COMP1021
Introduction to Computer Science

Introduction to Turtle Graphics

David Rossiter, Gibson Lam and Oz Lam

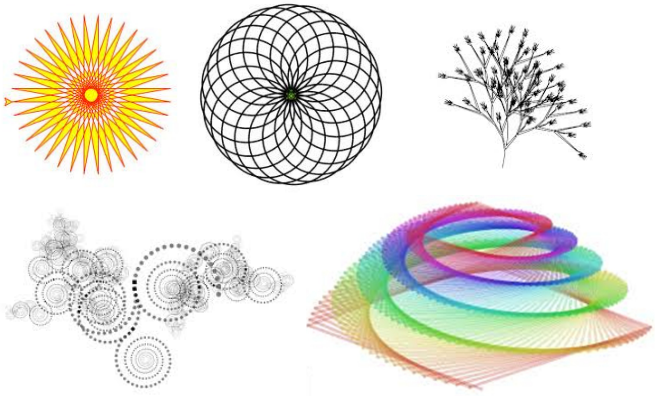
Outcomes

- After completing this presentation, you are expected to be able to:
  1. Explain what a turtle in turtle graphics can do in Python
  2. Use the turtle to draw a variety of things in a Python program

Turtle Graphics

- We will use 'turtle graphics' a lot, to help us learn Python
- Turtle graphics is a simple but powerful kind of graphics programming
- It has a turtle and a turtle window
- The turtle moves around the turtle window and draws things
- You write commands to control the turtle

Some Examples of Turtle Graphics



Getting Started with the Turtle

- To start using the turtle in Python, you need to tell Python that you want to use the turtle commands, by typing this: `import turtle`
- After that, you can start to draw things by using various turtle commands
- Sometimes when you have finished drawing, you use the `turtle.done()` command to tell Python that you have finished your program
- (Actually you usually don't need `turtle.done()` until we do advanced things later in the course)

A Turtle Graphics Program

- The basic structure of a turtle graphics program looks like this:

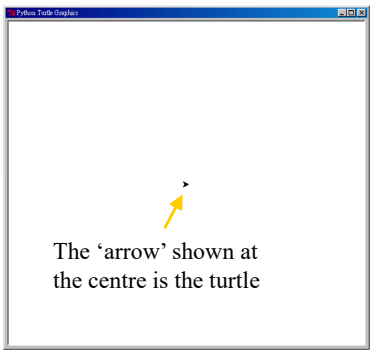
```
import turtle

... Draw things using the turtle ...

turtle.done()
```

- When a line of code is executed which tells the turtle to do something a window appears which shows the result

The Turtle

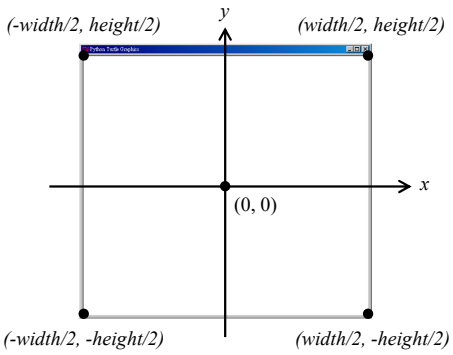
- The turtle has a position, an orientation and a pen
  - The position is where the turtle is, inside the turtle window
  - The orientation is the direction the turtle is looking at, ('orientation' basically means 'angle')
  - The turtle holds a pen, which has a colour and thickness
- The turtle can put the pen down or lift the pen up
- When the pen is down and the turtle moves, the turtle draws something

The Turtle Window



- The turtle moves around and draws inside the turtle window
- Initially the turtle starts at position (0, 0), which is in the middle of the window, and it points towards the right hand side of the window

The Turtle Graphics Coordinate System

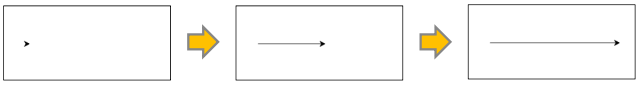


Walking Forwards

- You can ask the turtle to walk forwards
- For example, to tell the turtle to walk 250 pixels forwards, you can use this line of code:

```
turtle.forward(250)
```

