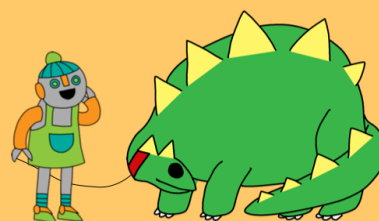# 🍓 Projects

Popular Pets

Create pie charts and bar graphs from data you collect.
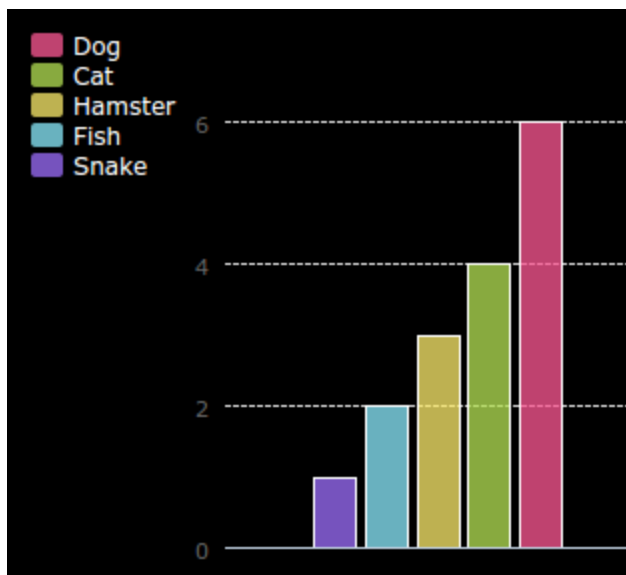
`Python`

Step 1   ## Introduction

In this project you create pie charts and bar graphs from data that you collect from members of your Code Club.

Instructions

When you click Run the charts are generated from data using the PyGal python module.

What you will learn

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://ww w.raspberrypi.org/curriculum/programming/builder/)**

Additional information for educators

If you need to print this project, please use the **printer-friendly version** (h ttps://projects.raspberrypi.org/en/projects/popular-pets/print).

Use the link in the footer to access the GitHub repository for this project, which contains all resources (including an example finished project) in the 'en/resources' folder.

Step 2 — What you will need

Software

- Trinket online editor

Note - this project uses trinket's pygal module and will only work with trinket.

Step 3 — Create a Pie Chart

Pie Charts are are useful way of showing data. Let's do a survey of favourite pets in your Code Club and then present the data as a Pie Chart.

- Ask your club leader to help organise a survey. You could record the results on a computer connected to a projector or a whiteboard that everyone can see.

  Write a list of pets and make sure everyone's favourite is included.

  Then get everyone to vote for their favourite by putting their hand up when it gets called out. Only one vote each!

  For example:

```
Dog 6
Cat 4
Hamster 3
Fish 2
Snake 1
```
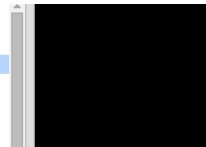
- Open the blank Python template Trinket: **jumpto.cc/python-new (htt p://jumpto.cc/python-new)**.

- Let's create a pie chart to show the results of your survey. You'll be using the PyGal library to do some of the hard work.

  First import the Pygal library:

  ```python
  import pygal
  ```

- Now let's create a Pie chart and render (display) it:

  ```python
  import pygal

  piechart = pygal.Pie()
  piechart.render()
  ```
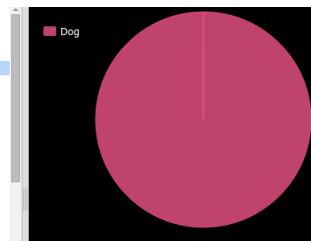
  Don't worry, it gets more interesting when you add data!

- Let's add in the data for one of the pets. Use the data that you collected.

  ```python
  import pygal

  piechart = pygal.Pie()
  piechart.add('Dog', 6)
  piechart.render()
  ```
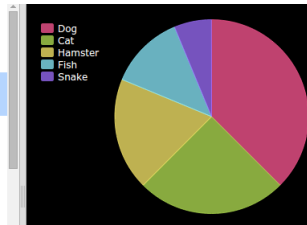
  There's only one piece of data so it takes up the whole pie chart.

- Now add the rest of the data in the same way.

