



Project:  
Rock Paper Scissors



Getting started

SEARCH



RESOURCES



CONCEPTS



- ✓ 1. This project rocks!
- ✓ 2. Getting started
- ★ 3. Project: Rock Paper Scissors



## Getting started

Here are some steps to get you started in the Rock Paper Scissors project! As you work through the project, make sure to test your program by running it.

### 1. Download the starter code

To get the starter code, go to [this GitHub page](#). If you've never used GitHub, don't worry—just read the directions given in the README and download the files as described. If you want to download all the files at once, just click on the green **Clone or download** button and then click **Download ZIP**.

The starter code gives you a place to begin, with **Player** and **Game** classes that are mostly empty. Over the course of the project, you will be greatly expanding the classes and methods in this program.

*Read* the starter code, and *run* it on your computer to see what it does.

Try importing it into the Python interpreter and experimenting with the **Player** and **Game** objects.

### 2. Create a player subclass that plays randomly

The starter **Player** class always plays **'rock'**. That's not a very good strategy! Create a subclass called **RandomPlayer** that chooses its move at random. When you call the **move** method on a **RandomPlayer**



Knowledge

Get support and stay on track

< Project:  
Rock Paper Scissors



Getting started

SEARCH



RESOURCES



CONCEPTS



- ✓ 1. This project rocks!
- ✓ 2. Getting started
- ★ 3. Project: Rock Paper Scissors

Change the code so it plays a game between two **RandomPlayer** objects.

### 3. Keep score

The starter **Game** class does not keep score. It doesn't even notice which player won each round. Update the **Game** class so that it displays the outcome of each round, and keeps score for both players. You can use the provided **beats** function, which tells whether one move beats another one.

Make sure to handle ties — when both players make the same move!

### 4. Create a subclass for a human player.

The game is a lot more interesting if you can actually play it, instead of just watching the computer play against itself. Create a **HumanPlayer** subclass, whose **move** method asks the human user what move to make. (Take another look back at the project demo to see what this can look like!)

Set the program to play a game between **HumanPlayer** and **RandomPlayer**.

### 5. Create player classes that remember

At the end of each game round, the **Game** class calls the **learn** method on each player object, to tell that player what the other player's move was. This means you can have computer players that change their moves depending on what has happened earlier in the game. To do this, you will need



Project:  
Rock Paper Scissors



Getting started

SEARCH



RESOURCES



CONCEPTS



- ✓ 1. This project rocks!
- ✓ 2. Getting started
- ★ 3. Project: Rock Paper Scissors

Create a **ReflectPlayer** class that remembers what move the *opponent* played last round, and plays that move *this* round. (In other words, if you play **'paper'** on the first round, a **ReflectPlayer** will play **'paper'** on the second round.)

Create a **CyclePlayer** class that remembers what move `_it_` played last round, and cycles through the different moves. (If it played **'rock'** this round, it should play **'paper'** in the next round.)

*(Something to think about: What should these classes do on the first move?)*

Test each of these player classes versus **HumanPlayer**.

## 6. Validate user input

The human player might sometimes make typos. If they enter `roxk` instead of `rock`, the **HumanPlayer** code should let them try again. (See how this works in the demo if you type something in that isn't a valid move.)

## 7. Announce the winner

It's up to you how long the game should run. The starter code always plays three rounds, but that's not the only way it could work. You could choose to continue until the player types **quit**, or you could have the game run until one player is ahead by three points, or any other rule that makes sense to you.

<

Project:  
Rock Paper Scissors

SEARCH

RESOURCES

CONCEPTS

✓

1. This project rocks!

✓

2. Getting started

★

3. Project: Rock Paper Scissors

☰

Getting started

are.

8. Check your code formatting

Use the **pycodestyle** tool to check the formatting of your code. Make the edits that it recommends, then re-run it to see fewer and fewer warnings. By the time you're done, it should display *no warnings or errors* at all.

NEXT