- After running the code, you can actually see the turtle moving from one position to another

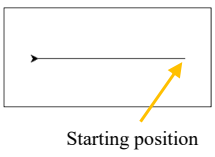


- When the turtle moves, it draws a line

Walking Backwards

- Similarly, you can ask the turtle to move backwards using the `turtle.backward()` command
- For example, the following line of code asks the turtle to move backwards by 200 pixels:

```
turtle.backward(200)
```



Changing the Line Thickness

- You can draw a thicker line by changing the *width* of the pen the turtle is holding using the `turtle.width()` command
- The following code draws a thin line and then a thick line by changing the width of the pen to 5 pixels:

```
turtle.forward(125)
turtle.width(5)
turtle.forward(125)
```

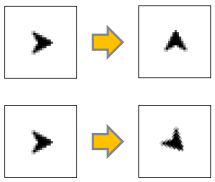
Turning Left and Right

- The turtle can do much more than walking forwards and backwards horizontally
- You can turn turtle using the `turtle.left()` and `turtle.right()` commands
- For example, to turn left by 90 degrees:

```
turtle.left(90)
```

- And, to turn right by 45 degrees:

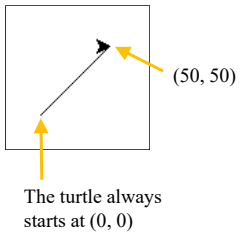
```
turtle.right(45)
```



Moving to an Arbitrary Location

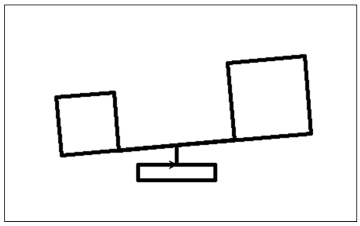
- The turtle can move around the window by walking forwards, backwards and turning
- Sometimes it may be useful to ask the turtle to move directly to a particular position in the window
- To do that you can use the `turtle.goto()` command
- For example, you can ask the turtle to move to the location (50, 50) using this line of code:

```
turtle.goto(50, 50)
```



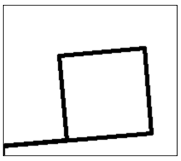
An Example Using Multiple Commands

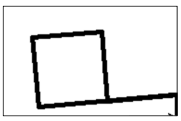
- This example demonstrates how the turtle reacts to a combination of the turtle commands that we have met so far
- Result:



Example Code 1/2

```
turtle.left(5) # Tell the turtle to turn left 5 degrees

# Draw the right box and the connecting line
turtle.forward(175)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)

turtle.right(90)

# Draw the left box and the connecting line
turtle.forward(225)
turtle.right(90)
turtle.forward(75)
turtle.right(90)
turtle.forward(75)
turtle.right(90)
turtle.forward(75)
```

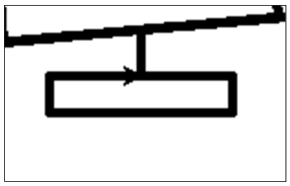




- Anything on the right side of # is ignored by Python
- So you can write anything you like on the right of an #
- We say those are comments

Example Code 2/2

```
# Return to the centre
turtle.left(90)
turtle.forward(75)
turtle.right(95)

# Draw the base
turtle.forward(25)
turtle.left(90)
turtle.forward(50)
turtle.right(90)
turtle.forward(20)
turtle.right(90)
turtle.forward(100)
turtle.right(90)
turtle.forward(20)
turtle.right(90)
turtle.forward(50)
```



Lifting Up and Putting Down the Pen

- By default, when the turtle moves, the pen draws
- If you want the turtle to move without drawing, you tell the turtle to lift up the pen:

