

Introduction

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



Step 1: Using hex colour codes

✓ Activity Checklist

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: jump.to/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 jump.to/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.

Save Your Project

Step 2: A Colour Dictionary

✓ Activity Checklist

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 than 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.

Save Your Project

Challenge: More colours!

Can you add more colours to your dictionary and try them out? Use 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()
```



Save Your Project

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:



[Save Your Project](#)