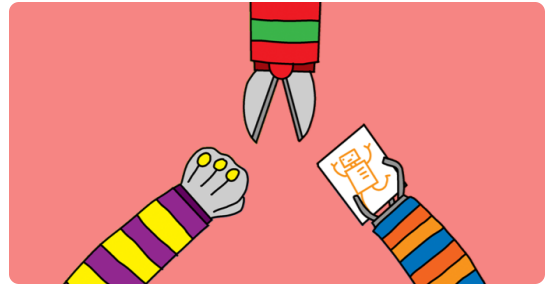




Projects

Rock, Paper, Scissors

Create your own 'Rock, Paper, Scissors' game.



Step 1 Introduction

In this project you will make a Rock, Paper, Scissors game and play against the computer.

Rules: You and the computer both choose rock, paper or scissors. The winner is decided by these rules:

- Rock blunts scissors
- Paper covers rock
- Scissors cut paper

```
rock (r), paper (p) or scissors (s)? s  
>8 vs ____  
Player wins!
```

Additional information for club leaders

If you need to print this project, please use the **Printer friendly version** (<https://projects.raspberrypi.org/en/projects/rock-paper-scissors/print>).



Club leader notes

Introduction:

In this project, children will learn how to make a program that plays Rock, Paper, Scissors using variables and conditional statements.

Online Resources

This project uses Python 3. We recommend using **trinket** (<https://trinket.io/>) to write Python online. This project contains the following Trinkets:

- ‘Rock, Paper, Scissors’ Resources – jumpto.cc/rps-go (<http://jumpto.cc/rps-go>)

There is also a trinket containing the completed project:

- ‘Rock, Paper, Scissors’ Finished – trinket.io/python/e1e1d873be (<https://trinket.io/python/e1e1d873be>)

Offline Resources

This project can be **completed offline** (<https://www.codeclubprojects.org/en-GB/resources/python-working-offline/>) if preferred. You can access the project resources by clicking the ‘Project Materials’ link for this project. This link contains a ‘Project Resources’ section, which includes resources that children will need to complete this project offline. Make sure that each child has access to a copy of these resources. This section includes the following files:

- rock-paper-scissors/rock-paper-scissors.py

You can also find a completed version of this project in the ‘Volunteer Resources’ section, which contains:

- rock-paper-scissors-finished/rock-paper-scissors.py

(All of the resources above are also downloadable as project and volunteer **.zip** files.)

Learning Objectives

- Variables;
- Selection (**if**, **elif**, **else**);
- Boolean **==** and **and**.

This project covers elements from the following strands of the **Raspberry Pi Digital Making Curriculum** (<http://rpf.io/curriculum>):

- **Use basic programming constructs to create simple programs.** (<https://www.raspberrypi.org/curriculum-programming/creator>)

Challenges

- “ASCII art” – Use conditional logic to display ASCII art for rock, paper and scissors.
- “Create a new game” – Duplicate the Rock, Paper, Scissors game and remix it to create a new game.



Project materials

Project resources

- .zip file containing all project resources (<https://projects-static.raspberrypi.org/projects/rock-paper-scissors/06b14a127bcddaa3e22ccd8545c27c6ce508da3a/en/resources/rock-paper-scissors-project-resources.zip>)
- Online Trinket containing all 'Rock, Paper, Scissors' project resources (<http://jumpto.cc/rps-go>)
- rock-paper-scissors/rock-paper-scissors.py (<https://projects-static.raspberrypi.org/projects/rock-paper-scissors/06b14a127bcddaa3e22ccd8545c27c6ce508da3a/en/resources/rock-paper-scissors-rock-paper-scissors.py>)

Club leader resources

- .zip file containing all completed project resources (<https://projects-static.raspberrypi.org/projects/rock-paper-scissors/06b14a127bcddaa3e22ccd8545c27c6ce508da3a/en/resources/rock-paper-scissors-volunteer-resources.zip>)
- Online completed Trinket project (<https://trinket.io/python/e1e1d873be>)
- rock-paper-scissors-finished/rock-paper-scissors.py (<https://projects-static.raspberrypi.org/projects/rock-paper-scissors/06b14a127bcddaa3e22ccd8545c27c6ce508da3a/en/resources/rock-paper-scissors-finished-rock-paper-scissors.py>)

Step 2 Player's Turn

First, let the player choose Rock, Paper or Scissors.

