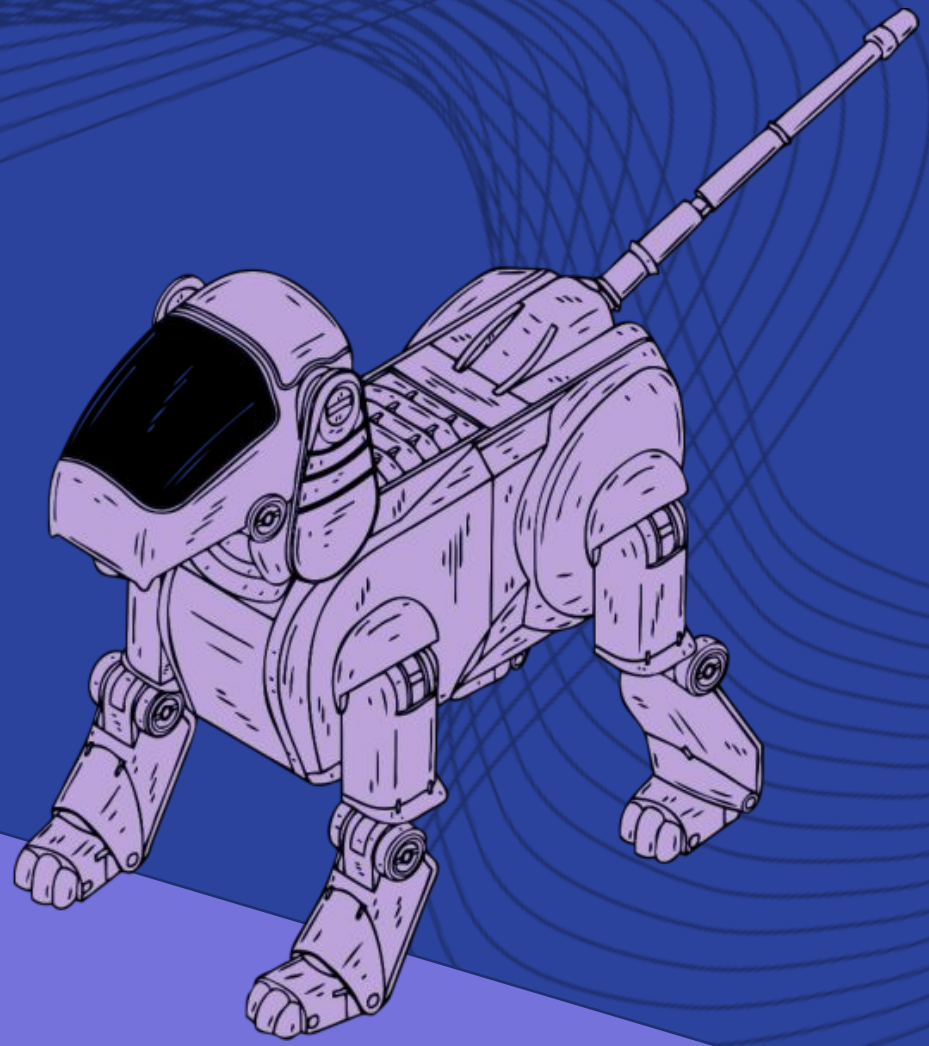
Tessellations

Level 3 – Python

Arts and Creativity



cair 4 YOUTH

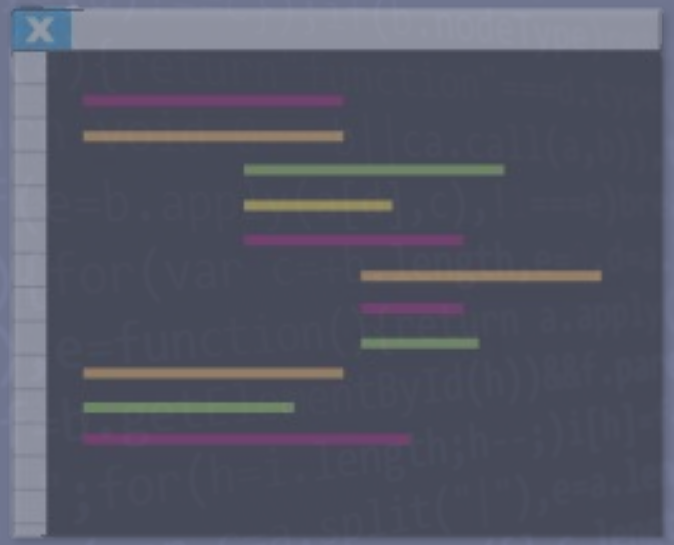
Introduction to Coding

What is Python?

- Python is a popular general-purpose programming language that can be used for a wide variety of applications.
- Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high-level dynamic data types, and classes.

Where to access Python?

- https://www.python.org/downloads/ - downloadable app for PCs (allows you to save files directly onto a computer)
- https://trinket.io/ - online version (allows you to create an account, much like scratch)



cair
4 YOUTH

Introduction

- A beehive is a structure in which honey bees live and raise their young. Natural beehives (or "nests") are made by honey bee colonies,
  while domesticated honey bees are kept
  in man-made beehives in a location known as an apiary.
- Beehives have hexagonal cells in them, packed together. This is called a honeycomb. This honeycomb is said to have a tessellated shape formation.

