Exercise: Rock, Paper, Scissors, Fire, Water

Table of Contents

Background

Rock, paper, scissors is a simple game where two players each simultaneously choose one of three items (rock, paper, or scissors). If both players chose the same item, the game ends in a draw; otherwise, rock beats scissors, scissors beat paper, and paper beats rock.

You have been provided with two versions of a program that determines the winner of this game. One version should work properly; the other has some bugs. Both versions of the program have a function called `rps()`; the header and docstring of this function are provided below:

```python
def rps(item1, item2):
    """ Determine the winner of a game of rock, paper, scissors.

    Args:
        item1 (str): one of the following strings: "rock", "paper", or
            "scissors".
        item2 (str): another of the three strings mentioned above. May or may
            not be identical to item1.

    Returns:
        int: 0 if a tie, 1 if item1 defeats item2, or 2 if item2 defeats item1.
    """
```

Problem statement

Write a Pytest test script to test the `rps()` function described above.

Pytest is a Python library that makes it easy to write and run tests for your code. You can learn more about Pytest here. For now you only need to know the basics of how to write a test script. You can find a simple example of a Pytest test script here.

IMPORTANT	Pytest test scripts names should start with word `test_` and should not have any arguments. For this exercise, your test script should be named `test_rps.py`.

Instructions

At the start of your test script, import the `rps()` function from one of the two scripts. (You can edit this import statement to apply your tests to the other script.)

Write one or more test functions that collectively cover all nine possible rock, paper, scissors outcomes:

- rock vs. rock
- rock vs. paper
- rock vs. scissors
- paper vs. paper
- paper vs. scissors
- paper vs. rock
- scissors vs. scissors
- scissors vs. rock
- scissors vs. paper

IMPORTANT	Functions in a pytest script must start with the word `test_` and must not have any arguments.

NOTE	You can use the provided `rps.py` and `rps_buggy.py` files to test your script. Your script will be submitted to Gradescope, however, this script will not be autograded. Instead, your instructor will manually grade your script.

Installing Pytest

Before you can run your script, you must have installed Pytest. If you have already installed Pytest, you can skip this section. If you have not yet installed Pytest, do so by typing the following at the terminal (Windows users, replace `python3` with `python`):

```
python3 -m pip install --upgrade pip
python3 -m pip install --upgrade pytest
```

Running your test script

Your test script should be saved in the same directory as the program you are testing. Ensure that

VS Code is opened to that directory. Then open the terminal by pressing Ctrl+` or going to the View menu and selecting Terminal. The following example shows how to run a test script called `test_rps.py`; replace this with the name of your script.

```
pytest test_rps.py
```

If that doesn't work, try the following (Windows users, replace `python3` with `python`):

```
python3 -m pytest test_rps.py
```