is "rock" and the user's action is "scissors", then the user (us!) will lose the game. However, if the computer's action is "rock", but the user's action is "paper", then the computer loses!

Replace [Insert your comment here] with your comment describing this case.

```
#[Insert your comment here]
elif computer_action == "rock":
    if user_action == "scissors":
        print("You lose!")
    else:
        print("You win!")
```

- 8) (Lines 19-24) In the third case, if the computer's action is "paper" and the user's action is "rock", then the user will lose the game. Otherwise, if the computer's action is "paper" and the user's action is "scissors", then we win the game.

Replace [Insert your comment here] with your comment. Replace [Describe the outcome here] with a statement you want to print when the user's win or lose the game.

```
#[Insert your comment here]
elif computer_action == "paper":
    if user_action == "rock":
        print("[Describe the outcome here]")
    else:
        print("[Describe the outcome here]")
```

- 9) (Lines 25-30) We will now finish off the game with the final scenario, that is, the computer's action is "scissors", and the user's action is either "rock", or "paper".

Replace [Insert your comment here] with your comment. Complete the code by replacing [Computer action here] with the

computer's action for this scenario; [User action here] with the user's action; and [Describe the outcome here] with the game's outcome.

```
#[Insert your comment here]
elif computer_action == "[Computer action here]":
    if user_action == "[User action here]":
        print("[Describe the outcome here]")
    else:
        print("[Describe the outcome here]")
```

- 10) Congratulations, you have developed your very own rock, paper, scissors game! Click “Run” to play your game a few times and see how many times you can win against the computer!
- 11) Click [here](#) to download the full set of code. If you cannot download the text file, please email us at scichal@robogals.org and we can send you a copy!

Alternatively, you can copy and paste this set of code to Replit.
Note: This set of code is not complete! You have to complete steps 2-9 before running this program!

```
#Import the random module to randomise the computer's choice
import random
#Asking for the user's input
user_action = input("Enter your choice (rock, paper, scissors): ")
#[Insert your comment here]
possible_actions = ["rock", "paper", "scissors"]
computer_action = random.choice(possible_actions)
#[Insert your comment here]
print(f"\nYou chose {user_action}, computer chose {computer_action}.\n")
#If the user's and the computer's choices are the same, it is a tie
if user_action == computer_action:
    print(f"Both players selected {user_action}. It's a tie!")
#[Insert your comment here]
elif computer_action == "rock":
```

```
if user_action == "scissors":
    print("You lose!")
else:
    print("You win!")
#[Insert your comment here]
elif computer_action == "paper":
    if user_action == "rock":
        print("[Describe the outcome here]")
    else:
        print("[Describe the outcome here].")
#[Insert your comment here]
elif computer_action == "[Computer action here]":
    if user_action == "[User action here]":
        print("[Describe the outcome here]")
    else:
        print("[Describe the outcome here]")
```

Extension:

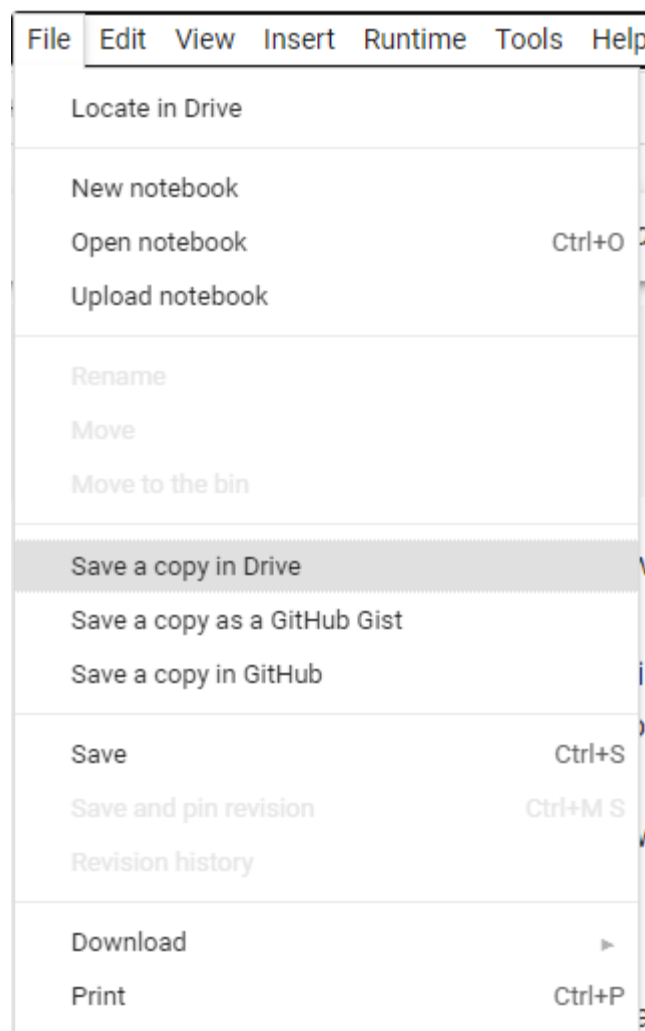
You can add in code to play the game several times in a row. To do this, you can add a while loop after importing the random module. You will also need to add an if statement, so the user can decide if they want to continue playing the game or not.