cair
4 YOUTH

Task

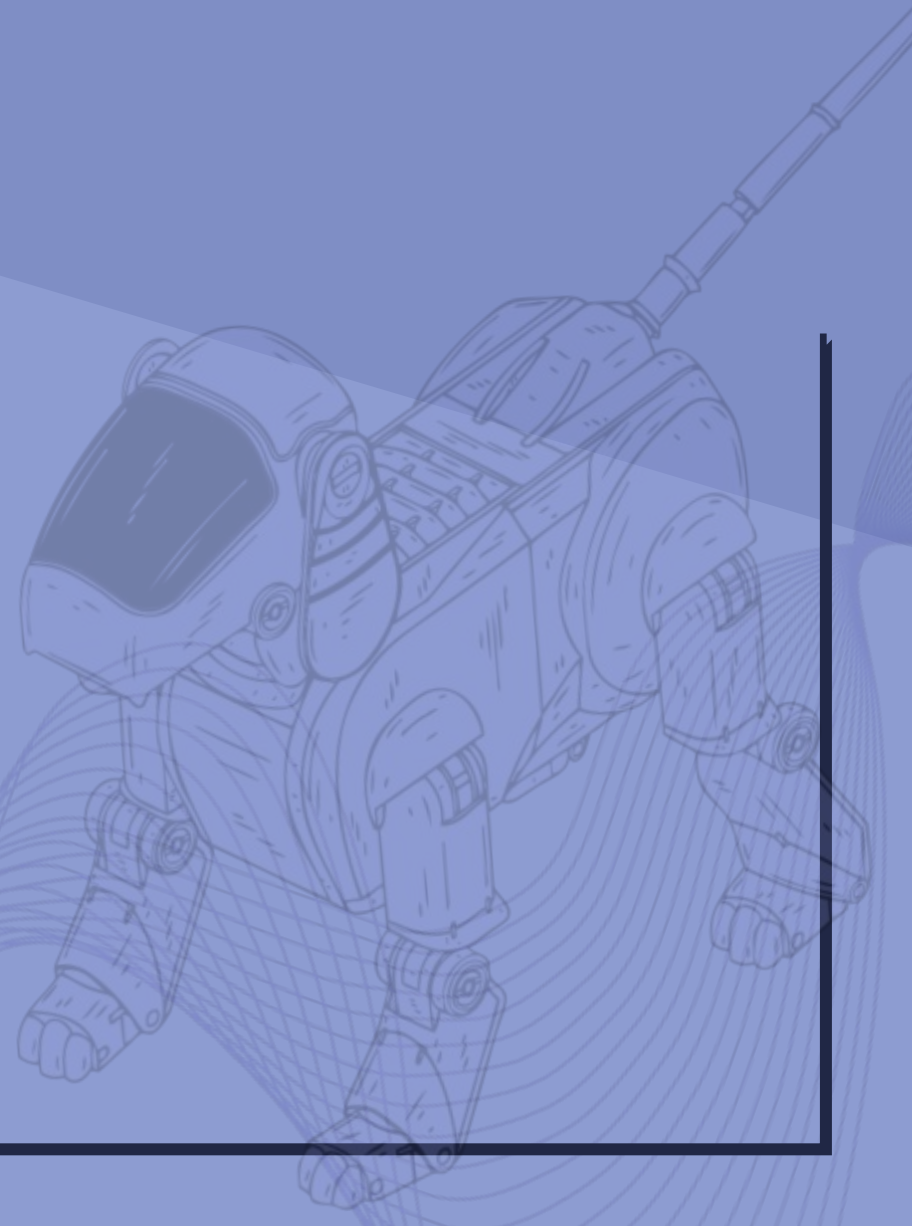
Beehive

- Your task today is to build a small section of a beehive.
- This drawing will be created using a library in Python called Turtle graphics, in which we will use different angles to fit together with our honeycomb shape!

Process

Your code should:

- ✔ Import the Turtle graphic library,
- ✔ Control the speed, fill colour and line colour of the turtle,
- ✔ Use for loops and subroutines to create the hexagonal shapes,
- ✔ Apply previous knowledge of angles to help move the turtle along.



cair
4 YOUTH

How does turtle graphics work?

- Python is a high-level programming language which allows its users to use different programming styles to create simple or complex programs.
- It is designed to be easy to read and simple to implement.

- Python libraries are a set of useful functions that eliminate the need for writing codes from scratch.
- The library we will be importing is ‘Turtle’ which allows the user to create graphic shapes.
- The turtle understands simple commands such as left and right turns, moving forwards and moving backwards.

Simple Turtle commands:

- fd(10) – moves the turtle forward 10 pixels,
- backward(5) – moves the turtle backward by 5 pixels,
- right(35) – moves the turtle clockwise by an angle of 35 degrees,
- left(55) – moves the turtle counter-clockwise by an angle of 55 degrees;
- goto(x, y) – moves the turtle to the position x, y.

Step 1

Importing turtle

In the code here we have imported the turtle library and changed the shape of the cursor to look like a turtle too!

