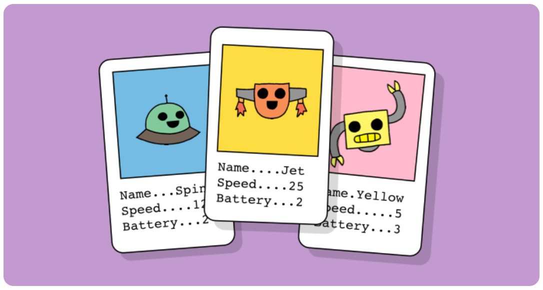
Projects

Robo-Trumps

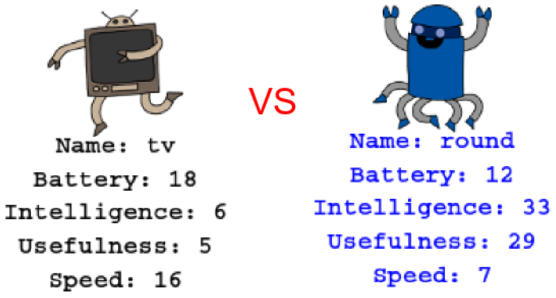
Create a database of robots and play Robo-Trumps with a friend!

Python



Step 1 Introduction

In this project you will read data from a file to create robot trump cards. Then you can play Robo-Trumps with a friend.



```
Name: tv
Battery: 18
Intelligence: 6
Usefulness: 5
Speed: 16
```

```
Name: round
Battery: 12
Intelligence: 33
Usefulness: 29
Speed: 7
```

Additional information for club leaders

If you need to print this project, please use the **Printer friendly version** (https://projects.raspberrypi.org/en/projects/robo-trumps/print).

Club leader notes

Introduction:

In this project learners will read data from a file into a dictionary, to create robot trump cards. The cards can then be used to play Robo-Trumps with a friend.

Online Resources

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

- **'Robotrumps' starting point – jumpto.cc/trumps-go** (http://jumpto.cc/trumps-go)

There is also a trinket containing a sample solution to the challenges:

- **'Robotrumps' Finished – trinket.io/python/9ccc368bd5** (https://trinket.io/python/9ccc368bd5)

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:

- robotrumps/robotrumps.py
- robotrumps/cards.txt
- Several robot .gif images

You can also find a completed version of this project's challenges in the 'Volunteer Resources' section, which contains:

- robotrumps-finished/robotrumps.py
- robotrumps-finished/cards.txt
- Several robot .gif images

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

Learning Objectives

- Consolidation: data structures, reading from file and turtle graphics;
- Using images in turtle graphics;
- Sharing trinket projects.

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

- **Combine programming constructs to solve a problem.** (https://www.raspberrypi.org/curriculum/programming/builder)

Challenges

- Add more robots - Editing a .txt file and working with images;
- Add more stats to the robots - handling data;
- Play Robotrumps with a friend.

Frequently Asked Questions

- **Offline Python does not work with .png images. .gif images have been supplied for offline use.**
- Note that this project uses both text input and turtle graphics, you can adjust the amount of space given to each in trinket.
- You need to click in the trinket text window before typing.
- Children can use the ‘Share’ option in trinket to get a link to their project. If they have access to email then they can email the link to a friend, otherwise they can just read it out, note it down on a piece of paper or save it in a text file on the local network.

Project materials

Project resources

- **.zip file containing all project resources** (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-project-resources.zip)
- **Online Trinket containing ‘Robo-Trumps’ starter resources** (http://jumpto.cc/trumps-go)
- **robo-trumps/robo-trumps.py** (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-robo-trumps.py)
- **robo-trumps/cards.txt** (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-cards.txt)
- **robo-trumps/space.gif** (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-space.gif)

- robo-trumps/rainbow.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-rainbow.gif)
- robo-trumps/bird.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-bird.gif)
- robo-trumps/dog.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-dog.gif)
- robo-trumps/jet.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-jet.gif)
- robo-trumps/round.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-round.gif)
- robo-trumps/brains.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-brains.gif)
- robo-trumps/twoheads.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-twoheads.gif)
- robo-trumps/shades.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-shades.gif)
- robo-trumps/hair.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-hair.gif)
- robo-trumps/tv.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-tv.gif)
- robo-trumps/yellow.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-yellow.gif)

Club leader resources

- .zip file containing all completed project resources (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robotrumps-volunteer-resources.zip)