  For example:

```
import pygal

piechart = pygal.Pie()
piechart.add('Dog', 6)
piechart.add('Cat', 4)
piechart.add('Hamster', 3)
piechart.add('Fish', 2)
piechart.add('Snake', 1)
piechart.render()
```

- And to finish off your chart, add a title:

```
piechart = pygal.Pie()
piechart.title = 'Favourite Pets'
piechart.add('Dog', 6)
piechart.add('Cat', 4)
piechart.add('Hamster', 3)
piechart.add('Fish', 2)
piechart.add('Snake', 1)
piechart.render()
```
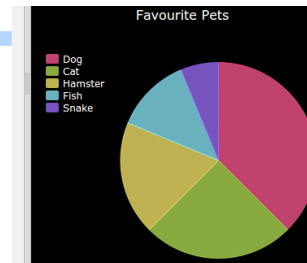
Step 4   Challenge: Create your own bar chart

You can create bar charts in a similar way. Just use `barchart = pygal.Bar()` to create a new barchart, and then add data and render in the same way as for a pie chart.

Collect data from your Code Club members to create your own bar graph.

Make sure that you choose a topic that everyone will know about!

Here are some ideas:

- What is your favourite sport?
- What is your favourite ice cream flavour?
- How do you get to school?
- What month is your birthday?
- Do you play Minecraft? (yes/no)

Don't ask questions that give personal data such as where people live. Ask your club leader if you're not sure.
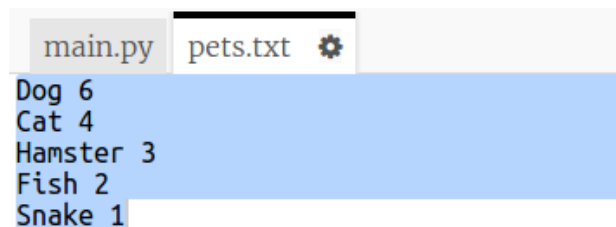
Examples:
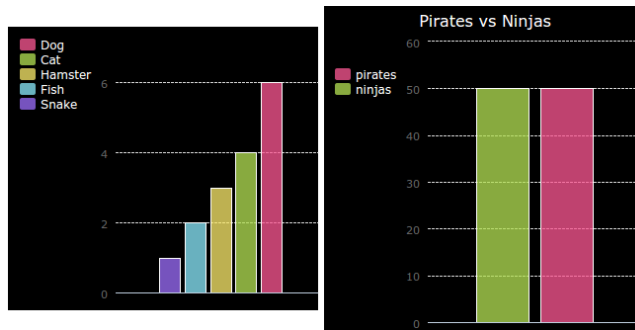
Step 5   Read data from a file

It's useful to be able to store data in a file rather than having to include it in your code.

- Add a new file to your project and call it `pets.txt`:

- Now add data to the file. You can use the favourite pets data that you collected or the example data.

- Switch back to `main.py` and comment out the lines that render (display) charts and graphs (so that they aren't displayed):

```
piechart.add('Snake', 1)
#piechart.render()


barchart = pygal.Bar()
barchart.title = 'Favourite Pets'
barchart.add('Dog', 6)
barchart.add('Cat', 4)
barchart.add('Hamster', 3)
barchart.add('Fish', 2)
barchart.add('Snake', 5)
#barchart.render()
```

- Now let's read the data from the file.

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

for line in file.read().splitlines():
    print(line)

file.close()
```

```
Powered by  trinket
Dog 6
Cat 4
Hamster 3
Fish 2
Snake 1
```

The `for` loop will loop over the lines in the file. `splitlines()` removes the newline character from the end of the line as you don't want that.

- Each line needs to be separated into a label and a value:

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

for line in file.read().splitlines():
    label, value = line.split(' ')
    print(label, value)

file.close()
```

```
Powered by  trinket
Dog 6
Cat 4
Hamster 3
Fish 2
Snake 1
```

This will split the line at the spaces so don't include spaces in the labels. (You can add support for spaces in labels later.)

- You might get an error like this:

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

for line in file.read().splitlines():
    label, value = line.split(' ')
    print(label, value)

file.close()
```

```
ValueError: need more than 1 values to unpack on line 30 in main.py
```

This happens if you have an empty line at the end of your file.

You can fix the error by only getting the label and value if the line isn't empty.

To do this, indent the code inside your `for` loop and add the code `if line:` above it:

```python
file = open('pets.txt', 'r')

for line in file.read().splitlines():
  if line:
    label, value = line.split(' ')
    print(label, value)

file.close()
```

- You can remove the `print(label, value)` line now everything is working.

- Now let's add the label and value to a new Pie Chart and render it:

```python
piechart2 = pygal.Pie()

