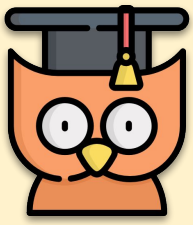
[the academy_of_code]

Ranelagh MDS
Python Programming

www.theacademyofcode.com/handouts

Lesson 1 - Python Intro



Learning Outcomes:

- Small Motor Skills Development
- Introduction to Python

Game Time - Typing Skills

We've had an awesome 6 weeks with HTML and CSS. We're about to start into a module on Python which will involve a bit of typing. Lets polish up on our typing skills. Today's three typing activities are:

- BBC Dancemat
  - Level 1
  - Level 2
  - Level 3
- Keyboard Climber
- Typing Attack

CTRL + Click

Game Time - An Introduction to Python

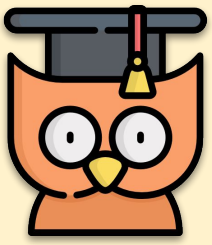
For the remainder of the class, we're going to use python to battle through the dungeon in **Code Combat.**

You can play at this link

www.codecombat.com/play/dungeon

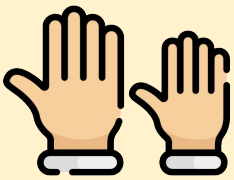
CTRL + Click

Lesson 2 - Block Turtle I

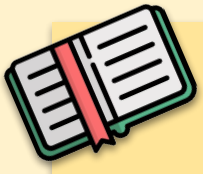


Learning Outcomes:

- Meeting and talking about Tina
- Making Tina draw some simple shapes in Block Turtle
- Working out different locations for Tina to get to on a map.
- Drawing multiple circles.



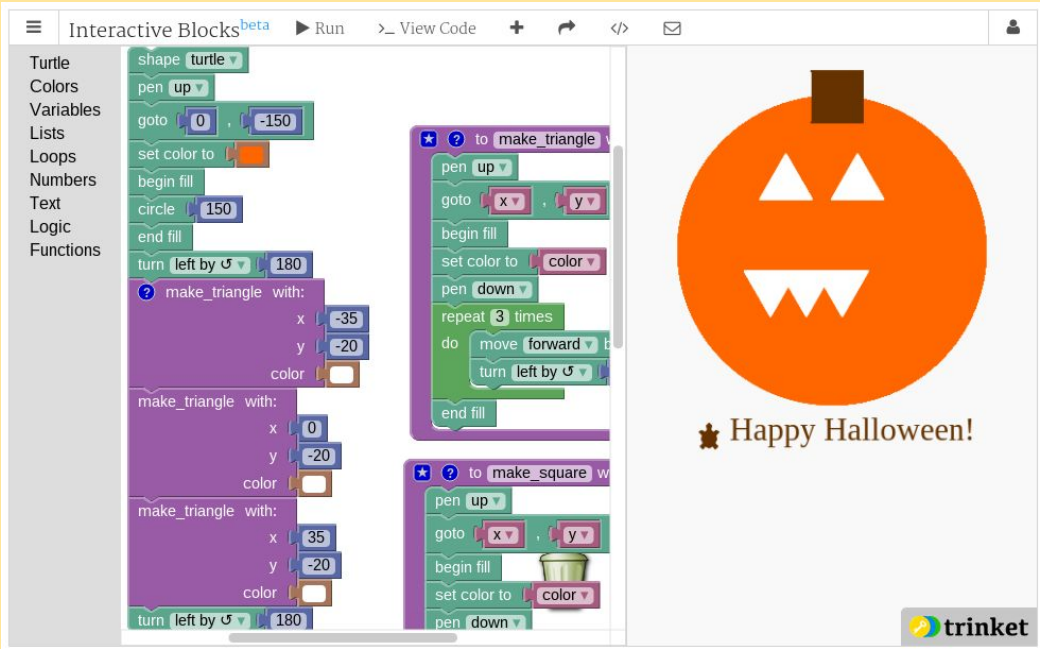
REMEMBER: Put up your hand. We love to help!



Block Turtle

Today we're going to get started with coding. We will be creating images and graphics in Python Blocks with Tina the Turtle.

Tina the turtle moves around your screen just like the mouse arrow, but unlike the mouse, Tina leaves a trail behind her. Tina can draw all kinds of shapes and pictures - you just need to give them the right commands.