- Online completed Trinket project (https://trinket.io/python/9ccc368bd5)
- robo-trumps-finished/robo-trumps.py (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-robo-trumps.py)
- robo-trumps-finished/cards.txt (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-cards.txt)
- robo-trumps-finished/space.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-space.gif)
- robo-trumps-finished/rainbow.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-rainbow.gif)
- robo-trumps-finished/bird.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-bird.gif)
- robo-trumps-finished/dog.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-dog.gif)
- robo-trumps-finished/jet.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-jet.gif)
- robo-trumps-finished/round.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-round.gif)
- robo-trumps-finished/brains.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-brains.gif)
- robo-trumps-finished/twoheads.gif (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-twoheads.gif)

- **robo-trumps-finished/shades.gif** (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-shades.gif)
- **robo-trumps-finished/hair.gif** (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-hair.gif)
- **robo-trumps-finished/tv.gif** (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-tv.gif)
- **robo-trumps-finished/yellow.gif** (https://projects-static.raspberrypi.org/projects/robo-trumps/d8863a216824e35ef27d7dcea445c6a6e54afeb9/en/resources/robo-trumps-finished-yellow.gif)

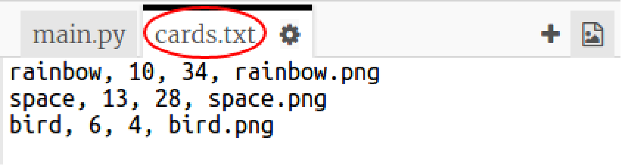
Step 2 Read robot data from a file

It's often useful to be able to read information from a file. You can then change the data in the file without having to change your code.

- Open this trinket: **jumpto.cc/trumps-go** (http://jumpto.cc/trumps-go).

- Your starter project includes a `cards.txt` file which contains data about robots.

  Click on `cards.txt` to see the data:



```
rainbow, 10, 34, rainbow.png
space, 13, 28, space.png
bird, 6, 4, bird.png
```

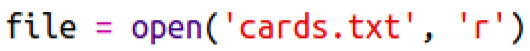
  Each line has data about a robot. The data items are separated by commas.

  Each line contains the following information:

name, intelligence rating, how long the battery lasts, image file name

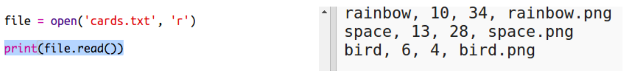
- Let's read the data in from the file so that you can use it.

  The first step is to open the `cards.txt` file in your script:



```
file = open('cards.txt', 'r')
```

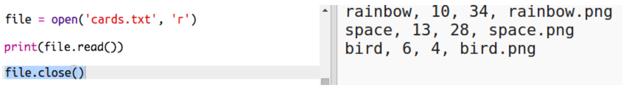
- Now you can read the data from the file:



```
file = open('cards.txt', 'r')
print(file.read())
```

```
rainbow, 10, 34, rainbow.png
space, 13, 28, space.png
bird, 6, 4, bird.png
```

- You should always close a file when you have finished with it:

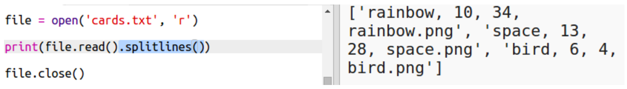


```
file = open('cards.txt', 'r')
print(file.read())
file.close()
```

```
rainbow, 10, 34, rainbow.png
space, 13, 28, space.png
bird, 6, 4, bird.png
```

- That gives us the file as one string, you need to break it down into the individual pieces of data.

  First, you can split the file into a list of lines:

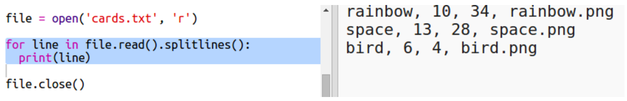


```
file = open('cards.txt', 'r')
print(file.read().splitlines())
file.close()
```

```
['rainbow, 10, 34,
rainbow.png', 'space, 13,
28, space.png', 'bird, 6, 4,
bird.png']
```

  Look carefully at the output. There are three items in the list, each one is a line from the file.