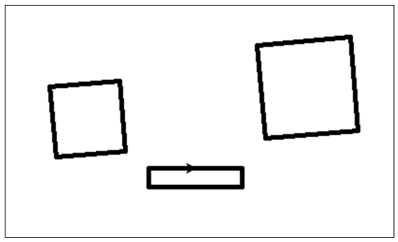
```
turtle.up()
```

- Nothing is drawn when the turtle moves after that
- To let the turtle draw again, tell the turtle to put the pen down using this code:

```
turtle.down()
```

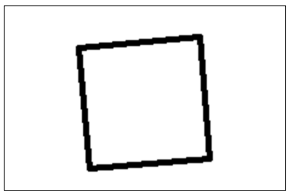
Another Example

- The following example is exactly the same as the previous example, but this time the pen has been lifted up and put down at appropriate places
- Result:



Pen Up/Down Example Code 1/2

```
turtle.left(5) # Tell the turtle to turn left 5 degrees

turtle.up() # Lift up the pen.

# From now on, the turtle will not draw anything when it moves.

turtle.forward(75) # Turtle walks forward but draws nothing

turtle.down() # Put down the pen

# The turtle draws again

# Draw the right box
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.right(90)
```



Pen Up/Down Example Code 2/2

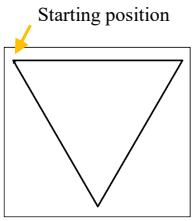
```
# Draw the left box
turtle.up()
turtle.forward(150)
turtle.down()

turtle.forward(75)
turtle.right(90)
turtle.forward(75)
turtle.right(90)
turtle.forward(75)
turtle.right(90)
turtle.forward(75)

# Return to the centre
turtle.up()
turtle.left(90)
turtle.forward(75)
turtle.right(95)
turtle.forward(25)
turtle.down()
```

```
# Draw the base
turtle.left(90)
turtle.forward(50)
turtle.right(90)
turtle.forward(20)
turtle.right(90)
turtle.forward(100)
turtle.right(90)
turtle.forward(20)
turtle.right(90)
turtle.forward(50)
```

Drawing Shapes

- We can draw lots of different shapes with lines
- Here is an example that draws a triangle:

```
turtle.forward(500)
turtle.right(120)
turtle.forward(500)
turtle.right(120)
turtle.forward(500)
```



- A hollow triangle is drawn

Drawing Solid Shapes

- You can tell the turtle to draw filled shapes
- Before drawing a shape, you need to run this line of code:

```
turtle.begin_fill()
```

- And after finishing the shape, use this line of code:

```
turtle.end_fill()
```

- The shape will then be immediately filled with the *fill colour*

Changing the Pen Colour

- You can easily change the colours
- The initial colour of the pen is black
- You can change the pen colour (=line colour) using the `turtle.color()` command
- Here is an example changing the pen colour to red:

```
turtle.color("red")
```

- And to change the pen colour to yellow:

```
turtle.color("yellow")
```

Changing the Fill Colour

- You can also change the fill colour using the `turtle.fillcolor()` command, like this:

```
turtle.fillcolor("yellow")
```

- If you want to, you can change both colours at the same time:

Line colour Fill colour

```
turtle.color("red", "yellow")
```

- After this line, every line drawn by the turtle will be red and every filled shape will be filled with yellow

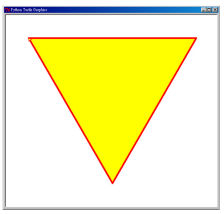
An Example of Using Colours

- This example illustrates how to draw a yellow triangle with a red border

```
turtle.color("red", "yellow")

turtle.begin_fill()

turtle.forward(500)
turtle.right(120)
turtle.forward(500)
turtle.right(120)
turtle.forward(500)

turtle.end_fill()
```

Draw the triangle

Fill the shape drawn between these two lines

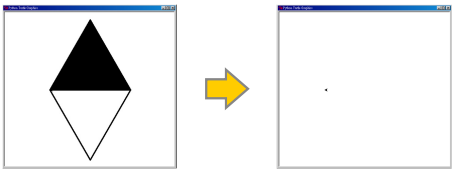


Clearing the Screen

- To clear everything drawn so far, run the following line of code:

```
turtle.clear()
```

- The position and orientation of the turtle remains the same as whatever it was before running the code



Speeding Up the Turtle

- The turtle initially moves fairly slowly
- You can adjust the moving speed of the turtle using the `turtle.speed()` command
- The initial speed value is 3
- For example, you can make the turtle move faster by running this line of code:

```
turtle.speed(8)
```


The Turtle Speed Value

- The speed value is an integer from 0 to 10
- The speed changes from very slow to very fast from 1 to 10, i.e.