- Open this trinket: **jump to cc/rps-go** (<http://jump to cc/rps-go>).
- The project already contains the code to import a function that you are going to use in this project.

```
main.py
#!/bin/python3

from random import randint
```

You'll use `randint` later to generate random numbers.

- First, let the player choose Rock, Paper or Scissors by typing the letter 'r', 'p' or 's'.

```
from random import randint
player = input('rock (r), paper (p) or scissors (s)?')
```

```
rock (r), paper (p) or
scissors (s)? r
```

- Now print out what the player chose:

```
player = input('rock (r), paper (p) or scissors (s)?')
print(player, 'vs')
```

```
rock (r), paper (p) or
scissors (s)? p
p vs
```

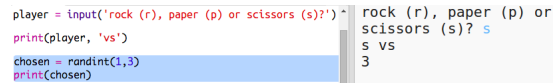
- Test your code by clicking **Run**. Click in the trinket output window and enter your choice.

Step 3 Computer's Turn

Now it's the computer's turn. You can use the `randint` function to generate a random number to decide between rock, paper and scissors.

- Use `randint` to generate a random number to decide whether the computer has chosen rock, paper or scissors.

```
player = input('rock (r), paper (p) or scissors (s)?')
print(player, 'vs')
chosen = randint(1,3)
print(chosen)
```

A screenshot of a code editor with a light gray background. The code is written in a monospaced font. The first line is `player = input('rock (r), paper (p) or scissors (s)?')`. The second line is `print(player, 'vs')`. The third line is `chosen = randint(1,3)`. The fourth line is `print(chosen)`. To the right of the code, there is a small window showing the output of the program: `rock (r), paper (p) or scissors (s)? s` followed by `s vs` and `3`.

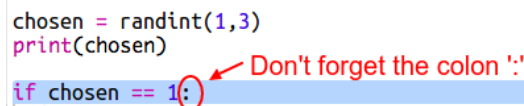
- Run your script lots of times (you'll need to enter 'r', 'p' or 's' each time.)

You should see that 'chosen' is randomly set to either 1, 2 or 3.

- Let's say:
 - 1 = rock (r)
 - 2 = paper (p)
 - 3 = scissors (s)

Use `if` to check if the chosen number is 1 (`==` is used to see if 2 things are the same).

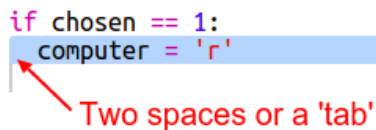
```
chosen = randint(1,3)
print(chosen)
if chosen == 1:
```

A screenshot of a code editor showing the code from the previous block. The line `if chosen == 1:` is highlighted in blue. A red arrow points to the colon at the end of the line, with the text "Don't forget the colon ':'" written in red.

- Python uses **indentation** (moving the code to the right) to show which code is inside the `if`. You can either use two spaces (tap the spacebar twice) or tap the **tab key** (usually above CAPSLOCK on the keyboard.)

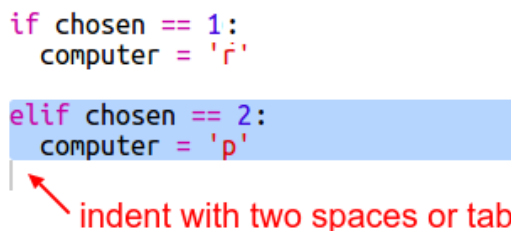
Set `computer` to 'r' inside the `if` using indentation:

```
if chosen == 1:
    computer = 'r'
```

A screenshot of a code editor showing the code from the previous block. The line `computer = 'r'` is indented under the `if` statement and is highlighted in blue. A red arrow points to the indentation, with the text "Two spaces or a 'tab'" written in red.

- You can add an alternative check using `elif` (short for *else if*):

```
if chosen == 1:
    computer = 'r'
elif chosen == 2:
    computer = 'p'
```

A screenshot of a code editor showing the code from the previous block. The `elif` block is highlighted in blue. A red arrow points to the indentation of the `elif` block, with the text "indent with two spaces or tab" written in red.

This condition will only be checked if the first condition fails (if the computer didn't choose 1)

- And finally, if the computer didn't choose 1 or 2 then it must have chosen 3.