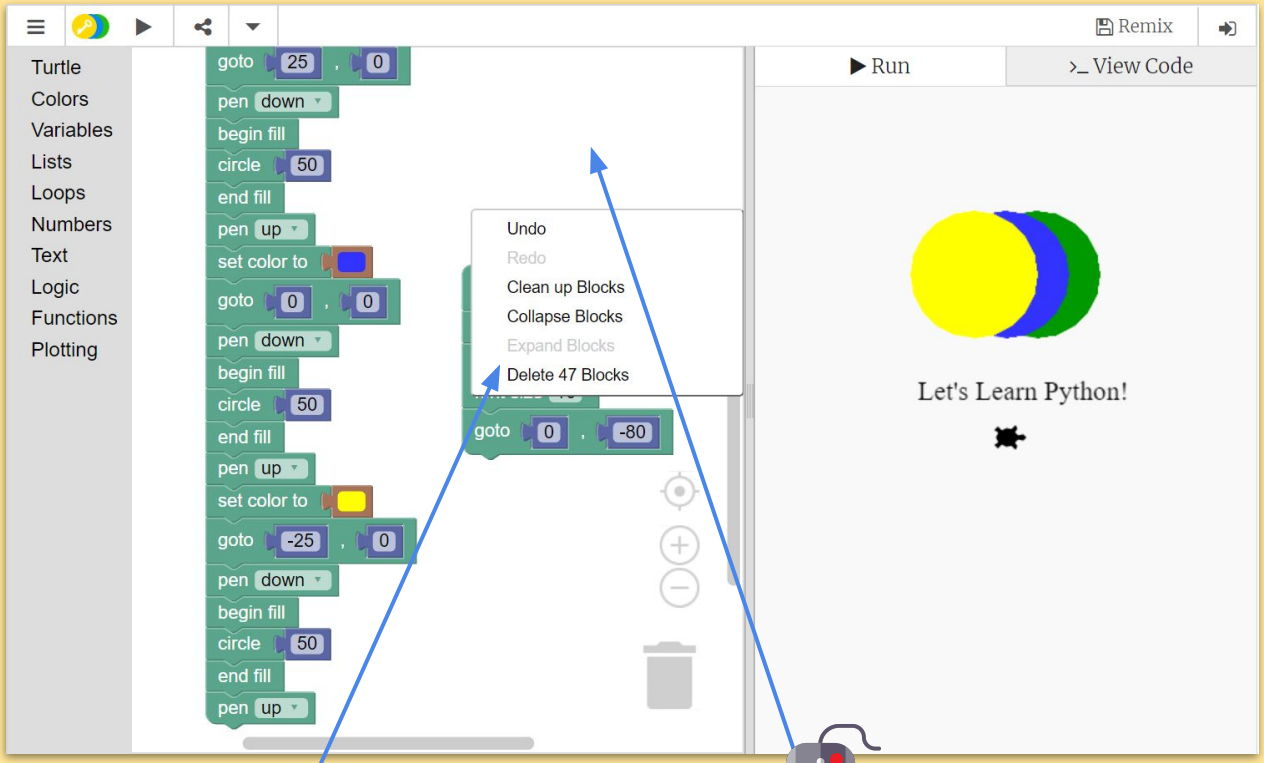


A very advanced drawing example. Can you see Tina?

Let's Get Coding

1 Open Trinket, if it isn't already open. **CTRL** + **CLICK**

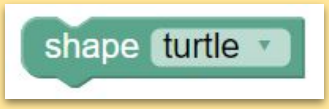
www.bit.ly/blankblocktrinket



2 If there are blocks already here, Right Click on this white area and Left Click on 'Delete Blocks' to clean up your **workspace**.

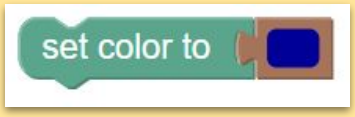
Now we're ready to code. We are going to draw a simple shape (a circle) using Tina the Turtle.

3 Set the style of the turtle with a function called shape.



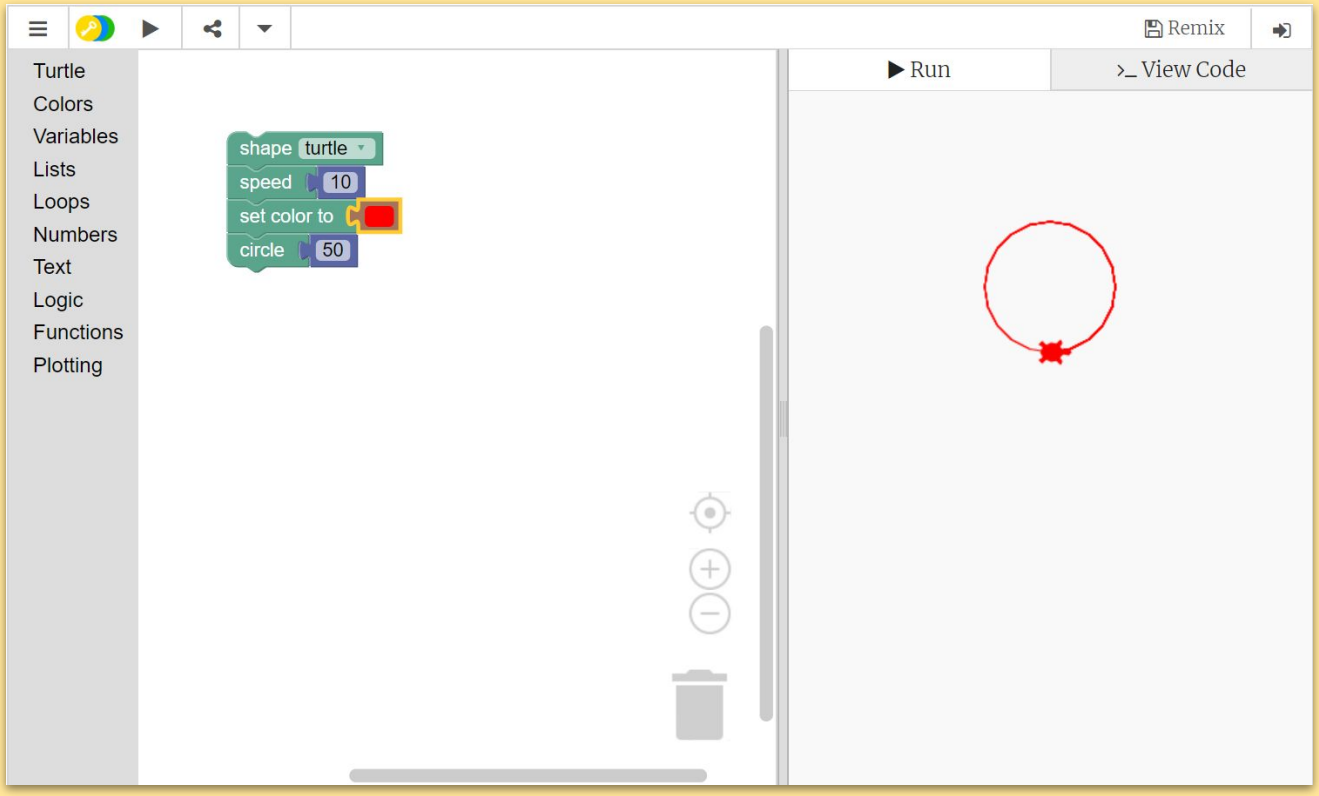
4 Set the turtle's speed, choosing a number between 1 and 10 (10 is the fastest).