Also, by using a delay we can control the speed of the turtle to make it go as fast, or as slow, as we want. The last part of this code sets the colours for the outline and fill with the shapes we will create.

```
1  #Beehive and tesselations
2  from turtle import *
3  #changes cursor shape
4  shape("turtle")
5  #Changes the speed of the turtle
6  delay(0)
7  #Subroutine to build the hexagon
8  pencolor("orange")
9  #Changes the outline of the shapes
10 fillcolor("gold")
11 begin_fill()
12 #Changes the fill colour of the shape
```

cair
4 YOUTH

Step 2

Creating a subroutine

To create the shape itself subroutines and loops have been used. The use of the subroutine means that once the code has been written at the top it can simply be called multiple times in the program with just the heading name. In addition, the for loops are used to reduce the amount of repetitive code that is written since a hexagon is a regular shape.

Here the two subroutines draw hexagons that are the same size but hex1() draws a group of three hexagons whereas hex2() draws a hexagon but repeats the round two extra times so that we can go back in the pattern as we go through the code.

```
13 #Creates a subroutine to draw the shape in one place
14 def hex1():
15     for loop in range(3):
16         for loop in range(6):
17             fd(25)
18             rt(60)
19         lt(120)
20 def hex2():
21     for loop in range(8):
22         fd(25)
23         rt(60)
```

Step 3

Creating subroutines

These are another two subroutines called hex3 () and hex4().