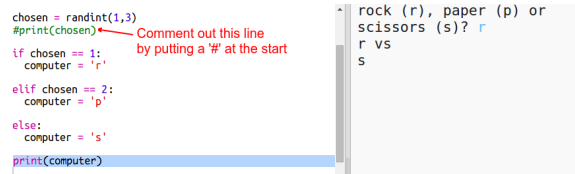
This time we can just use `else` which means otherwise.

```
if chosen == 1:
    computer = 'r'

elif chosen == 2:
    computer = 'p'

else:
    computer = 's'
```

- Now, instead of printing out the random number that the computer chose you can print the letter.



```
chosen = randint(1,3)
#print(chosen)
if chosen == 1:
    computer = 'r'
elif chosen == 2:
    computer = 'p'
else:
    computer = 's'
print(computer)
```

You can either delete the line `print(chosen)`, or make the computer ignore it by adding a `#` at the start of the line.

- Test your code by clicking Run and choosing your option.
- Hmm, the computer's choice gets printed on a new line. You can fix that by adding `end=' '` after `vs`, that tells Python to end with a space instead of a new line.



```
print(player, 'vs', end=' ')
chosen = randint(1,3)
#print(chosen)
```

- Play the game a few times by clicking Run and making a choice.

For now you'll have to work out who won yourself. Next you'll add the Python code to work this out.

Step 4 Check the Result

Now let's add the code to see who won.

- You need to compare the `player` and `computer` variables to see who won.

If they're the same then it's a draw:

```
print(computer)
if player == computer:
    print('DRAW!')
```

rock (r), paper (p) or scissors (s)? s
s vs s
DRAW!

- Test your code by playing the game a few times until you get a draw.

You'll need to click **Run** to start a new game.

- Now let's look at the cases where the player chose 'r' (rock) but the computer didn't.

If the computer chose 's' (scissors) then the player wins (rock beats scissors).

If the computer chose 'p' (paper) then the computer wins (paper beats rock).

We can check the player choice *and* the computer choice using `and`.

```
if player == computer:
    print('DRAW!')

elif player == 'r' and computer == 's':
    print('Player wins!')

elif player == 'r' and computer == 'p':
    print('Computer wins!')
```

- Next let's look at the cases where the player chose 'p' (paper) but the computer didn't:

```
elif player == 'r' and computer == 's':
    print('Player wins!')

elif player == 'r' and computer == 'p':
    print('Computer wins!')

elif player == 'p' and computer == 'r':
    print('Player wins!')

elif player == 'p' and computer == 's':
    print('Computer wins!')
```

- And finally, can you add the code to check for the winner when the player chose 's' (scissors) and the computer chose rock or paper?
- Now play the game to test your code.

```
rock (r), paper (p) or scissors (s)? s  
s vs p  
Player wins!
```

Click **Run** to start a new game.

Step 5 Challenge: ASCII Art

Instead of using the letters r, p and s to represent rock, paper and scissors, can you use ASCII art?

For example:

```
rock (r), paper (p) or scissors  
(s)? s  
>8 vs ____  
Player wins!
```

Where:

```
rock: 0  
paper: ____  
scissors: >8
```

- Instead of saying `print computer` you'll need to add a new line to each of the options in the `if` to print out the correct ASCII art.

Hints:

```
if chosen == 1:  
    computer = 'r'  
    print('0') ——— ASCII art rock
```

```
#print(computer)
```

- Instead of saying `print player` you'll need to add a new if statement to check which item the player chose and print out the correct ASCII art:

Hint:

```
player = input('rock (r), paper (p) or scissors (s)?')  
if player == 'r':  
    print('0', end=' ')
```

Remember that adding `end=' '` to the end of a `print` makes it end with a space instead of a new line.

Step 6 Challenge: Create a new game

Can you create your own game like Rock, Paper, Scissors with different objects?

Click the 'Duplicate' button to make a copy of your Rock, Paper Scissors project to start from.

This example uses Fire, Logs and Water:

```
Fire, Logs, Water
Fire burns Logs
Logs make a bridge over Water.
Water puts out Fire
fire (f), logs (l) or water (w)? l
@@@ vs ~~~
Player wins!
```

Published by Raspberry Pi Foundation (<https://www.raspberrypi.org>) under a Creative Commons license (<https://creativecommons.org/licenses/by-sa/4.0/>).

View project & license on GitHub (<https://github.com/RaspberryPiLearning/rock-paper-scissors>)