5 Set the colour that the turtle will use to draw the shape.

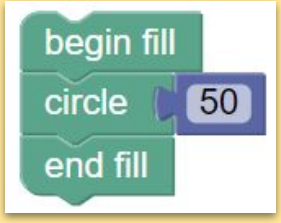


6 Now, tell your turtle to draw a circle.

7 Run the code above. You should get a result like this:



8 Fill the inside of the circle with a colour (same colour as the outline) using the following block.



Challenge

If you've finished this tasks, play around with the other blocks for a few minutes to see what else you can do with Tina.

See if you can:

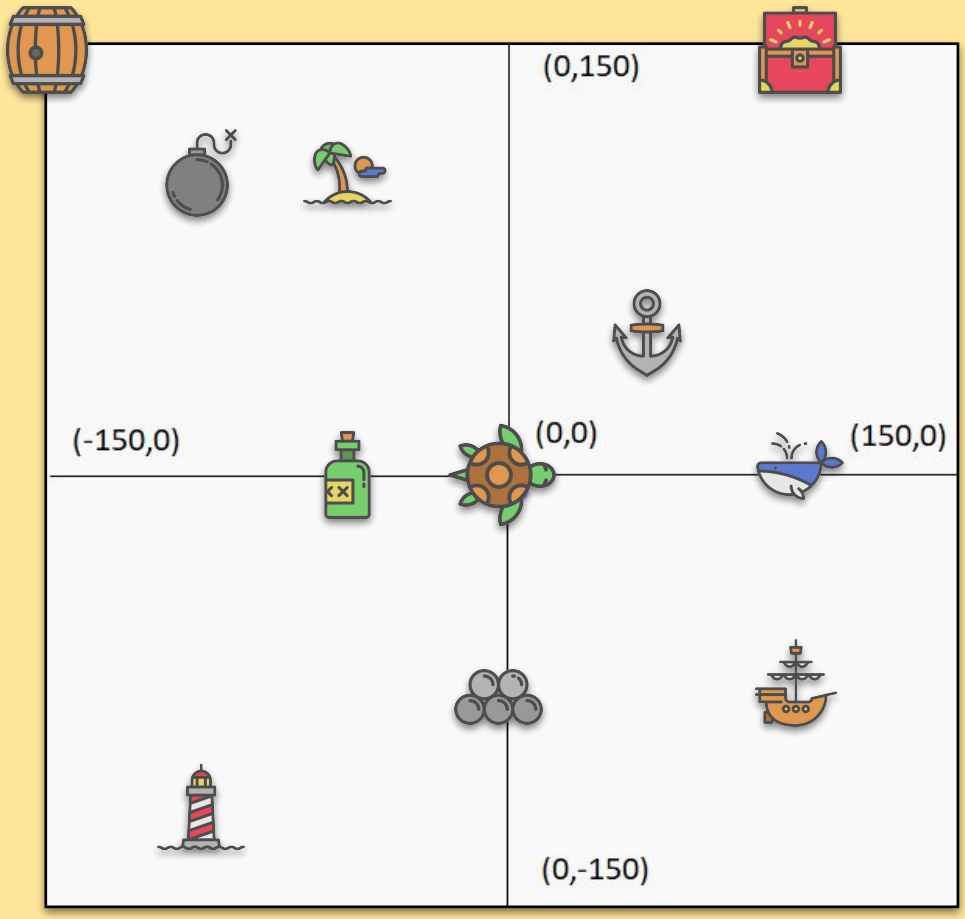
- Make Tina draw **another circle**
    - in a different colour
    - In a different position
- Make Tina **write some text**

Drawing Location - coordinates

Sometimes we may wish to move the turtle before we start drawing something so we can draw shapes in different locations on the screen.

Drawing Location - Treasure Hunt

As a class, try and guess the approximate **coordinates** of each of the items of treasure for Tina.



