

Week 0, Problem 2: Rochambeau

Copied from:

<https://www.cs.hmc.edu/twiki/bin/view/CS5/RPSWeek0Gold3/22/2017>

[20 points; individual or pair]

A rock, paper, scissors program

This second problem asks you to practice

- creating new python files (in this case, by copying old ones or starting with the code below)
- writing a bit of your own code (by altering our starter program, if you wish)
- text-based input and output in Python

Start with a new file named `hw0pr2.py`

Start with your interactive file from lab—or - if you don't have that one - create a new file named `hw0pr2.py` in a text editor...

- Remember to be sure to type the `.py` extension: it enables syntax highlighting, i.e., code colorization...
- Start by pasting in the following code, which gets you on the way and matches some of our class discussion

```
• # coding: utf-8
• #
• # hw0pr2.py
• #
•
• """
• Notes on this rps function:
•
•
• """
•
•
• import random          # imports the library named random
```

```

•
• def rps():
•     """ this plays a game of rock-paper-scissors
•         (or a variant of that game ...)
•         inputs: no inputs      (prompted text doesn't count as input)
•         outputs: no outputs    (printing doesn't count as output)
•     """
•     user = input("Choose your weapon: ")
•     comp = random.choice( ['rock','paper','scissors'] )
•     print()
•
•     print('the user (you) chose', user)
•     print('the comp (I)   chose', comp)
•     print()
•
•     if user == 'rock':
•         print('Ha! I really chose paper -- I WIN!')
•
•     print("Better luck next time...")
•

```

What's next? What's required?

- Save this file as `hw0pr2.py`
 - *Do I really have to use the name `hw0pr2.py`?* **Yes!** Please do...

The submission system is picky about the name of the files you're submitting!
- From within ipython - and from within the correct directory! - run the file with `run hw0pr2`
 - This loads the file. You should up-arrow to do this each time you change the file!
 - When it successfully runs, then, type `rps()` to run the `rps` function.
- The basic requirements are these: use the discussion from class and `hw0pr2.py`'s starter code (below) to create a program that
 - invites the user to play a game of "rock-paper-scissors."
 - It must let the user choose from among at least three options
 - They don't have to be rock, paper, and scissors -- feel free to vary the actors!

- **but your program does have to work differently for each of three distinct inputs** - well, at least three inputs...
- Your program is welcome to play an honest game of RPS
 - but you're also welcome to create a player that always wins (or, if you prefer, that always loses). * Your program must echo the choice the user made
 - You may assume the user will type her/his choice correctly, according to your game
- Your program should reveal the choice that it makes (whether fair or not)
- Your program should print out who won that round (or whether it was a tie, or some other outcome...)
- Adding side comments to the graders who will be running your program is optional, but strongly encouraged!

More details

- In brief, the program should ask the user to choose rock, paper, or scissors (or your own variants!) Then, it should repeat back to the user their choice, it should "reveal" its own choice, and then report the results. The program can play fairly, can always win, or can always lose—it's up to you. If RPS is unfamiliar, in the game of [rock-paper-scissors](#), rock defeats scissors, scissors defeat paper, and paper defeats rock.
- You may assume that the user will input one of `rock`, `paper`, or `scissors`. Case matters! We'll stick with lower case...
- You may write the dialog however you like—below is an example dialog if you'd like one to follow. We are **positive** that you can improve on this interaction, however! Here are two distinct runs of the program:

```
In [1]: run hw0pr2.py
```

```
In [2]: rps()
Choose your weapon: rock
the user (you) chose rock
the comp (I) chose scissors
```

```
Ha! I really chose paper -- I WIN!
```

```
Better luck next time...
```

```
In [3]: rps()  
Choose your weapon: paper  
the user (you) chose paper  
the comp (I) chose dynamite
```

```
Dynamite!? I REALLY WIN!  
You can't play again.
```

- **Grutors** are Mudd's name for the students who grade assignments and hold tutoring hours. Note that taunting and/or praising the grutors with your rock-paper-scissors dialog is encouraged—and even more, as the semester wears on and the grutors become increasingly sleep-deprived.

Other Possibilities

- Remember that there is one other Python (and two Picobot) problems also due this week, so proceed at your own risk!
- Too much time on your hands? Add "lizard" and "spock" as noted [at this RPSSL link](#).
Even more time? Consider [RPS-25](#), a strict superset of RPS.
But if you have enough time for [RPS-101](#), there's a problem!
- Want your program to continue playing many times? Use a `while True: loop`.

We'll provide two examples instead of detailed explanations:

- ```
while True:
 print("Still running...")
 response = input("Play again? ")
 if response == 'n':
 break
```

Here is another possibility:

```
running = True
while running:
 response = input("Play again? ")
 if response == 'n':
```

```
running = False
```

Feel free to ask one of the tutors/instructors about this... .

These aren't officially extra-credit, though it's common for the graders to award additional points for creative, interesting, and/or unusual submissions!

However, if you're looking for *official* extra-credit challenges, you can always expand your interactive fiction - or Picobot challenges numbers 5 and 6...

### **Submitting your `hw0pr2.py` file**

---

- Once your program plays a game of rock-paper-scissors (fairly or not), submit it at the [submission site](#). Be sure to indicate the partner you worked with (if any).
- You can always submit again if you want to change or update your file—all of the versions are kept, but only the last one before the deadline is graded. This makes the submission site a way to access the same files from different computers.

### **On to Problem 10, the choice problem...**

---

- Problem 10 ("one-zero") each week offers a *choice* of some sort... . Some weeks the choice will be two problems from which you can choose one (or do both for extra credit). This week the "choice" is more literal, both literally and literarily! [Here's the link to week 0's Problem 10 \(the choice problem\)](#)
- Problems 3 and 4 on this assignment emphasize a very different way of thinking about computation. They don't use python, but a simulation and language named *Picobot*. We will discuss Picobot in the second day of the first week of CS 5 classes. [Here's the link to the Picobot problems.](#)