Fireworks & Bonfire Python

We're going to use a programme called Thonny, a package called Turtle, and some loops and random numbers to draw some fireworks! First, make sure you're connected to the internet, then open Thonny, and lets go!

When you're coding, things like capital and lowercase letters, using the tab key to indent lines, and what order your lines of code are in are important so keep an eye out! Don't worry if you make a mistake though, it's easily fixed!

Make 1 firework

1. First let's get 1 firework working, type the following lines into Thonny, then press the play button:

```
1  import turtle
2
3  bob = turtle.Turtle()
4  bob.forward(100)
```

What we've done here is load a package of extra code called turtle in the background, we've created a new turtle object, we've stored it in a variable called 'bob' and then we've told bob to move forward 100 steps.

2. A window should have popped up, and 'bob' should have drawn a line across the screen. To make our firework, we need bob to go backwards, turn a bit, then draw another line!

```
1  import turtle
2
3  bob = turtle.Turtle()
4  bob.forward(100)
5  bob.backward(100)
6  bob.left(10)
```

Now bob will walk forward, draw his line, walk backwards, and turn slightly to the left

3. We need bob to keep repeating these steps, but we don't want to type these instructions out loads of times! We can use a loop to get the computer to do this for us. Copy these instructions, and use the tab key to indent lines that tell bob to go forwards, backwards and left

```
1  import turtle
2
3  bob = turtle.Turtle()
4
5  for i in range(36):
6      bob.forward(100)
7      bob.backward(100)
8      bob.left(10)
```

You've made a loop! You should see bob slowly go around making 36 lines on the screen that look a little bit like a firework.

If for some reason the window that pops up crashes, just click the red stop button in Thonny.

4. Bob is a bit slow, we can speed him up by adding the line bob.speed(100)

```
4  bob.speed(100)
```

Make the firework appear in a random place

5. We need to import some more code, this time a package called random

```
2  import random
```

6. Now we're going to create 2 random numbers, between -250 and 250, and we're going to store them in variables called x and y:

```
7  x = random.randint(-250,250)
8  y = random.randint(-250,250)
```

7. Now we're going to tell bob to go to a position on the screen, but we're going to use those variables, x and y, and because they are storing random numbers for us, they will pass in those random numbers to bob, and he'll start somewhere randomly on the screen!

```
10 bob.goto(x,y)
```

8. You might have noticed, bob is still drawing a path on the screen, we need him to turn the pen off when he's moving between fireworks, we do it like this:

```
10 bob.penup()
11 bob.goto(x,y)
12 bob.pendown()
```

This instruction tells bob to turn his pen off, move to a new location, then put his pen back down again.

Make the firework a random size

9. Now that we know how to create random numbers, we're going to use the same technique to make our fireworks different sizes!

```
7  x = random.randint(-250,250)
8  y = random.randint(-250,250)
9  size = random.randint(10,200)
```

We've created a new variable called size, and size is allocated a number between 10 and 200

```
15 for i in range(36):
16     bob.forward(size)
17     bob.backward(size)
18     bob.left(10)
```

We take that variable size, and pass it into the bob.forward and bob.backward function. This means he'll walk forwards and backwards by a random number.

Make the firework a random colour

10. To make our fireworks random colours, we're going to use a random number generator again!

```
11 r = random.randint(0,255)
12 g = random.randint(0,255)
13 b = random.randint(0,255)
```

One of the ways computers store colours is to use the RGB colour system. 3 different numbers between 0 and 255, combined together, represent a colour. White is 255, 255, 255, black is 0,0,0, and all other colours are in between.

```
15 bob.color(r,g,b)
16 bob.penup()
17 bob.goto(x,y)
18 bob.pendown()
```

In coding, we use the American spelling, color. Here we pass in the random numbers we assigned to r, g and b to the function that sets the pen colour

```
5 turtle.colormode(255)
```

This is quite a random instruction, but is needed to make sure the code works properly.

Make the background black

11. Here we create a new variable called screen, and we set the background colour to black

```
6 screen = turtle.Screen()
7 screen.bgcolor("black")
```

Use a loop to create multiple fireworks!

12. We're going to use one last loop!

```
10 for i in range(10):
11
12     x = random.randint(-250,250)
13     y = random.randint(-250,250)
14     size = random.randint(10,200)
15
16     r = random.randint(0,255)
17     g = random.randint(0,255)
18     b = random.randint(0,255)
19
20     bob.color(r,g,b)
21     bob.penup()
22     bob.goto(x,y)
23     bob.pendown()
24
25     for i in range(36):
26         bob.forward(size)
27         bob.backward(size)
28         bob.left(10)
```

Add in the line **for i in range(10):** before the lines which set up the x and y variable.

Then select all of the lines underneath them, and press the tab key so they are indented.

The first loop you wrote, is inside this new loop, so the lines bob.forward, bob.backward and bob.left should be double indented

Congratulations!! You've done it!! You've coded a python programme which imports other packages, and you've used variables, loops and random numbers to generate a fireworks animation! Why don't you try tweaking your programme to see what happens:

- Experiment by changing how far bob turns, and how many lines he draws
- Try changing the random number limits, if something breaks can you fix it?
- Can you pick a new name for your turtle and make the required changes in all of your code?
- Can you get the turtle to make different shapes?