Try writing your own code to implement this feature! If you get stuck, have a look at the code in this website.

<https://realpython.com/python-rock-paper-scissors/>

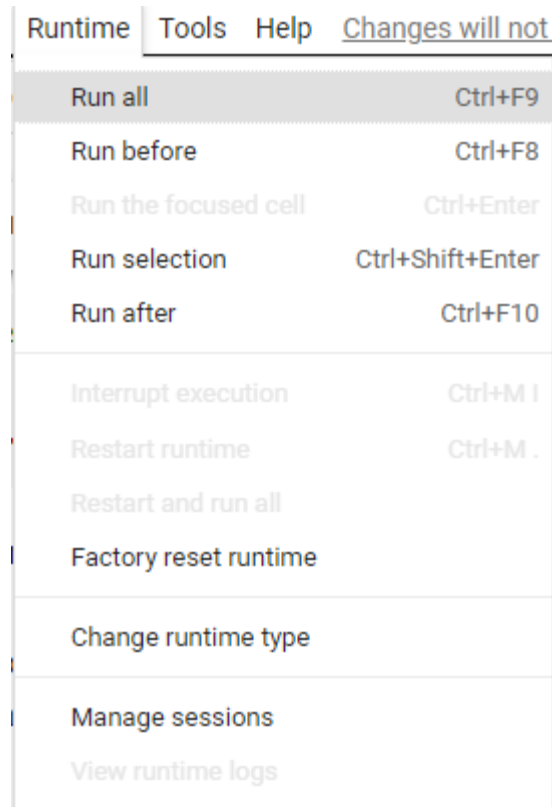
Part B: Adventure Game

- 1) You will need a Google account for this part. If you haven't got one, create a Google account.
- 2) Navigate to our [sample Colab file](https://colab.research.google.com/drive/10N7hndkhaT7bbcjlhX9xcitLFZwZeuuF?usp=sharing):
<https://colab.research.google.com/drive/10N7hndkhaT7bbcjlhX9xcitLFZwZeuuF?usp=sharing>
- 3) Click File → Save a copy to Drive.



- 4) Navigate to your Google Drive to find the copy you just created.
This is the file you will edit.

- 5) Read the text and code in the Colab file. Fill in the blanks in the code.
- 6) To test smaller pieces of your code, you can click the play button besides each chunk of code. To run your entire program click Runtime → Run All.



- 7) Make sure to save your code. When you are finished, download it as an .ipynb file.

Reflection Questions:

For both parts:

- Are there any improvements you would make to this challenge?
- What real world application can you apply the challenge to?
- What are the key concepts of science and engineering that relate to this challenge?

Part A:

- Are there any other games you could code?
- What could you add to this game to make it more exciting?
- How would you make this into a two player game for you and a friend?

Part B:

- How did you use functions in Python to create your game?
- How did you collect user input in your game?
- What other things could you change in your game to make it more robust (i.e. less prone to error)?

Submission Guidelines:

- **Part A:** Copy and paste your code to a text file. Submit your code with a screenshot of the output of your code. Include a short summary that addresses the Reflection Questions.
- **Part B:** Download and submit your code as an .ipynb file. Include a short summary that addresses the Reflection Questions.

Note: Remember, if you want to upload pictures of your Minor Challenge that also include you, please check if it is OK with your mentor first.

- There is a submission form directly on the Minor Challenge page here:
<https://sciencechallenge.org.au/index.php/minor-challenges/>.
Fill out the details and make sure you upload your submission.

Learn More! Resources:

- Python is a powerful programming language with many tools and applications. If it is confusing at first, don't worry! If you want to dive deeper into using functions in Python, for example passing arguments into a function, watch [this video](#).
<https://www.programiz.com/python-programming/function>
- CodeCombat is a fun and interactive learning platform. You can learn other programming concepts, how to code in Python, JavaScript, and HTML. It is a free website and no registration is required. (Select Student, then choose "No Class Code").
<https://codecombat.com/play?classCode=>

Bibliography:

- W3schools.com. 2022. *Python - Global Variables*. [online] Available at: <https://www.w3schools.com/python/python_variables_global.asp> [Accessed 26 February 2022].
- Wilkerson, C., 2022. *Make Your First Python Game: Rock, Paper, Scissors! – Real Python*. [online] Realpython.com. Available at: <<https://realpython.com/python-rock-paper-scissors/>> [Accessed 26 February 2022].