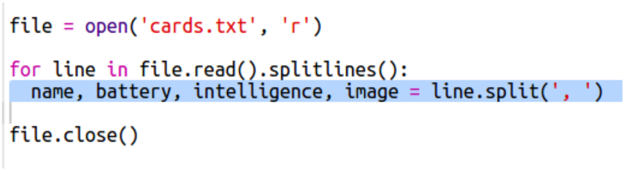
- Now you can loop over those lines one at a time



```
file = open('cards.txt', 'r')
for line in file.read().splitlines():
  print(line)
file.close()
```

```
rainbow, 10, 34, rainbow.png
space, 13, 28, space.png
bird, 6, 4, bird.png
```

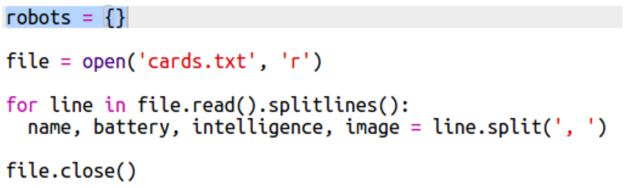
- Instead of printing out the lines, read them in to variables:



```
file = open('cards.txt', 'r')
for line in file.read().splitlines():
  name, battery, intelligence, image = line.split(', ')

file.close()
```

- You want to be able to use this data later to look up the values for a particular robot. Let's use the robot's name as a key to a dictionary.

Add a `robots` dictionary:

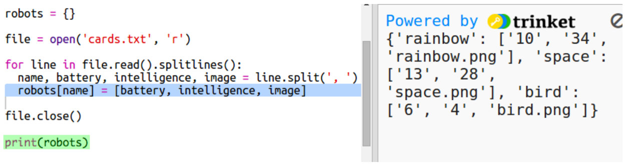


```
robots = {}

file = open('cards.txt', 'r')

for line in file.read().splitlines():
  name, battery, intelligence, image = line.split(', ')

file.close()
```

- Now let’s add an entry to the robots dictionary for each robot.

  The name is the key and the value is a list of data for that robot.

  Add the highlighted code:



```
robots = {}

file = open('cards.txt', 'r')

for line in file.read().splitlines():
  name, battery, intelligence, image = line.split(', ')
  robots[name] = [battery, intelligence, image]

file.close()

print(robots)
```

```
{'rainbow': ['10', '34',
'rainbow.png'], 'space':
['13', '28',
'space.png'], 'bird':
['6', '4', 'bird.png']}
```

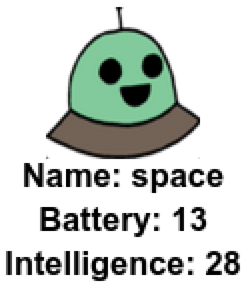
You can remove `print robots` when you have tested your script.

Step 3 Display the data

Now you can display the robot data in a more interesting way.

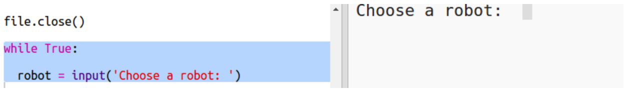
Let’s display a robot trump card with an image and data for its intelligence and usefulness.

When you’ve completed this step you’ll be able to display robots like this:



Name: space
Battery: 13
Intelligence: 28

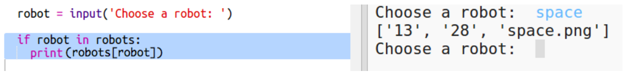
- Ask the user which robot they would like to see:



```
file.close()

while True:

  robot = input('Choose a robot: ')
```

```
Choose a robot: 
```

- If the robot is in the dictionary then lookup its data:

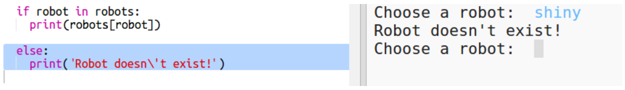


```
  robot = input('Choose a robot: ')

  if robot in robots:
    print(robots[robot])
```

```
Choose a robot:  space
['13', '28', 'space.png']
Choose a robot: 
```

  Test your code by entering a robot name.

- If the robot doesn't exist then give an error:



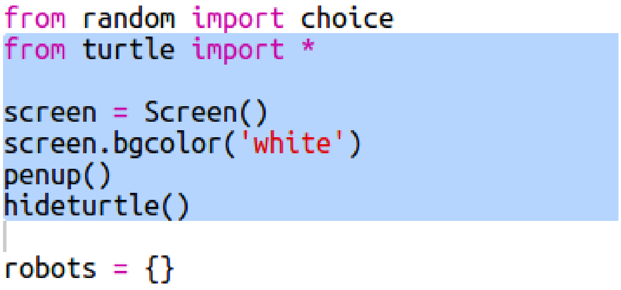
```
  if robot in robots:
    print(robots[robot])

  else:
    print('Robot doesn\'t exist!')
```

```
Choose a robot:  shiny
Robot doesn't exist!
Choose a robot: 
```

Test your code by entering a robot name that isn't in the dictionary.

