Projects

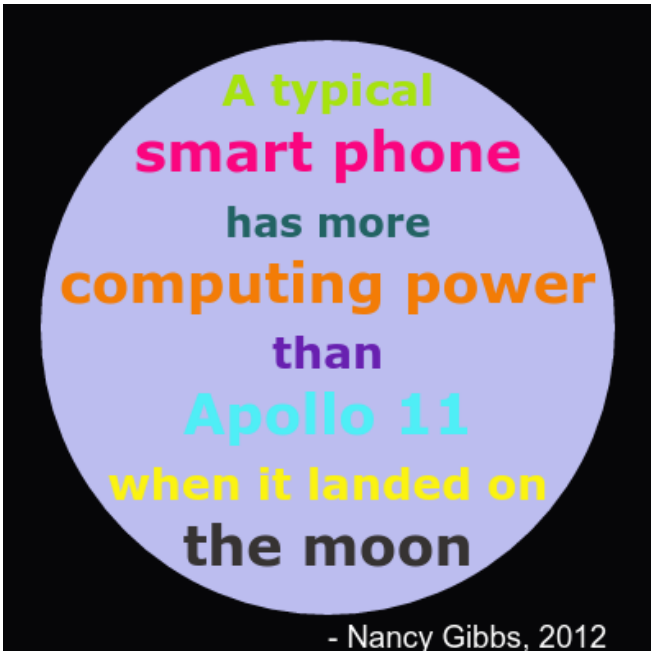
Colourful Creations

Create your own custom colours, and use them to create a colourful poster.

Python

Step 1 Introduction

In this project you will create a dictionary of colours which maps hard to remember colour codes into friendly names.



Additional information for club leaders

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

Club leader notes

Introduction:

This project introduces dictionaries by creating a dictionary that maps from human-friendly colour names to hex codes. The colour codes are then looked up in the dictionary and used to create a colourful poster.

Online Resources

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

- **'Colourful Creations' starting point – jumpto.cc/python-new** (http://jumpto.cc/python-new)

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

- **'Colourful Creations' Finished – trinket.io/python/41a99e668b** (https://trinket.io/python/97822f48b7)

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:

- colourful-creations/colourful-creations.py

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

- colourful-creations-finished/colourful-creations.py

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

Learning Objectives

- Dictionaries - creating and looking up values;
- Turtle graphics - text, fonts and colours;

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)
- **Design basic 2D and 3D assets.** (https://www.raspberrypi.org/curriculum/design/creator)

Challenges

- More colours! - Use a colour picker website to find hexcodes for more colours and add them to a dictionary.
- Create a poster - Create a custom colour palette dictionary and use it to create a poster using Turtle graphics.

Frequently Asked Questions

- Children may need reminding about the comma ‘,’ at the end of each dictionary entry.

Project materials

Project resources

- **.zip file containing all project resources** (https://projects-static.raspberrypi.org/projects/colourful-creations/3d38cce4fa1e787d441b2c144a36d5b1eb3d5dfd/en/resources/colourful-creations-project-resources.zip)
- **Online blank Python Trinket** (http://jumpto.cc/python-new)
- **Offline blank Python file** (https://projects-static.raspberrypi.org/projects/colourful-creations/3d38cce4fa1e787d441b2c144a36d5b1eb3d5dfd/en/resources/new-new.py)

Club leader resources

- **.zip file containing all completed project resources** (https://projects-static.raspberrypi.org/projects/colourful-creations/3d38c

ce4fa1e787d441b2c144a36d5b1eb3d5dfd/en/resources/colourful-creations-volunteer-resources.zip)

- **Online completed Trinket project** (https://trinket.io/python/97822f48b7)
- **colourful-creations-finished/colourful-creations.py** (https://projects-static.raspberrypi.org/projects/colourful-creations/3d38cce4fa1e787d441b2c144a36d5b1eb3d5dfd/en/resources/colourful-creations-finished-colourful-creations.py)

Step 2 Using hex colour codes

Python turtle has predefined colours such as 'red' and 'white' but you can also use hex colour codes (you may have seen these in the HTML & CSS course.)