They both draw hexagons that are the same size as the previous ones except one takes left turns and the other takes right turns to complete the shape. Also, hex4() specifically goes around 10 the shape 10 times so that the angle will fit when the main code is written.

```
24 def hex3():
25     for loop in range(8):
26         fd(25)
27         lt(60)
28 def hex4():
29     for loop in range(10):
30         fd(25)
31         rt(60)
32
```

Step 4

Main code

This is where the main code for the honeycomb picture will be done. So far, the for loop draws the top line of shapes but by using this loop we can avoid repeating code in the program. Furthermore, the subroutines have been called in to the main code to be displayed once the code is run.

Another command turtle uses is pu() and pd(), which stands for pen up and pen down. This means that the turtle can move without leaving a line behind it.

```
33 #main code
34 for loop in range(3):
35     hex1()
36     pu()
37     fd(25)
38     lt(60)
39     pd()
40     hex2()
41 hex1()
42 pu()
43 rt(60)
44 fd(75)
45 lt(60)
46 fd(25)
47 lt(180)
48 pd()
49 hex2()
50 pu()
51 fd(25)
52 lt(60)
53 pd()
```

Step 5

Moving the turtle

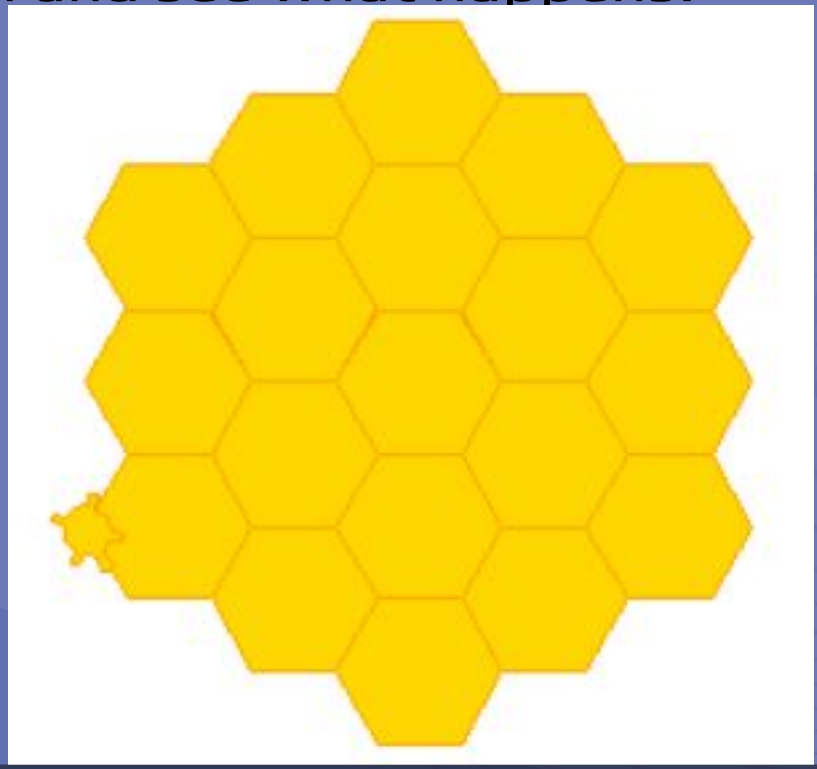
Following on from the last slide, we again are calling subroutines to fit into this tessellated program. The program used a lot of commands between the pu() and pd() statements as the turtle must move through many different lines before getting to the point where another hexagon is needed.

The last line of the program is the end fill button which tells the program to stop the fill colour we coded at the start of the programme.

```
54 hex2()
55 pu()
56 lt(120)
57 pd()
58 hex4()
59 for loop in range(2):
60     hex3()
61     rt(120)
62 lt(120)
63 pu()
64 fd(25)
65 rt(60)
66 fd(25)
67 lt(60)
68 fd(25)
69 rt(180)
70 pd()
71 hex2()
72 fd(25)
73 rt(60)
74 hex3()
75 end_fill()
```

What our graphic should look like

Run the finished program and see what happens!



Extension:

- Once you have completed this project, try experimenting with the colours and see if you can make the outline thicker on the design.