Barrel of Oil
(-150,_____)

Bottle of Poison
(-50,_____)

Whale
(100,_____)

Lighthouse
(-100,_____)

Ship
(____, -75)

Desert Island
(____,____)

Cannonballs
(____,____)

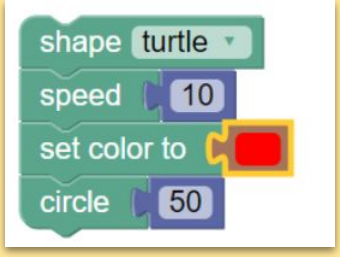
Bomb
(____,____)

Treasure
(____, 150)

Anchor
(____,____)

Task 1 - Drawing Two Circles

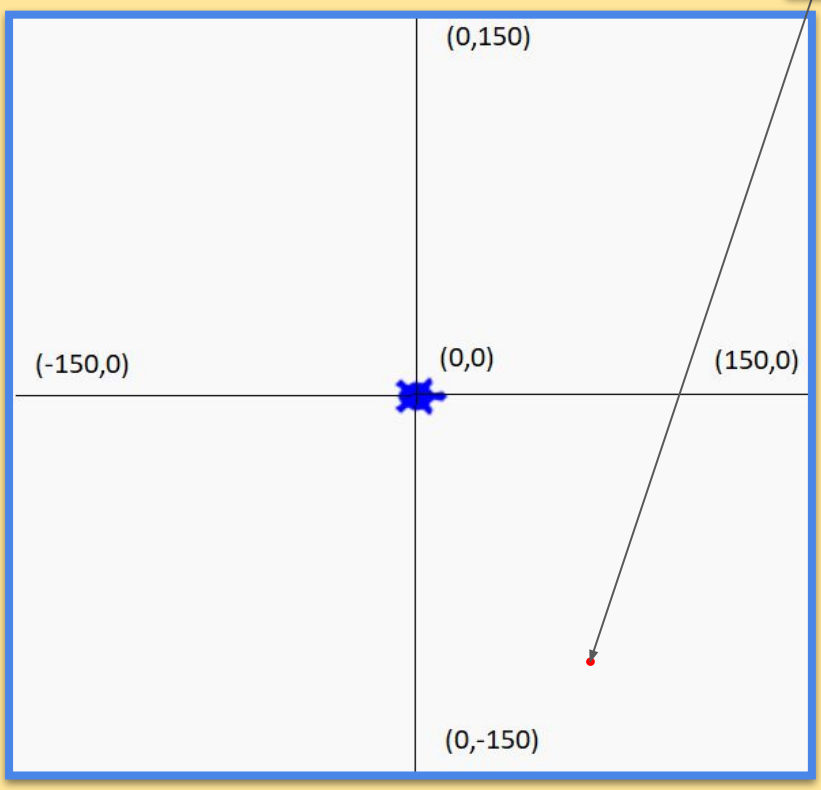
1. Set up Tina the Turtle as before being sure to include **shape, speed, colour** and **circle.**



2. Now we want to move her to a different location. We need to use **Pen up** to make sure Tina does not draw lines while moving to a different location.

3. Tell Tina to go to a particular point on the screen by telling her the **x-coordinate** (horizontal/ left right) and the **y-coordinate** (vertical/up-down)

(70,-100) is here

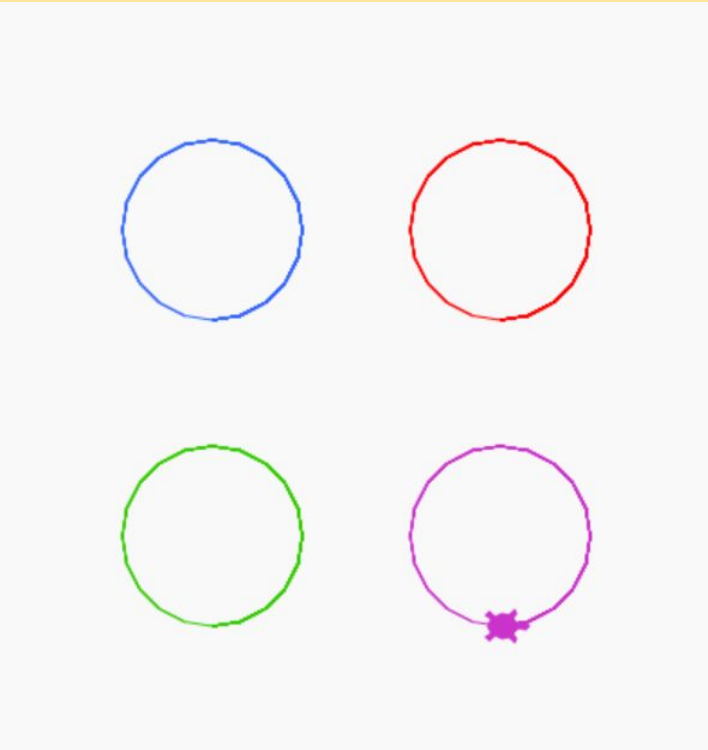


4. Put the pen back down on the screen using **Pen Down** so that Tina can start drawing again.

5. Start drawing your shape.

Task 2 - Drawing Four Circles

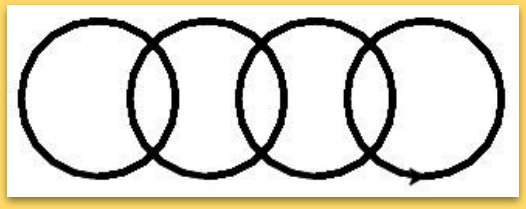
Finish the code on the right so that it makes four circles similar to the drawing below.