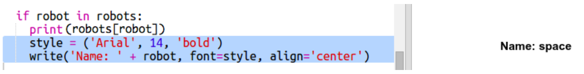
- Now you're going to use the Python turtle to display the robot data.

  Import the turtle library at the top of your script and setup the screen and turtle:



```
from random import choice
from turtle import *

screen = Screen()
screen.bgcolor('white')
penup()
hideturtle()

robots = {}
```

- Now add code to get the turtle to print the name of the robot:



```
  if robot in robots:
    print(robots[robot])
    style = ('Arial', 14, 'bold')
    write('Name: ' + robot, font=style, align='center')
```

Name: space

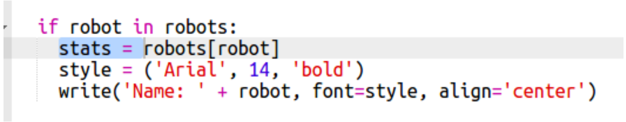
- Try changing the `style` variable until you are happy with the text.

  Instead of `Arial` you could try: `Courier`, `Times` or `Verdana`.

  Change `14` to a different number to change the size of the font.

  You can change `bold` to `normal` or `italic`.

- Store the list of stats for the robot in a variable instead of printing them out:

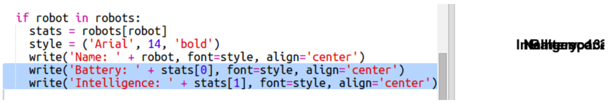


```
if robot in robots:
  stats = robots[robot]
  style = ('Arial', 14, 'bold')
  write('Name: ' + robot, font=style, align='center')
```

- You can now access the stats for the robot as items in a list:

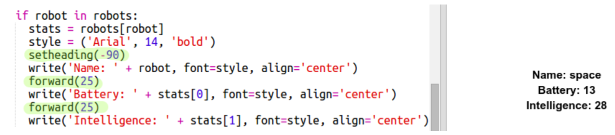
  - `stats[0]` is intelligence
  - `stats[1]` is battery
  - `stats[2]` is the image name

  Add code to display the intelligence and battery stats:



```
if robot in robots:
  stats = robots[robot]
  style = ('Arial', 14, 'bold')
  write('Name: ' + robot, font=style, align='center')
  write('Battery: ' + stats[0], font=style, align='center')
  write('Intelligence: ' + stats[1], font=style, align='center')
```

- Oh dear! The stats are all on top of each other. You'll need to add code to move the turtle:

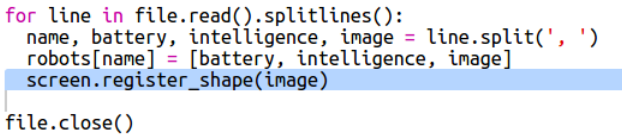


```
if robot in robots:
  stats = robots[robot]
  style = ('Arial', 14, 'bold')
  setheading(-90)
  write('Name: ' + robot, font=style, align='center')
  forward(25)
  write('Battery: ' + stats[0], font=style, align='center')
  forward(25)
  write('Intelligence: ' + stats[1], font=style, align='center')
```

```
Name: space
Battery: 13
Intelligence: 28
```

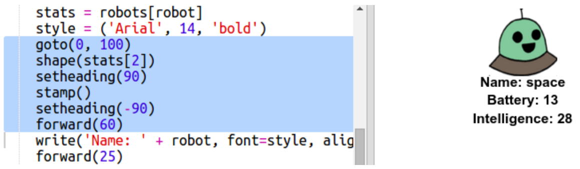
- And finally, let's add the robot image to complete the display.