Links to everyday life

Nature

This project was based around the production of honey from a bee hive, which is populated with many busy little bees just like you have been!

Creativity

This project makes you think deeply about just how much art can actually be incorporated into technology which we see all around us every single day.

Design

You can use this project as a stepping stone by experimenting and designing new graphics for any occasion!

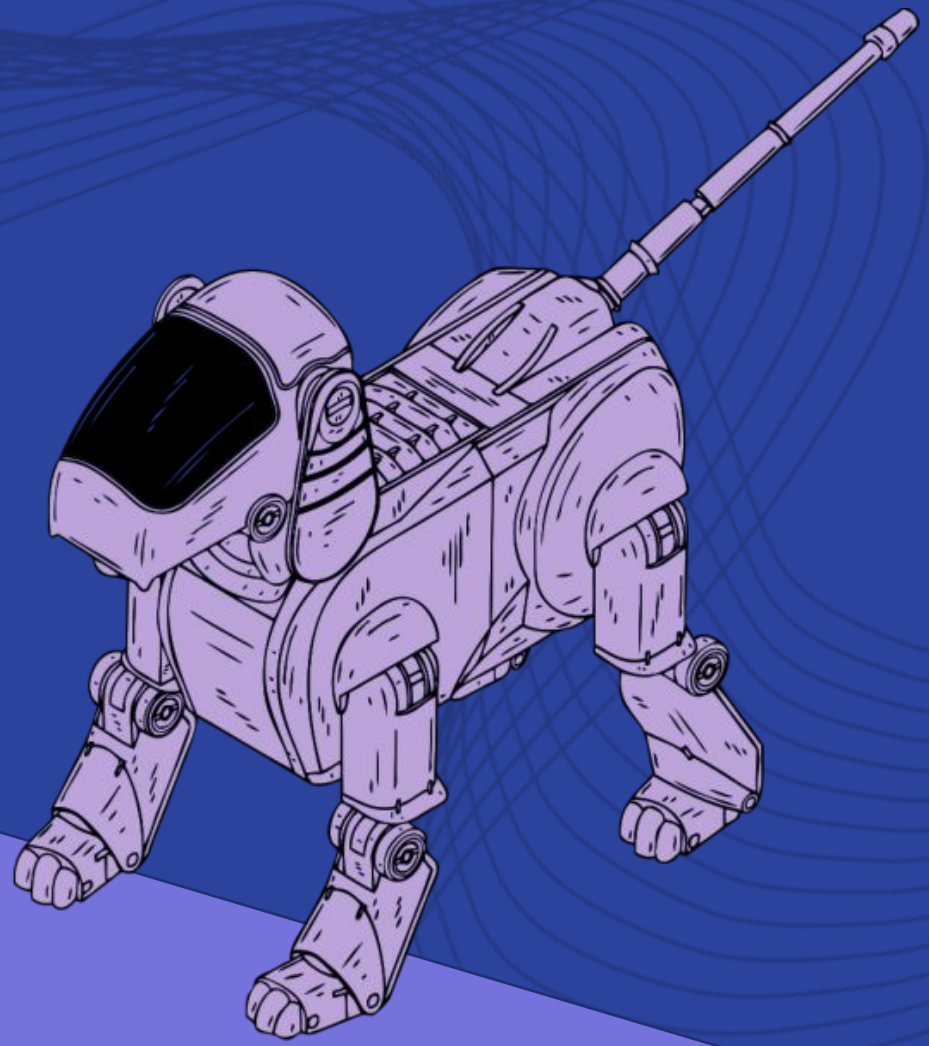
Conclusion

Learning outcomes:

- ✔ You should have confidently been able to import a library into Python,
- ✔ You should be confident in using subroutines to create regular shapes.
- ✔ You should be comfortable using angles to navigate through regular shapes,
- ✔ You should be confident in manipulating the speed and colour within a turtle graphics programme.

Congratulations!

You have completed tessellating hexagons



cair
4 YOUTH