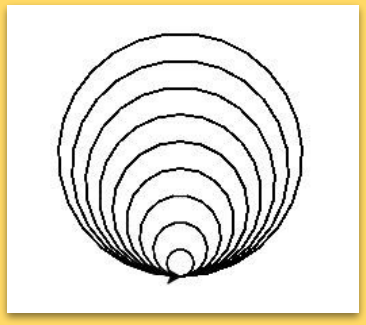


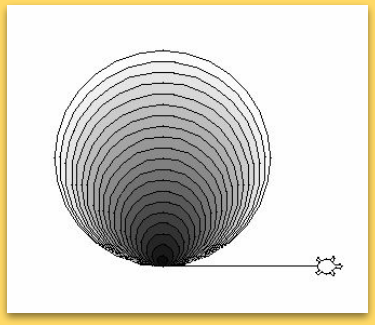
```
shape turtle
speed 10
set color to
pen up
goto -80 , 20
pen down
circle 50
set color to
pen up
goto 80 , 20
pen down
circle 50
set color to
pen up
```

Challenge

Try making the image these images.

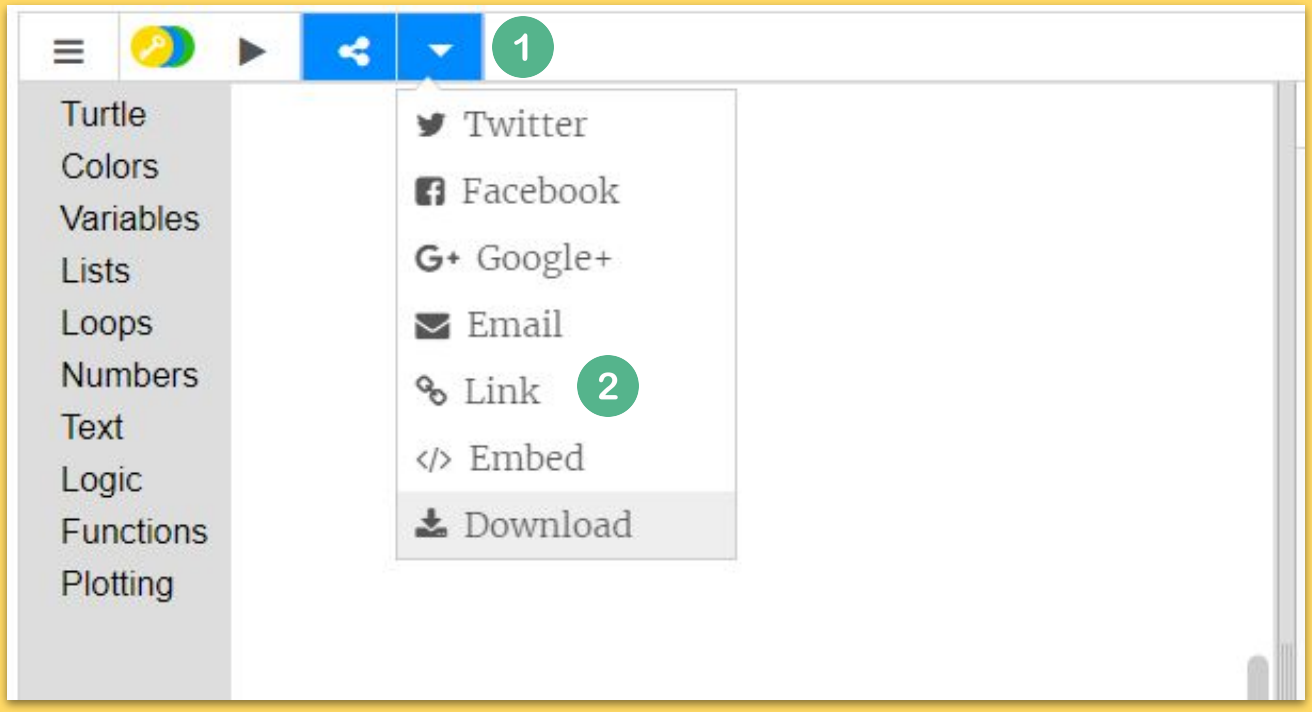






Now it's time to Save our Work

Make sure you do this properly, **you'll need it next week!**



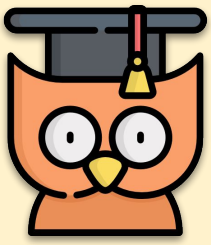
To save your Python project:

1. Open the drop down menu.
2. Click **Link**
3. Paste the Link into Notepad and save to your USB
   If you're not sure to do any of this, **ask your tutor.**

Extra Challenge

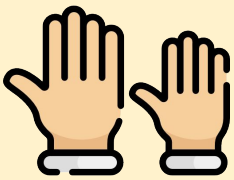
If you're teacher is happy with your work and when you've saved it properly, play the ***Basics*** section (Part 1) of Lightbot at this LINK.

Lesson 3 - Block Turtle II



Learning Outcomes:

- Making Tina draw some simple shapes at different locations on the map - Olympics Rings
- Make other shapes with Tina - a square, triangle, pentagon, hexagon etc.



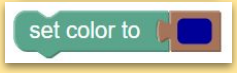
REMEMBER: Put up your hand. We love to help!

Refresher Task - Multiple Circles

Lets see if you can remember some of the things we learned about last week (it's ok to look back!). As a test, make **multiple circles** in various colours and positions. If you are stuck, open last weeks work. You will need:

- Shape 

- Speed 

- Colour 

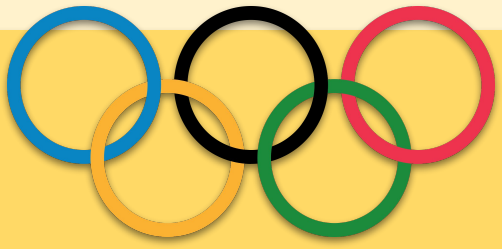
- Circle 

- Pen Up 

- Pen Down 

- Go to 

Extra Challenge



This challenge should be a tricky one.

Imagine *The Olympic Council of Ireland* have asked you to draw the Olympic flag so they can use it on their new website.

Use what you have learned in this lesson to write code to make Tina the Turtle draw the Olympic flag.

Writing with Turtles

Other than drawing lines and shapes, Tina the Turtle can also write words and sentences onto the canvas. We are now going to write something onto the canvas using Tina the Turtle.

www.bit.ly/blankblocktrinket

1. Set up Tina the Turtle
   Make turtle, set color, set speed etc.
2. Tell Tina what to write on the canvas.

3. Run the code above.
4. Change the color of the font.
5. Change the font size.

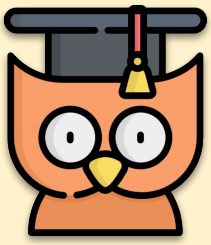
Revision Task

Fat Tony, a sweaty, overweight, middle-aged pizzeria owner wants you to come up with a new pizza recipe to lure in new customers. Use **Text** along with a picture of the pizza.

Use what you have learned so far to write code to make Tina the Turtle draw a delicious new pizza design with your own favourite toppings.

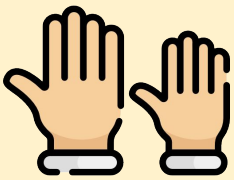
Hint: Once you have drawn the **outer crust** and **base** of the pizza, draw **different shapes** with **different colours** to draw your different toppings.

Lesson 4 - Python II - Rock Paper Scissors



Learning Outcomes:

- Creating a rock/paper/ scissors game
- Using simple if statements
- Announcing the winner



REMEMBER: Put up your hand. We love to help!

Let's Get Coding

1 **CTRL+CLICK www.bit.ly/rpstrinket**

This project contains these words.

```
from random import randint
```

We'll use them later to generate random numbers.

2 First, we're going to let the player choose Rock, Paper or Scissors by typing **r, p or s.** Add the blue lines of code. The second will print the player's choice.

```
from random import randint

player = input('rock (r), paper (p) or scissors (s)?')

print(player, 'vs')
```

3 Now to code the computers guess. randit will generate a random number to decide between rock, paper and scissors. Add in the blue lines and test it a few times to see that you're getting a random number.

```
player = input('rock (r), paper (p) or scissors (s)?')

print(player, 'vs')

chosen = randint(1,3)
print(chosen)
```

Coding the Computer

In our code, **1 = rock (r), 2 = paper (p)** and **3 = scissors(s).** We're going to use an **if statement** to check **if** the number 1 (for example) is randomly chosen, the computer will see that as a rock. Add the lines in blue making sure to **indent** (move to the right). We can use two spaces or a 'tab' for this. Tab is usually above the **Caps Lock** button on your keyboard.

```
chosen = randint(1,3)
print(chosen)

if chosen == 1:
  computer = 'r'
```

Don't forget the colon ':'

Two spaces or a 'tab'

```
chosen = randint(1,3)
#print(chosen)

if chosen == 1:
  computer = 'r'

elif chosen == 2:
  computer = 'p'

else:
  computer = 's'

print(computer)
```

Comment out this line by putting a '#' at the start

Short for "else if"

Add the rest of the code on the left, but first, *what does it do?* When finished, test your code to see if it works.

The computer's choice gets printed on a new line. You can fix that by adding `end=' '` after `vs`. This tells Python to end with a space instead of a new line. Play the game a few times by clicking Run and making a choice. For now you'll have to work out who won yourself. Next you'll add the Python code to work this out.

```
print(player, 'vs', end=' ')

chosen = randint(1,3)
#print(chosen)
```

```
rock (r), paper (p) or
scissors (s)? p
p vs r
```

Announcing the Winner

Lets look at a case where the player choses ‘r’ (rock). If the computer chose ‘s’ (scissors) then the player wins (rock beats scissors). If the computer chose ‘p’ (paper) then the computer wins (paper beats rock). We can check the player choice *and* the computer choice using the word **and**.

```
if player == computer:
  print('DRAW!')

elif player == 'r' and computer == 's':
  print('Player wins!')

elif player == 'r' and computer == 'p':
  print('Computer wins!')
```

Add in more elif’s so that every ‘p’ and ‘s’ combination is covered.

Challenge: ASCII Art

Instead of using the letters r, p and s to represent rock, paper and scissors, can you use ASCII art? For example:

```
rock (r), paper (p) or scissors
(s)? s
>8 vs ___
Player wins!
```

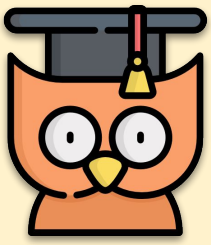
Challenge: Create a new game

Can you create your own game like Rock, Paper, Scissors with different objects? Click the ‘Duplicate’ button to make a copy of your Rock, Paper Scissors project to start from.

This example uses Fire, Logs and Water:

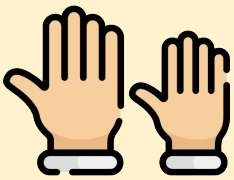
```
Fire, Logs, Water
Fire burns Logs
Logs make a bridge over Water.
Water puts out Fire
fire (f), logs (l) or water (w)? l
@@@ vs ~~~
Player wins!
```

Lesson 5 - Python III - Turtle Race



Learning Outcomes:

- Make a random race which races turtles for fun
- Learning about for loops
- Customising your game



REMEMBER: Put up your hand. We love to help!

Let's Get Coding

1 **CTRL+CLICK** this video to see the end result. The game will be similar to a horse racing game you may have seen before.

2 **CTRL+CLICK** this link: www.taoc.ie/blankpythontrinket

3 Write this code and see what happens. Then change the code so that it makes a pattern from **1-5** using the pattern below:

```
#!/bin/python3
from turtle import *

write(0)
forward(100)
write(5)
```

```
#!/bin/python3
from turtle import *

write(0)
forward(20)
write(1)
forward(20)
write(2)
forward(20)
write(3)
```

On the next page we'll discuss **for** loops, a method of making writing this long code easier, but for now, copy the code on the right.

For Loops - Making it easy

1 The code we wrote is very repetitive. There is an easier way to code this in python called a **for** loop. Change your code so it looks like the code on the right.

```
#!/bin/python3
from turtle import *

for step in range(6):
  write(step)
  forward(20)
```

We need to give it a **range of 6** because there are **6** numbers between 0 and 5.

```
#!/bin/python3
from turtle import *

penup()
goto(-140,140)

for step in range(15):
  write(step)
  forward(20)
```

2 Expand the code so that it matches what appears on the left.

But first, what does it do?

3 To draw the lines, we're going to have to turn the turtle to the right, before putting a pen down, drawing a line of 150, picking the penup and moving backward to the start of the line. Add the code in blue below and see what happens.

```
for step in range(15):
  write(step, align='center')
  right(90)
  forward(10)
  pendown()
  forward(150)
  penup()
  backward(160)
  left(90)
  forward(20)
```

`right(90)` makes the turtle turn right 90 degrees (a right angle.) Moving **`forward(10)`** before putting the pen down leaves a small gap between the number and the start of the line. After drawing the line you lift up the pen and go **`backward(160)`** the length of the line plus the gap.

Racing Turtles

1 When you use commands like **`forward(20)`** you are using a single turtle. But you can create more turtles. Add the following code to the end of your script (but make sure it's not indented)!

```
redTurtle = Turtle()
redTurtle.color('red')
redTurtle.shape('turtle')
redTurtle.penup()
redTurtle.goto(-160,100)
redTurtle.pendown()
```

The first line creates a turtle called 'red'. The next lines set the colour and shape of the turtle. Now it really looks like a turtle!

Now you need to make the turtle race by moving a **random number of steps** at a time. You'll need the **`randint`** function from the Python `random` library. Add this `import` line to the top of your script. The randint function returns a random number between the values chosen. The turtle will move forward **1, 2, 3, 4, or 5** steps each turn.

2

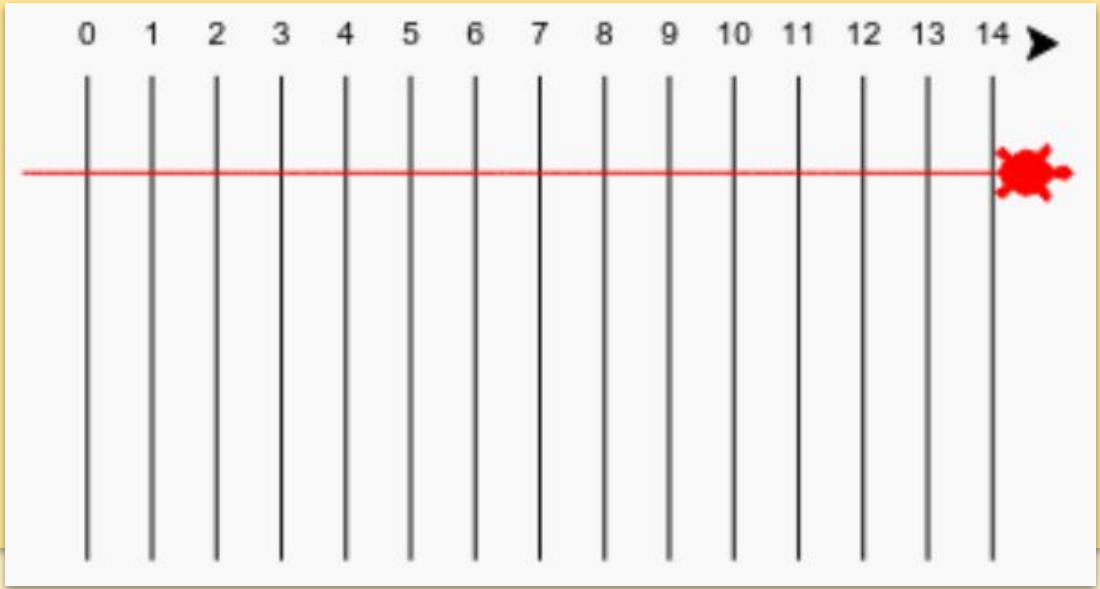
```
from turtle import *
from random import randint
```

```
redTurtle.goto(-160,100)
redTurtle.pendown()

for turn in range(100):
  redTurtle.forward(randint(1,5))
```

3

Test the code and you should get a one-turtle race! We'll add more next.