  You'll need to add a line to register the image when you read the data from `cards.txt`:



```
for line in file.read().splitlines():
  name, battery, intelligence, image = line.split(', ')
  robots[name] = [battery, intelligence, image]
  screen.register_shape(image)

file.close()
```

- And add code to position and stamp the image:

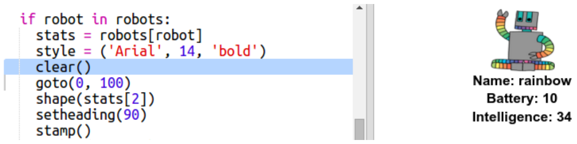


```
  stats = robots[robot]
  style = ('Arial', 14, 'bold')
  goto(0, 100)
  shape(stats[2])
  setheading(90)
  stamp()
  setheading(-90)
  forward(60)
  write('Name: ' + robot, font=style, alig
  forward(25)
```

```
Name: space
Battery: 13
Intelligence: 28
```

- Test your code by entering one robot and then another and you'll see that they display on top of each other!

  You need to clear the screen before displaying a robot:



```
if robot in robots:
  stats = robots[robot]
  style = ('Arial', 14, 'bold')
  clear()
  goto(0, 100)
  shape(stats[2])
  setheading(90)
  stamp()
```

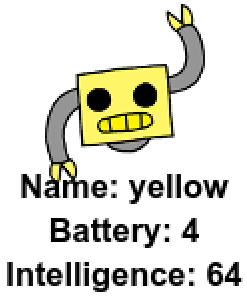
```
Name: rainbow
Battery: 10
Intelligence: 34
```

Step 4 Challenge: Add more robots

Can you add data about more robots to `cards.txt`.

Click on the images button to see the robot images that you can use.

You get to decide how much battery and intelligence they have.



```
Name: yellow
Battery: 4
Intelligence: 64
```

Step 5 Challenge: Add more stats to the robots

Can you think of more stats to add to the robots. You could add ‘speed’ or ‘usefulness’ or come up with your own ideas.

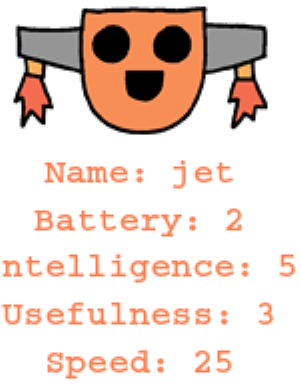
You’ll need to:

- Add data to the file for each new category
- Add the new category to the code that read in the data
- Write the new category out when you display a trump card

You could even add a colour and show the stats for robots in their own color.

Hint: Use `color('red')` to change the turtle’s text to red before writing.

Example:

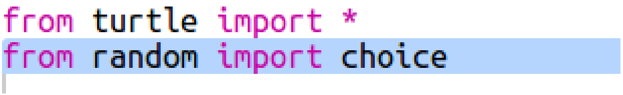


```
Name: jet
Battery: 2
Intelligence: 5
Usefulness: 3
Speed: 25
```

Step 6 Display a random robot

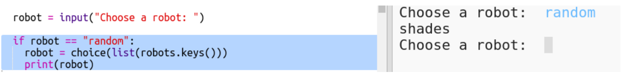
Let’s add code so that you get a random robot when you type Random instead of a robot name.

- First you’ll need to import the choice function from the random module:



```
from turtle import *
from random import choice
```

- You can use `choice` to pick a random robot name from the list of keys from the robot dictionary.



```
robot = input("Choose a robot: ")

if robot == "random":
  robot = choice(list(robots.keys()))
  print(robot)
```

```
Choose a robot:  random
shades
Choose a robot:
```

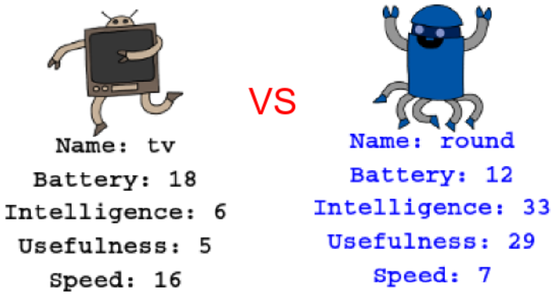
- In Python 3 you need to use `list` to turn the results of `keys` into a list.

  Tip: Make sure you check your brackets carefully!

Step 7 Challenge: Play robot trumps with a friend

Share your project with a friend and play Robotrumps. Both use the same project so that it's fair! Player one requests a random robot and then chooses a category. Player 2 then requests a random robot and you check who has the highest score for the chosen category, then swap over.

The game works best if you both play with the same deck of cards. Share a link to your trinket project with a friend so you can both use the same deck.



```
      Name: tv                 Name: round
    Battery: 18       VS       Battery: 12
 Intelligence: 6           Intelligence: 33
   Usefulness: 5             Usefulness: 29
     Speed: 16                  Speed: 7
```