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

for line in file.read().splitlines():
  if line:
    label, value = line.split(' ')
    piechart2.add(label, int(value))

file.close()

piechart2.render()
```
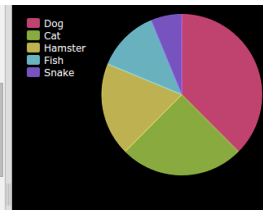

Note that `add` expects the value to be a number, `int(value)` turns the value from a string into an integer.

If you wanted to use decimals such as 3.5 (floating point numbers) you could use `float(value)` instead.

## Step 6   Challenge: Create a new chart from a file

Can you create a new bar graph or pie chart from data in a file? You'll need to create a new .txt file.

Tip: If you want to have spaces in the labels then use `line.split(': ')` and add colons to your data file, e.g. 'Red Admiral: 6'
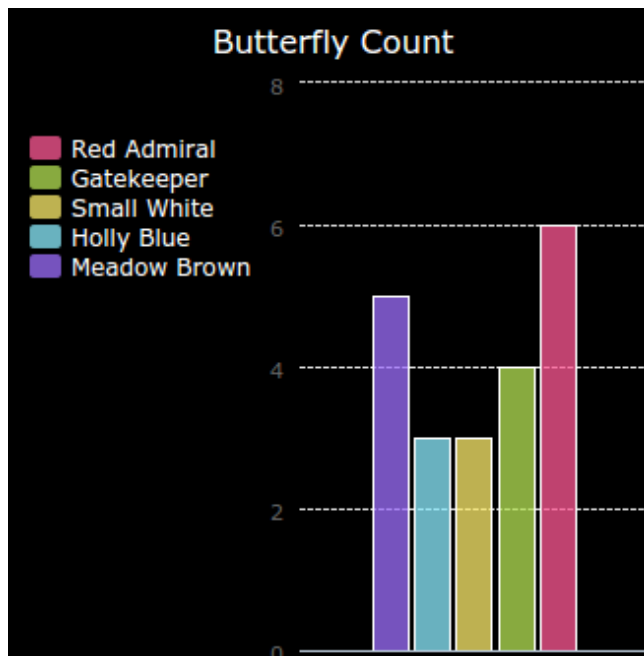
Step 7   Challenge: More charts and graphs!

Can you create a pie chart and a bar chart from the same file? You can either use the data you collected earlier or collect some new data.
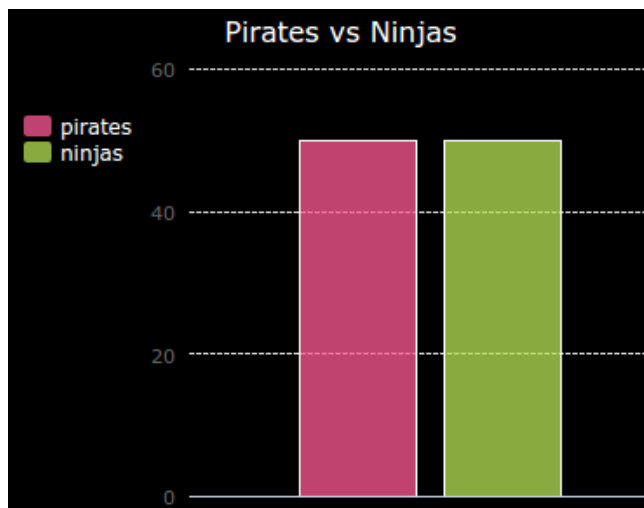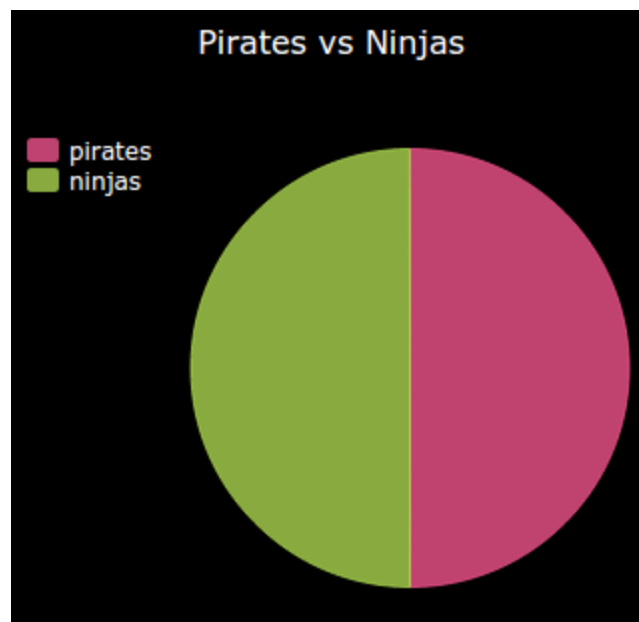
Published by **Raspberry Pi Foundation (https://www.raspberrypi.org)** under a **Creative Commons license (https://creativecommons.org/licenses/by-sa/4.0/)**.
View project & license on GitHub (https://github.com/RaspberryPiLearning/popular-pets)