- Open the blank Python template Trinket: **jumpto.cc/python-new** (http://jumpto.cc/python-new).
- Add the following set up code for using the turtle:

```
from turtle import *

screen = Screen()
screen.setup(400, 400)
screen.bgcolor('white')
```

  Notice that you used a named colour: 'white'.

- Turtle has a list of colour names that you can use, but sometimes you want to choose your own colours. Turtle also allows you to use hex colour codes.

  Open **jumpto.cc/colour-picker** (http://jumpto.cc/colour-picker) and choose colour you like. Find it's hex code beginning with a '#', such as '#A7E30E'.

- Copy the hex code, including the hash, by highlighting it and then right-clicking and choosing Copy, or using Ctrl-C.

- Now change the line of code that sets the screen colour to use your colour. For example:

```
from turtle import *

screen = Screen()
screen.setup(400, 400)
screen.bgcolor('#A7E30E')
```

You can use right-click and Paste or Ctrl-V to paste your hex code into trinket.

- Choose another hex colour code and use it to create coloured text:

```
screen = Screen()
screen.setup(400, 400)
screen.bgcolor('#A7E30E')

color('#FA057F')
style = ('Arial', 40, 'bold')
write('HELLO', font=style, align='center')
hideturtle()
```

You don't have to use the 'Arial' font, you could try 'Verdana', 'Times' or 'Courier'.

'40' is the font size, you can try changing that too.

- Try different colours until you get two that you really like that look good together.

Step 3 A Colour Dictionary

Using hex colour codes is really flexible but they are hard to remember.

As you probably already know, a dictionary allows you to look up a word, and see it's meaning. In Python, a dictionary is even more flexible that that - it allows you to look up a value for any 'key' in the dictionary.

Let's create a dictionary to map from human-friendly colour names (keys) to computer-friendly hex codes (values).

- A dictionary is contained in curly brackets.

  Create an empty dictionary called `colours`:

```
screen = Screen()
screen.setup(400, 400)

colours = { }

screen.bgcolor('#A7E30E')
```

- Choose cool names for your colours and edit the `colours =` line to add entries to the dictionary for them.

  Here's an example colour dictionary:

```
colours = {
  'verylime': '#A7E30E',
  'reallyraspberry': '#FA057F'
}
```

  A colon `:` separates the key (colour name) from the value (hex code.) You need a comma `,` between each key:value pair in the dictionary.

- Now you don't need to remember the hex codes, you can just look them up in the dictionary.

  Adapt the following code to use your colour names:

```
colours = {
  'verylime': '#A7E30E',
  'reallyraspberry': '#FA057F'
}

print(colours['verylime'])
print(colours['reallyraspberry'])
```

```
Powered by trinket
#A7E30E
#FA057F
```

  The key goes inside square brackets '[]' after the name of the dictionary.

- Now you can update your code to look up colours in the dictionary:

```
screen.bgcolor(colours['verylime'])

color(colours['reallyraspberry'])
style = ('Arial', 40, 'bold')
write('HELLO', font=style, align='center')
hideturtle()
```

- Test your code to make sure your text still displays correctly.

Step 4 Challenge: More colours!

Can you add more colours to your dictionary and try them out? Use **jumpto.cc/colour-picker** (http://jumpto.cc/colour-picker) to find more colours.

Don't forget to give your colours awesome names.

Here's some example code to remind you how to use the turtle:

```
penup()
goto(0, 100)
color(colours['reallyraspberry'])
style = ('Arial', 40, 'bold')
write('HELLO', font=style, align='center')
right(90)
forward(60)
color(colours['awesomeorange'])
write('WORLD', font=style, align='center')
hideturtle()
```

Step 5 Challenge: Create a poster

Designers often create a 'palette' of colours that work well together for a particular theme such as desert or space.

Can you create a new Python project that uses a dictionary for a themed colour palette. You could choose autumn, forest, sea, Christmas, ice cream, the colours of your favourite sports team or an idea of your own.

Create a poster using your colour palette dictionary.

You can also use other turtle commands that you know such as `forward`, `right`, `left`, `penup` and `pendown`.

Maybe you could add a border to your poster?

Other useful turtle commands:

- `circle(50)` draws a circle outline with radius 50.
- `dot(100)` draws a filled in circle with diameter 100.

Here's an example: