**Warminster School**

**CompSci**

**Turtle Workbook**



Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**HOW THIS BOOK WORKS**

Inside this workbook, you will find useful information, code snippets, and all the activities that you need to complete in this unit of work. Be sure to complete each activity in the order that they are shown in this book – don’t try and skip ahead until you have successfully completed each activity.

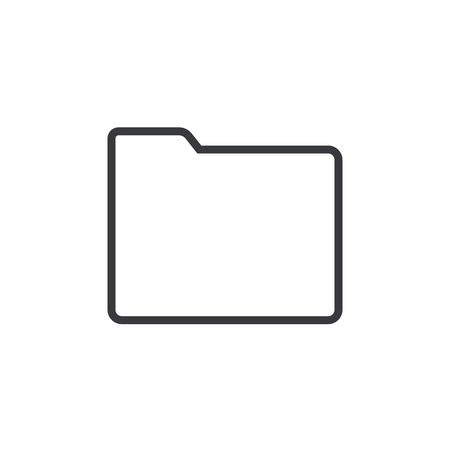
**HELP**

If you are stuck on a problem that you just can’t solve yourself, try these strategies to find an answer:

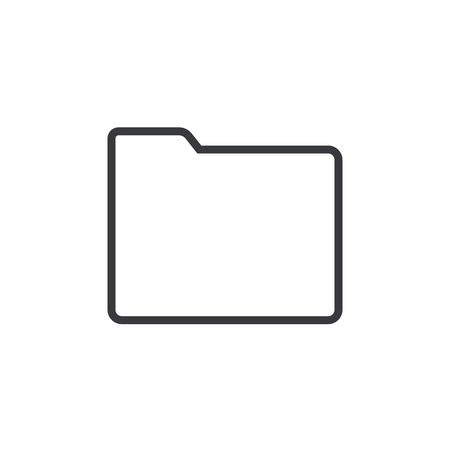
* Ask a friend sitting next to you (not on the other side of the room)
* Research online (Google will quite often have the answer to your problem)
* Ask your teacher (he’s not as scary as he seems and is happy to help you out)

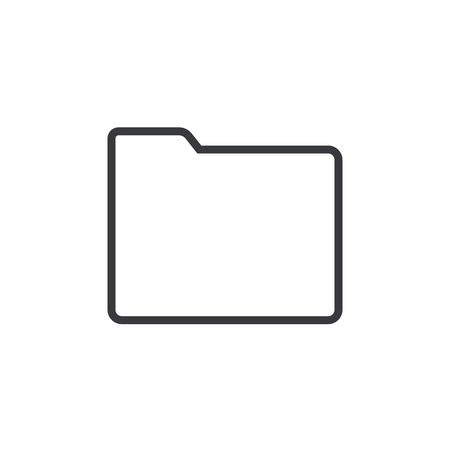
**GETTING ORGANISED**

Your files must be neatly organised on your computer. Setup the following folder structure inside your Year 7/*CompSci* folder (you should have this created already):

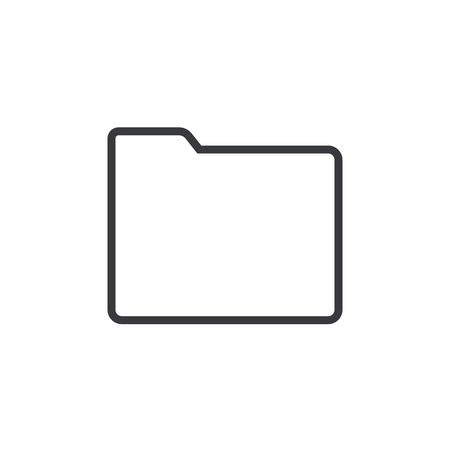


**Turtle**





**Activity Code**



**Assessment**

**Tutorial Code**

* **Tutorial Code** – each time you watch a video tutorial, you will be required to follow along and make the same program in Python that the teacher makes. Once you have finished watching the video tutorial and writing your code, save it into this folder.
* **Activity Code** – throughout this booklet, you will find a range of coding activities that you need to complete on your own. As you finish each activity, save the code into this folder with the specified name.
* **Assessment** – any assessment work that you complete must be saved into this folder.

Each week, you will be required to check-in with your teacher to provide an update on how you are progressing. Your teacher will access the work from the above folders and tick off all the items that you have completed. All work must be saved with appropriate and meaningful file names.

**YOUTUBE VIDEO TUTORIAL LINKS**

The following videos should be watched before attempting to complete the associated activities in this workbook. They will show you how to perform a range of useful functions using Python’s Turtle module and will have you coding your own graphics in no time at all!

* Tutorial 1 – [Introduction to the Turtle](https://youtu.be/Tk2Dhp876UY)
* Tutorial 2 – [Draw a Triangle](https://youtu.be/T8BL-MHMZkc)
* Tutorial 3 – [Draw a Square](https://youtu.be/yhhsJgXyok8)
* Tutorial 4 – [Draw a Circle](https://youtu.be/AyuLwv6dJdI)
* Tutorial 5 – [Draw an Oval](https://youtu.be/h_AQLDiqUYA)
* Tutorial 6 – [Draw a Star](https://youtu.be/XZcbrUGZkpw)
* Tutorial 7 – [Draw a Cool Pattern](https://youtu.be/WiShwgtWWHI)
* Tutorial 8 – [Draw a Factory](https://youtu.be/kX4VweLMglc)
* Tutorial 9 – [Draw the Aboriginal Australian Flag](https://youtu.be/RC4Xbil5LBc)
* Tutorial 10 – [Draw a Domino](https://youtu.be/8RdhPPKq5AI)
* Tutorial 11 – [Draw a Fallout Shelter Logo](https://youtu.be/DMx9Ce6K5Ro)
* Tutorial 12 – [Draw a Building](https://youtu.be/exsn0aVTTlo)
* Tutorial 13 – [Asking](https://youtu.be/pO0H1wRHKQ0) Questions
* Tutorial 14 – [Draw a House](https://youtu.be/WEUloPdIfRk)
* Tutorial 15 – [Draw a Mountain Range](https://youtu.be/UjQrJu8wJbA)



**Chapter 1**



Introducing the Turtle

**THE TURTLE**

This term, you will be expanding your knowledge of Python code to learn how to draw graphics on to the computer screen. To do this, you will be using the **Turtle** module – which is simply an extra part of Python. A module is a library of code written by a very intelligent person. We need to import the functions (snippets of code) inside of the library so that we can use these extra features in our code.

**IMPORTING THE TURTLE**

To import all of the functions in the Turtle module, we need to add the following line of code to the start of every program we write this term:



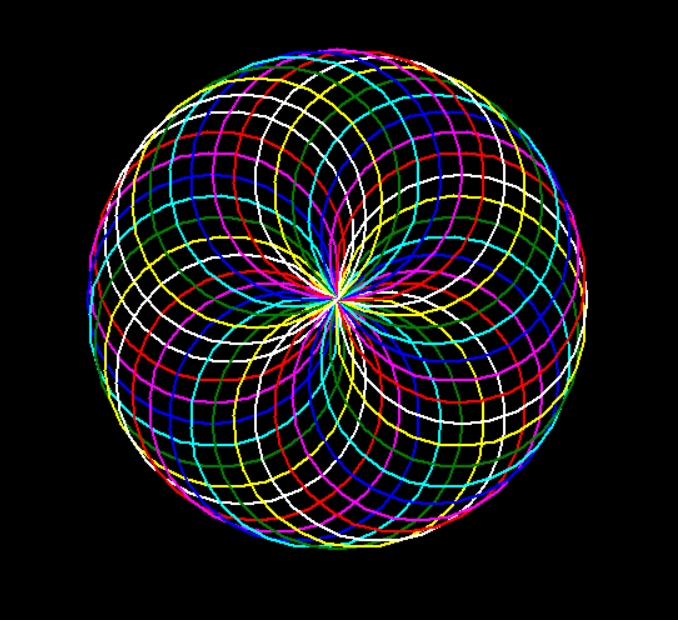
This line reads “from the turtle module, import all of the functions inside of it.” This means you will have access to all the code to allow you to draw graphics on the screen.

**WHAT THE TURTLE LOOKS LIKE**

When you run your first program with the Turtle, you will see a little symbol appear on the screen that will draw your graphics. Even though it doesn’t look like one, this is the Turtle.

**WHAT CAN THE TURTLE DO?**

The Turtle is a great way to learn how to code using Python. By using graphics, it’s a bit more interesting that the usual boring text you see on the screen when coding. The Turtle is capable of drawing almost anything you dream up – you just need to know how to code it! Cool 2D artworks and amazing patterns are just some of the things that the Turtle is capable of drawing.



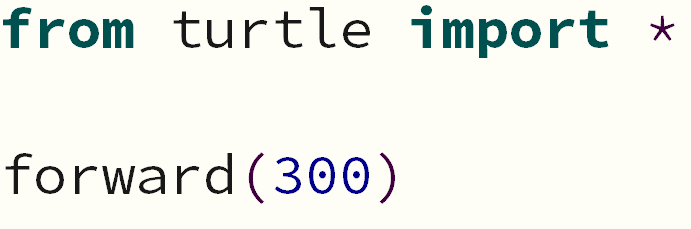
**Chapter 2**



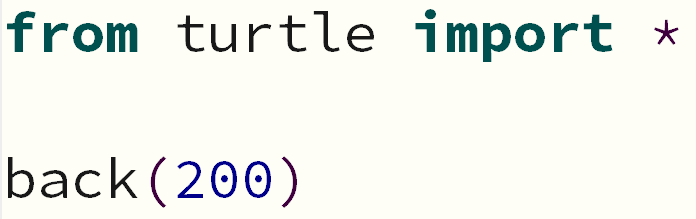
Movement & Colour

**MOVING THE TURTLE**

You need to think of the Turtle as a pen – as it moves across your screen (which is your canvas), it will draw a line. We can move the Turtle **forward** and **back**.

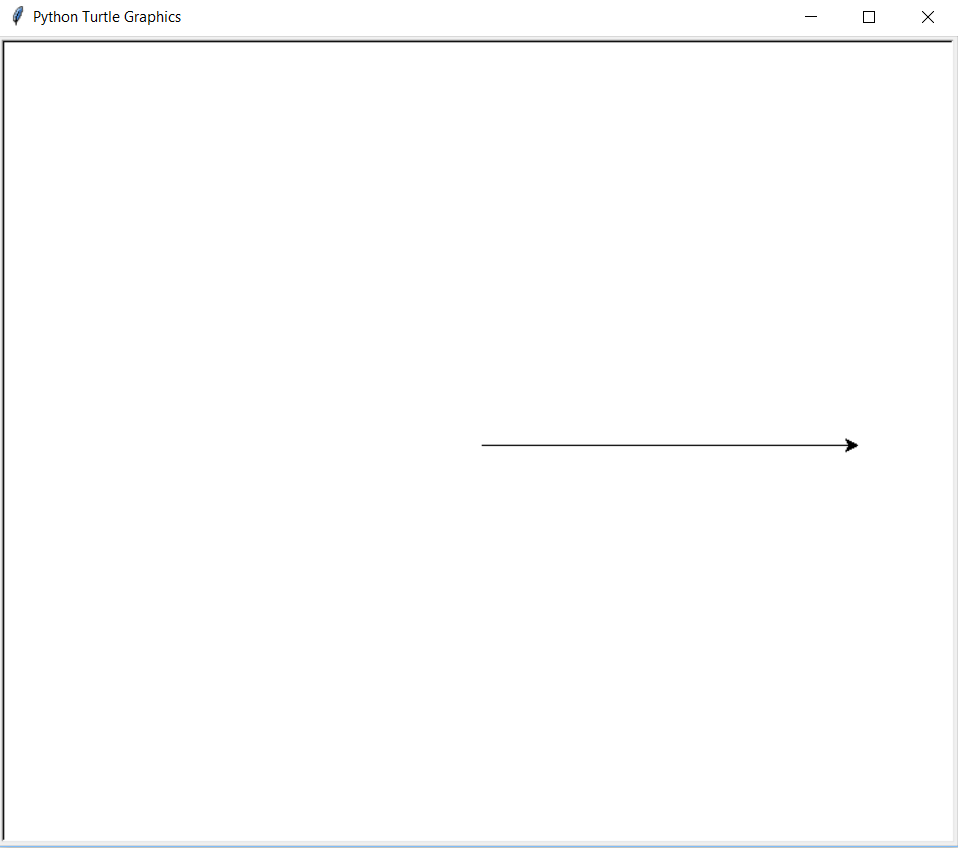
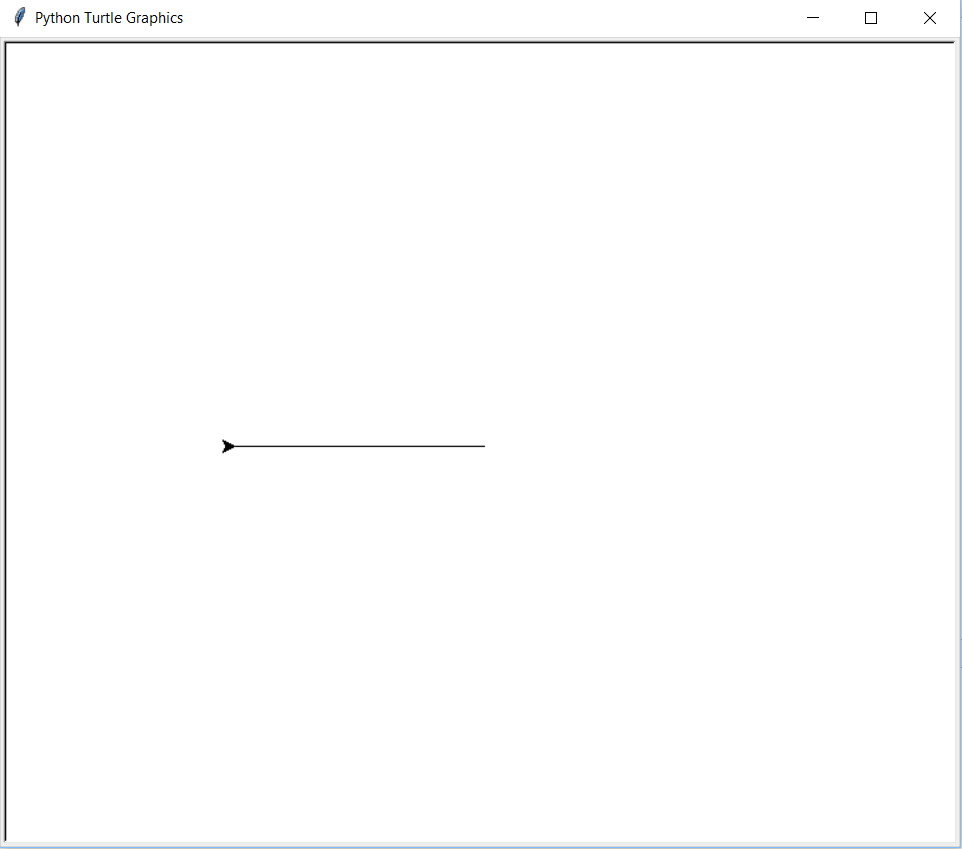


This code will move the Turtle *forward* 300 turtle steps.



This code will move the Turtle *back* 200 turtle steps.

The bigger the number, the further your turtle will go. Obviously, the smaller the number, the smaller the line that the Turtle will draw. Try it out!

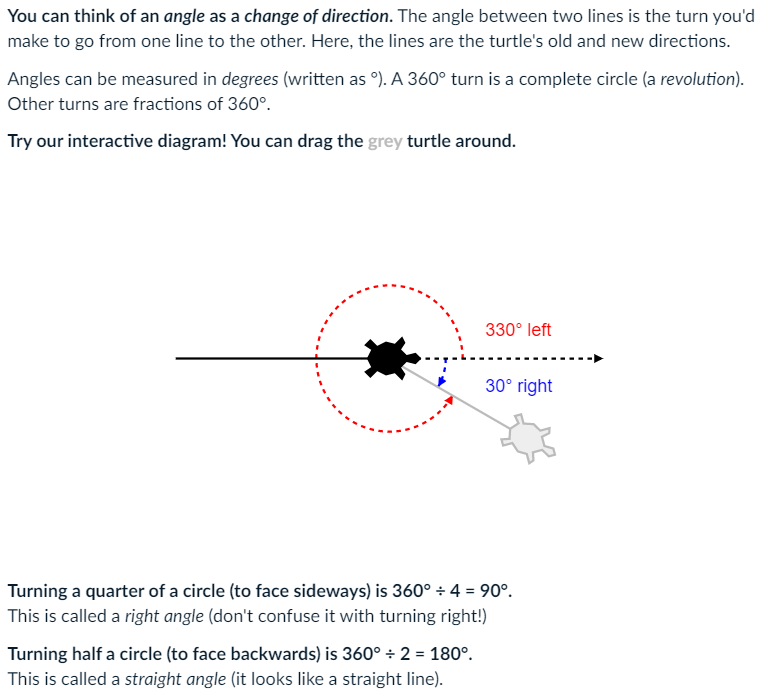
 

*This is what happens when you move* ***back*** *200 steps*

*This is what happens when you move* ***forward*** *300 steps*

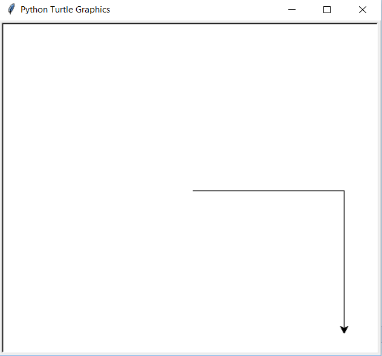
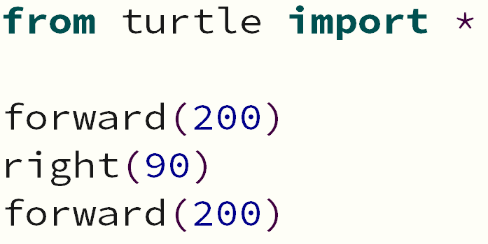
**CHANGING DIRECTION**

By default, **your Turtle will always start facing to the right**. It is, however, easy to change the direction in which the Turtle will draw a line. To do this, you will need to tell the Turtle which direction to turn (**left** or **right**) and at how many degrees. Below is an example of how many degrees you would need to turn the Turtle to head off towards the bottom right of the page. You could either turn left or right – it’s just the number of degrees that you turn that will change.

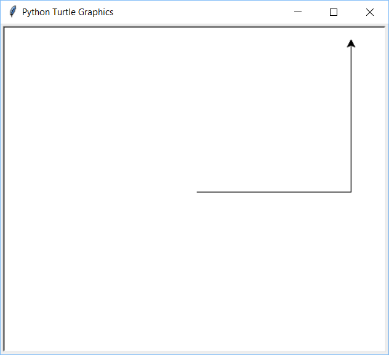
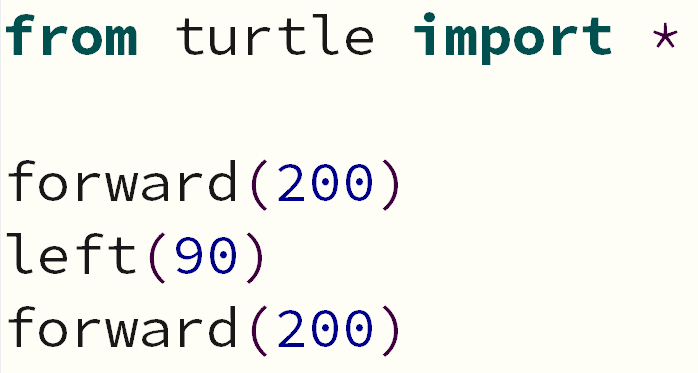


**EXAMPLES**

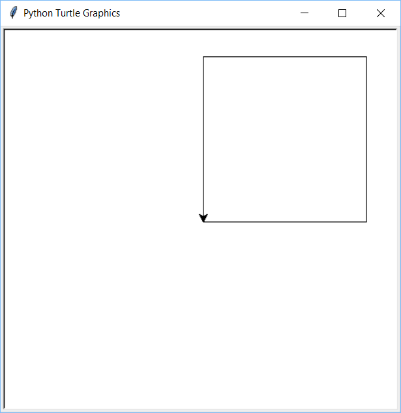
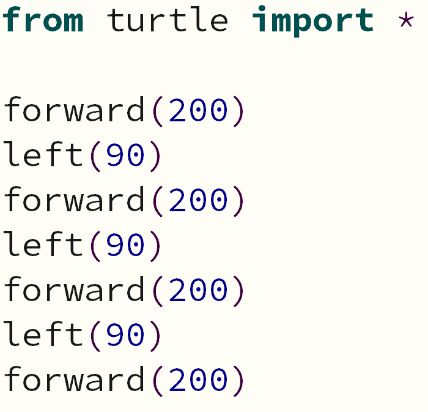
Let’s move your Turtle to create a right angle (90-degrees):



If we were to run the same code, but turn to the left, it would look like this:

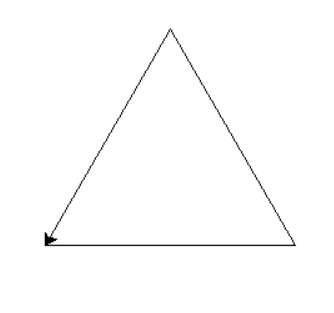


By continuing to turn at 90-degrees, you are able to make your first shape:



**CHANGING COLOUR & SIZE**

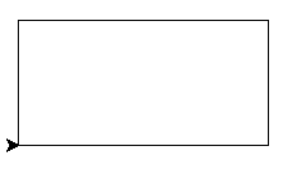
|  |  |
| --- | --- |
| You are able to add a splash of colour to your lines by using the **color** function. A list of all the different colours you can use in Python is on the next page. | You can also change the thickness of your line by using the **pensize** function. The higher the number, the thicker the line becomes: |
|  |  |
|  |  |

**ACTIVITIES TO COMPLETE** ✍



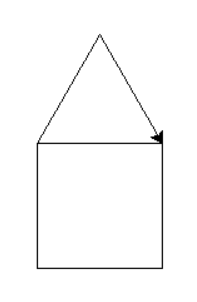
1. Write a program that will draw an equilateral triangle. Each side needs to be 100 steps in length. The angle at which you need to turn is 120-degrees.

*Save this program into your Activity Code folder as 1\_triangle.py*



1. Write a program that will draw a rectangle. The longer sides should be 200 steps in size. The shorter sides need to be 100 steps.

*Save this program into your Activity Code folder as 2\_rectangle.py*



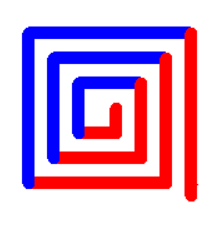
1. Write a program that will draw a simple house. All sides of the house and roof need to be 100 steps long. When drawing the house, try and do it so that the house is drawn with one single line – meaning, you should not draw over any existing lines.

*Save this program into your Activity Code folder as 3\_house.py*

1. Create a red heartbeat pattern that you would see on a heartbeat monitor. It will have 3 pulses that:

* Moves 20 steps forward before the start of the pulse;
* Turns 80° left to draw the start of the pulse;
* Moves 20 steps up to draw the start of the pulse;
* Turns 160° right at the top of the pulse;
* Moves 40 steps to draw the main part of the pulse;
* Turns 160° left at the bottom of the pulse;
* Moves another 20 steps to draw the last part of the pulse;
* Has a gap of 40 steps between each pulse

*Save this program into your Activity Code folder as 4\_heartbeat.py*



1. Write a program that will recreate the coloured pattern in the image on the right. You will need to use any 2 colours and a line thickness of 10. It might seem tricky, but once you work out the pattern, it is quite easy!

*Save this program into your Activity Code folder as 5\_Square\_pattern.py*

**MOVING TO SPECIFIC COORDINATES**

The screen in Python has been setup with an x-axis and a y-axis. The default Turtle window size is approximately   
750 x 640 steps. It looks like this:

**Y-AXIS**

**X-AXIS**

(0, 0)

(0, -320)

(-375, 0)

(375, 0)

(0, 320)

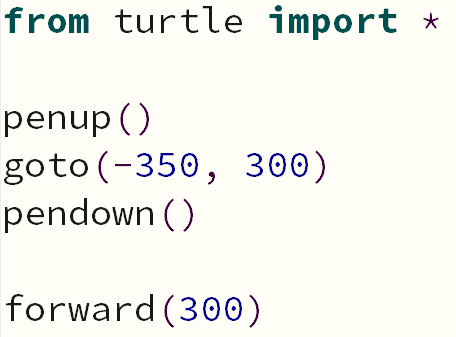
(-375,320)

(375,320)

(375,-320)

(-375,-320)

When you run your code, Python automatically positions your Turtle at the coordinates (0, 0) – which is the centre of the screen. You can, however, move the Turtle to any position you like on the screen. You will need to lift your Turtle up off the page, go to the desired coordinates and then put your Turtle back down on the page ready for drawing.



Remember to think of your Turtle as a pen that is drawing on paper.

**penup()** is the function used to lift your Turtle (your “pen”) up off the page. If you don’t lift your Turtle up off the page, it will draw a line while it is moving.

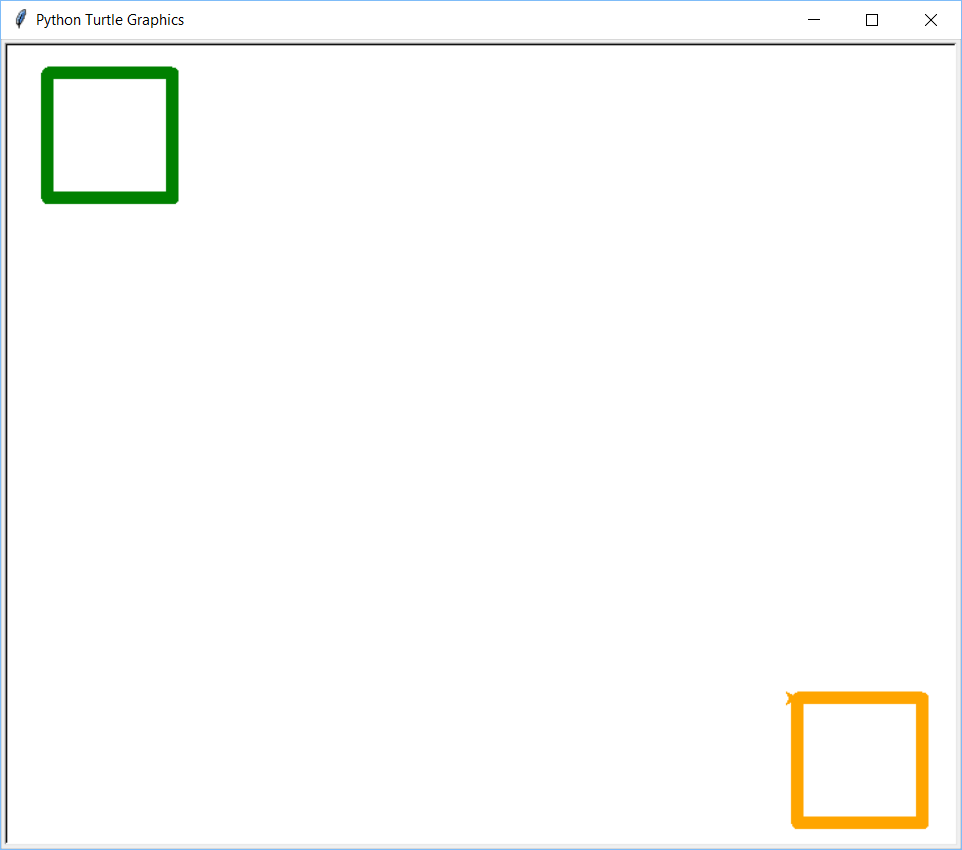
**goto()** is the function that moves your Turtle to the desired coordinates (x, y).

**pendown()** puts your Turtle (your “pen”) back down on the page ready to begin drawing.

**ACTIVITIES TO COMPLETE** ✍

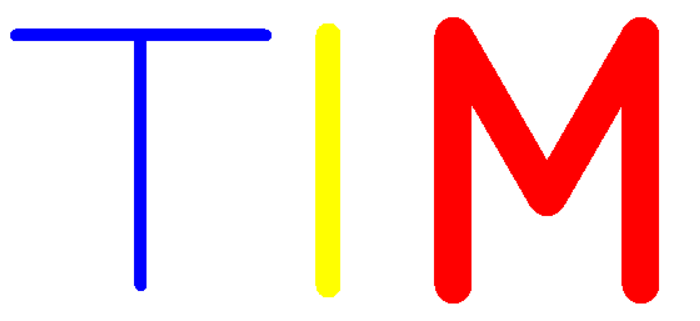
1. Write a program that will draw a square anywhere in the top left quadrant of the page with a green outline. Draw a second square anywhere in the bottom right quadrant of the page with an orange outline. Both squares must have equal sides of 100 steps and a pen size of 10.

*Save this program into your Activity Code folder as 6\_squares.py*



1. Write a program that will draw your name on the screen using only lines. Each letter in your name needs to be a different colour and a different thickness to the other letters in your name. Feel free to use a nickname or your last name if it makes things easier.

*Save this program into your Activity Code folder as 7\_name.py*



**Chapter 3**



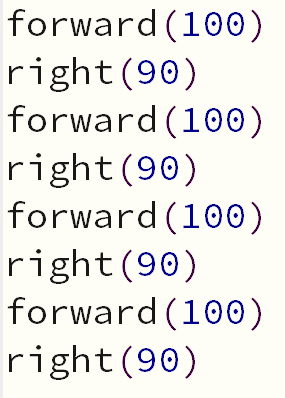
Drawing Shapes & Patterns   
with Loops

**DRAWING SHAPES WITH LOOPS**

As you saw in the previous chapter, drawing shapes using the Turtle is quite easy. But, it can get even easier (and quicker) when you use a loop! You might have noticed that when drawing things like squares, rectangles and triangles, the same lines of code are repeated numerous times. As we know, we can cut down the amount of repeated code by using a loop.

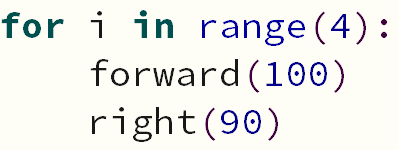
**EXAMPLE**

Instead of writing the first chunk of code to draw a square, it could be easily written with a for loop, like in the second:



Both snippets of code will draw the exact same square.

Try to always use the most efficient approach when writing code (which is the loop in this case).



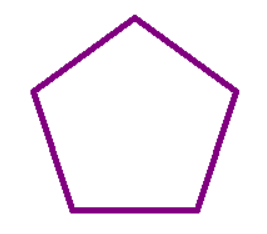
Below are some snippets of code that will draw certain shapes:

|  |  |  |
| --- | --- | --- |
| Square |  |  |
| Rectangle |  |  |
| Triangle |  |  |
| Circle | The number in brackets is the radius size – so, the bigger the number, the bigger the circle. |  |
| Oval | The shapesize() function allows you to distort the shape. |  |
| Star |  |  |

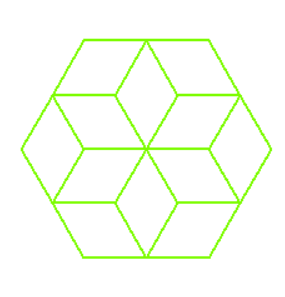
**DRAWING PATTERNS**

By using the same method that you used to draw basic shapes (like squares and triangles), you can create some pretty cool patterns. Check out the examples below:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

 **ACTIVITIES TO COMPLETE** ✍

1. Write the code to draw a pentagon with 100 step sides. It must have an outline of size 5 and colour of ‘hotpink’. Your code needs to be as efficient as possible, meaning that your shape must be drawn with a for loop. Remember, pentagons have 5 sides, so to work out the angle you need at each corner, divide 360 by 5.

*Save this program into your Activity Code folder as 8\_pentagon.py*

1. Using inspiration from the pattern examples above, create your own pattern using a for loop. Be sure to change the outline thickness and the colour. This will take some trial and error to get it looking good. Try a variety of shapes and angles.

*Save this program into your Activity Code folder as 9\_pattern.py*

**Chapter 4**



Fill Colours

**FILL COLOURS**

Not only can we change the colour of the lines that the Turtle draws, but we can also colour shapes in with a fill colour. To do this, we still use the **color** function, but also need to use the **begin\_fill** and **end\_fill** functions to let the Turtle know when to start and stop filling our shapes with colour.

**EXAMPLE**

|  |  |
| --- | --- |
| We know this is the code to draw a square with a red outline: | If we wanted to colour this square in red, we would need to add a couple of extra lines to this code: |
|  |  |
|  |  |

**OUTLINE & FILL COLOURS**

It is also possible to have a shape that has a fill colour with a different coloured outline. You just need to add both of your chosen colours to the **color** function.

**EXAMPLE**

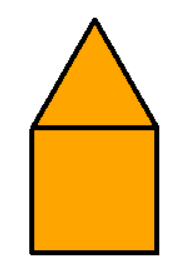
|  |  |
| --- | --- |
|  |  |

**HIDING THE TURTLE**

You have probably been wondering how to hide the little Turtle icon as it goes around drawing your graphics.

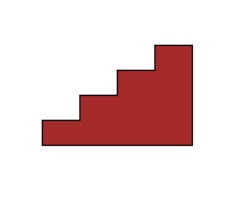
All you need to do is add the **hideturtle** function to the end of your code. Once the Turtle has drawn what it needs to, it will disappear.



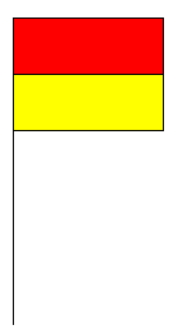
**ACTIVITIES TO COMPLETE** ✍

1. In Activity 3, you drew a simple house. All sides of the house and roof were 100 steps in length. You are to write the code to recreate this exact house, except this time, you are to colour it in. It must have a black outline with a fill colour of your own choosing.

*Save this program into your Activity Code folder as 10\_coloured\_house.py*

1. Write a program that will draw a staircase of 4 steps. They must be 20 turtle steps high and 30 turtle steps wide. The staircase will go up and to the right of the screen. The staircase will have a black outline and a brown fill colour, as shown in the example on the right.

*Save this program into your Activity Code folder as 11\_staircase.py*



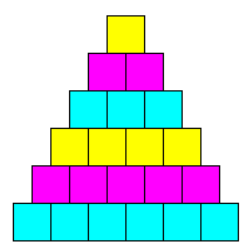
1. Write a program that will draw the red and yellow lifesaving flag. The flag is made up of the following features:

* a flagpole that is 200 steps high
* a red rectangle on top that is 120 steps long and 45 steps high
* a yellow rectangle on the bottom that is 120 steps long and 45 steps high

*Save this program into your Activity Code folder as 12\_lifesaver.py*

1. Write a program that will draw a colourful pyramid that is 6-storeys high. Each square block that you see in the pyramid has sides of 30 steps. You will need to use a loop to draw each row, as well as another loop to draw each square. You will need to be specific with your coordinates as well, so that your pyramid looks symmetrical. Use the colours of cyan, magenta and yellow.

*Save this program into your Activity Code folder as 13\_pyramid.py*



**Chapter 5**



Asking Questions &

Making Decisions

**ASKING QUESTIONS**

It is possible to have the Turtle draw something based upon a user’s response to a question.

**EXAMPLE**

You might ask the user the question, “what is your favourite colour?” Based upon their response, the Turtle will display a square with their favourite colour inside of it. In the example below, the user types in ‘red’ as their favourite colour. It then prints out a square using that colour.

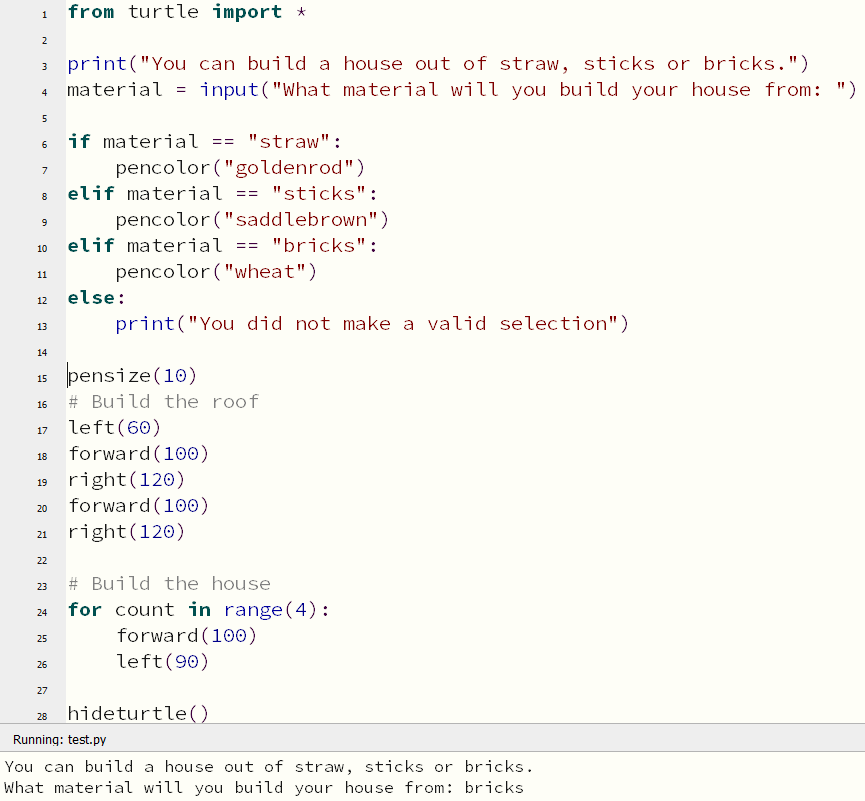
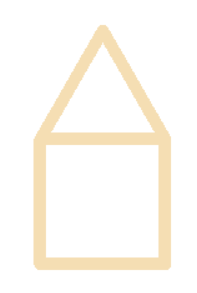


**MAKING DECISIONS**

If-statements can be used to draw particular thing on the screen, based upon what a user has typed in.

**EXAMPLE**

In the code below, the computer asks the user what material they would like to build their house out of. Once chosen, the computer will draw the house with a particular colour, depending on what the user has requested.



**ACTIVITIES TO COMPLETE** ✍

1. Write a program that will draw a wooden fence. The program will begin by asking the user 2 questions:

* How many planks they want in their fence
* What colour they want their fence to be

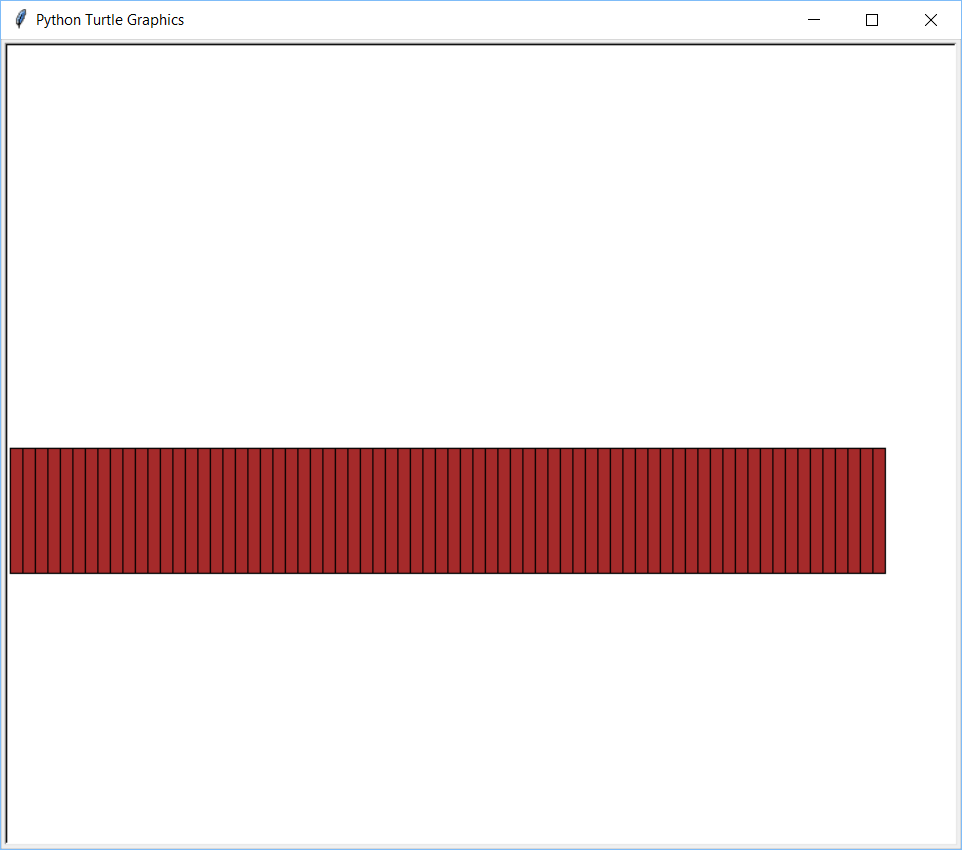
Once you have this information, you can begin drawing the fence. Your program should start off by moving 380 steps to the left so that it starts off at the left-edge of the screen.

Important things to note:

* The outline of the fence should be black.
* Each plank should be 100 steps high and 10 steps wide.
* You must be using loops to create this fence.

*Save this program into your Activity Code folder as 14\_fence.py*





**Chapter 6**



Final Tips

**FINAL TIPS**

Below, you will find some tips and tricks to help you out when coding your designs.

**SETTING THE SCREEN SIZE**

As mentioned earlier, when you run a program using the Turtle, the default screen size approximately 750 x 640 steps. You are able to adjust this to suit your needs. Simply use the **setup** function to do this. Put this line of code near the start of your program to set your own custom screen size.

**EXAMPLE**



**SETTING THE SPEED OF THE TURTLE**

The Turtle is able to draw at various speeds. This setting is controlled by the **speed** function, which can be set between 1 (slowest) and 10 (fastest). If you set the speed to 0, it has a special meaning - turn off the animation and draw as fast as possible.

Sometimes it is handy to watch the Turtle draw slowly to see exactly what way it is turning and how you might be able to fix an error. At other times, you just want the Turtle to draw things as quickly as possible so you don’t have to wait around and watch the animation. Either way, you have the option of setting its speed to your desired preference. Place the following code near the beginning of your program.

**EXAMPLE**



**SETTING THE BACKGROUND COLOUR**

To change the colour of your background, simply use the **bgcolor** function. You have access to all the different colours mentioned earlier in this book on Page 9.

**EXAMPLE**



**ADDING TEXT**

It is possible to add text to your art work by using the **write** function. You can adjust the font family, size and style as well.

**EXAMPLE**



**Chapter 7**



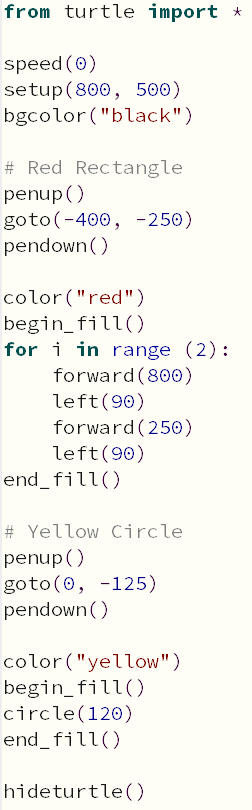
Putting It All Together

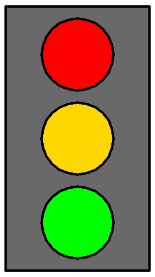
**PUTTING IT ALL TOGETHER**

You now have the skills to create some very cool artwork with code – it’s just a matter of bringing it all together.

**EXAMPLE**

To draw the Australian Aboriginal Flag, you will need 2 rectangles and a circle. Whatever you code first will be in the background (the rectangles), while the last thing you code with be on top in the artwork (the circle – which will sit on top of the rectangles).



**ACTIVITIES TO COMPLETE** ✍

1. Write a program that will draw a set of traffic lights. This graphic will be made up of a rectangular body with 3 circular lights on top of it. Use a black outline for all shapes.

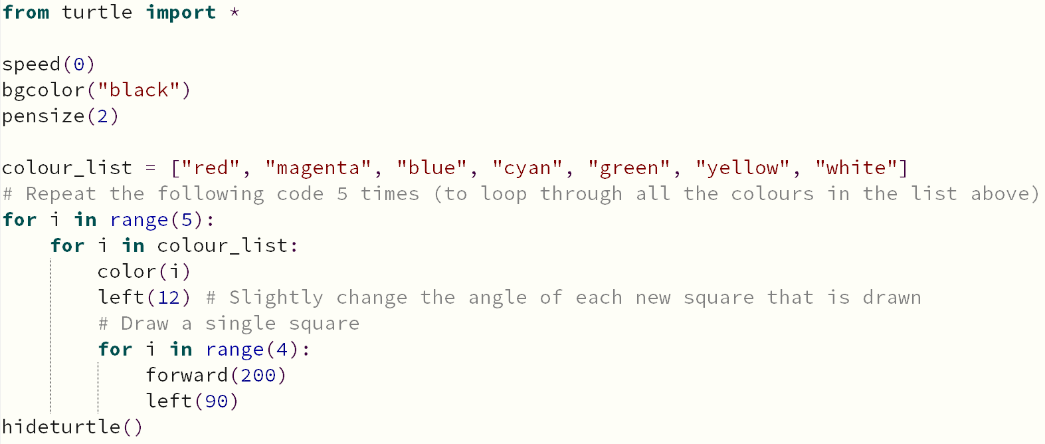
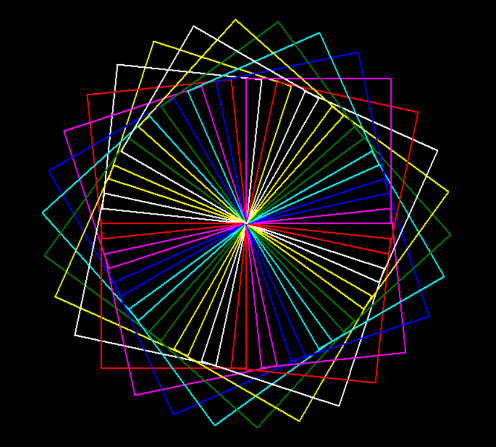
*Save this program into your Activity Code folder as 15\_traffic\_lights.py*



1. Write a program that will draw a truck. At the very least, it should include a cabin, trailer, tyres and some text on the side of the truck. You also need to include the sky and a road for the truck to drive on. Make sure the screen size is 800 wide by 500 high.

*Save this program into your Activity Code folder as 16\_truck.py*

**EXTENSION TASK** ✍

Analyse the code below – it will make the colourful pattern you see on the right.

Manipulate this code to create your own colourful pattern. It could be of any shape and may use whatever colours you would like. Get creative and finish this booklet with a bang!

**CONGRATULATIONS!**

You have now completed all of the Activities in this workbook. Make sure you have had your checklist fully signed off by your teacher. It is now time for you to start your assessment task.

**CHECKLIST NAME:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **CLASS:** \_\_\_\_\_\_

As you complete each of the tutorials and activities, you must show your teacher so that they can sign you off to say that you have successfully completed that task.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tutorial** | **Due** | **Teacher Signature** | **Activity** | **Due** | **Teacher Signature** |
| 1. Introduction to the Turtle | Wk 1 |  | 1. Triangle | Wk 1 |  |
| 1. Draw a Triangle | Wk 1 |  | 1. Rectangle | Wk 1 |  |
| 1. Draw a Square | Wk 1 |  | 1. House | Wk 1 |  |
| 1. Draw a Circle | Wk 1 |  | 1. Heartbeat | Wk 2 |  |
| 1. Draw an Oval | Wk 1 |  | 1. Square Pattern | Wk 2 |  |
| 1. Draw a Star | Wk 1 |  | 1. Squares | Wk 3 |  |
| 1. Draw a Cool Pattern | Wk 2 |  | 1. Name | Wk 3 |  |
| 1. Draw a Factory | Wk 2 |  | 1. Pentagon | Wk 3 |  |
| 1. Draw the Aboriginal Flag | Wk 2 |  | 1. Pattern | Wk 3 |  |
| 1. Draw a Domino | Wk 3 |  | 1. Coloured House | Wk 4 |  |
| 1. Draw a Fallout Shelter Logo | Wk 3 |  | 1. Staircase | Wk 4 |  |
| 1. Draw a Building | Wk 3 |  | 1. Lifesaver Flag | Wk 4 |  |
| 1. Asking Questions | Wk 4 |  | 1. Pyramid | Wk 4 |  |
| 1. Draw a House | Wk 4 |  | 1. Fence | Wk 5 |  |
| 1. Draw a Mountain Range | Wk 5 |  | 1. Traffic Lights | Wk 5 |  |
|  |  |  | 1. Truck | Wk 5 |  |
|  |  |  | EXTENSION TASK Colourful Pattern | Wk 5 |  |