Adding more Turtles

```
redTurtle = Turtle()
redTurtle.color('red')
redTurtle.shape('turtle')
redTurtle.penup()
redTurtle.goto(-160,100)
redTurtle.pendown()
```

Copy

```
blueTurtle = Turtle()
blueTurtle.color('blue')
blueTurtle.shape('turtle')
blueTurtle.penup()
blueTurtle.goto(-160,70)
blueTurtle.pendown()
```

Paste

```
greenTurtle = Turtle()
greenTurtle.color('green')
greenTurtle.shape('turtle')
greenTurtle.penup()
greenTurtle.goto(-160,40)
greenTurtle.pendown()
```

Paste

```
yellowTurtle = Turtle()
yellowTurtle.color('yellow')
yellowTurtle.shape('turtle')
yellowTurtle.penup()
yellowTurtle.goto(-160,10)
yellowTurtle.pendown()
```

Paste

```
for turn in range(100):
  redTurtle.forward(randint(1,5))
  blueTurtle.forward(randint(1,5))
  greenTurtle.forward(randint(1,5))
  yellowTurtle.forward(randint(1,5))
```

Copy
Paste
Paste
Paste

A one turtle race isn't much of a game! Add the code on the left to get the remaining turtles going.

You should be able to see a pattern here so don't forget...

REMEMBER TO:

COPY (*CTRL + C*)

AND

PASTE (*CTRL + V*)

Copy and Paste are a coders best friends!

Challenge: Do a twirl

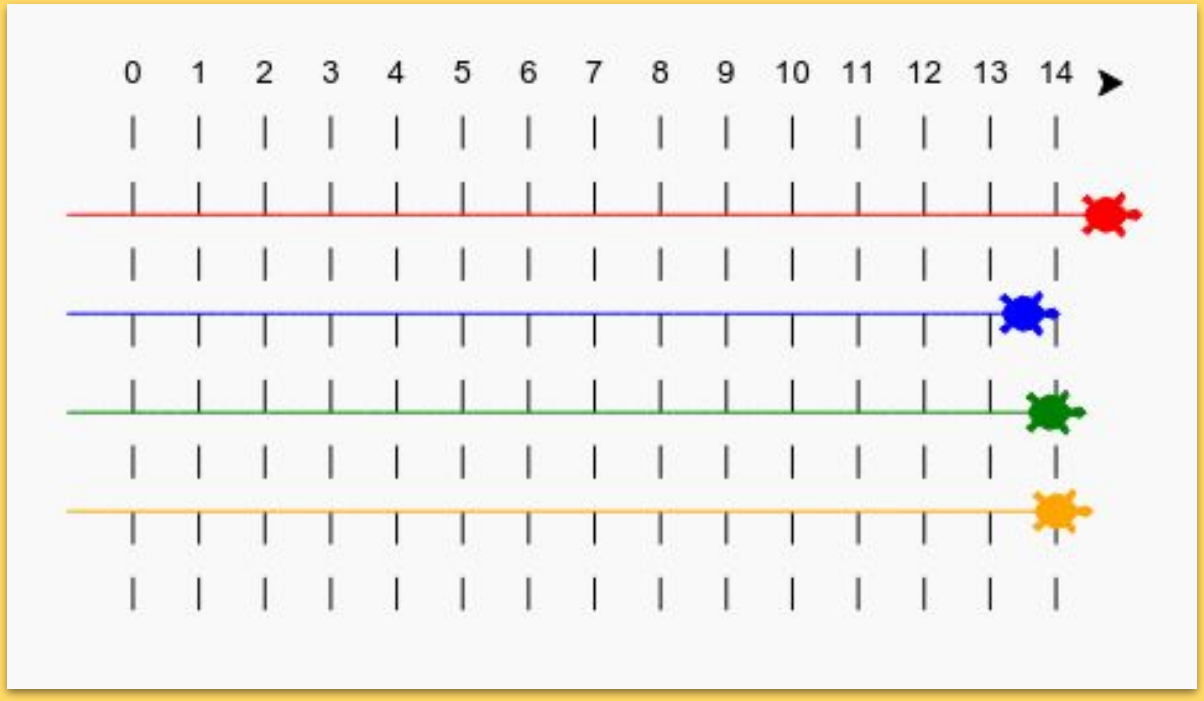
Can you use a **`for turn in range():`** loop to make each turtle do a 360 degree twirl after they get to the starting line? You'll need to make sure they are facing in the right direction at the start of the race!

`redTurtle.right(36)` will turn the red turtle right by 36 degrees.

Hint: A full turn is 360 degrees. A turtle could turn right 10 degrees 36 times, or left 5 degrees 72 times, or any other numbers make 360!

Challenge: Dashed lines

Can you use a loop to make the track lines dashed instead of solid?



Hint: Find the code that draws a straight line.

Try using:`for`, `forward()`, `penup()` and `pendown()`