1=slow speed … 6=normal speed … 10=fast speed

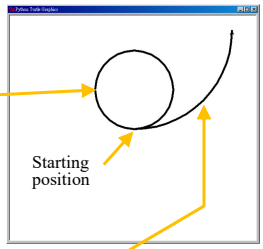
- A speed value of 0 is very fast
- If you use the value of 0 you will still see the turtle 'jump' immediately from one position to another instead of moving gradually

Drawing Circles and Arcs

- To draw a circle, use `circle()`:

```
turtle.circle( 100 )
```

radius=100 pixels



- You can also draw part of a circle:

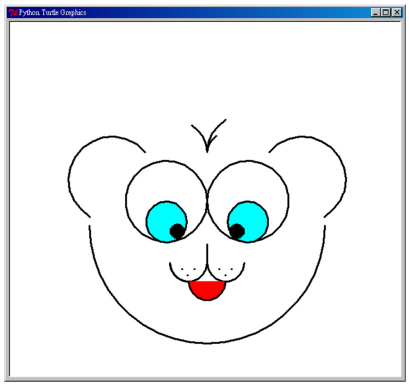
```
turtle.circle( 250 , 90 )
```

The centre of the circle is 250 pixels left of the turtle

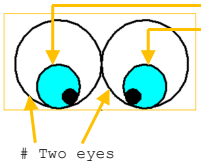
Draw 90 degrees of that circle

Drawing a Teddy Bear

- In this example, we will draw a cute teddy bear using all the commands we have learned



Teddy Bear Code 1/4



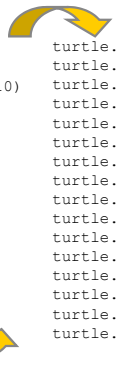
```
# Two eyes
turtle.circle(66)
turtle.right(180)
turtle.circle(66)
```

```
# Right eyeball
turtle.up()
turtle.left(90)
turtle.forward(33)
turtle.right(90)
turtle.forward(34)
turtle.down()
turtle.fillcolor("cyan")
turtle.begin_fill()
turtle.circle(33)
turtle.end_fill()
turtle.up()
turtle.forward(15)
turtle.left(90)
turtle.forward(4)
turtle.right(90)
turtle.down()
turtle.fillcolor("black")
turtle.begin_fill()
turtle.circle(11)
turtle.end_fill()
```

```
# Left eyeball
turtle.up()
turtle.goto(0, 0)
turtle.left(270)
turtle.forward(33)
turtle.left(90)
turtle.forward(34)
turtle.left(180)
turtle.down()
turtle.fillcolor("cyan")
turtle.begin_fill()
turtle.circle(33)
turtle.end_fill()
turtle.up()
turtle.backward(15)
turtle.right(90)
turtle.backward(4)
turtle.left(90)
turtle.down()
turtle.fillcolor("black")
turtle.begin_fill()
turtle.circle(11)
turtle.end_fill()
```

Teddy Bear Code 2/4

```
# Mouth
turtle.up()
turtle.goto(0, 0)
turtle.right(180)
turtle.forward(70)
turtle.down()
turtle.forward(30)
turtle.circle(30, 180)
turtle.up()
turtle.right(180)
turtle.circle(-30, -180)
turtle.left(180)
turtle.down()
turtle.circle(-30, 180)
turtle.right(180)
turtle.circle(30, 90)
turtle.right(90)
turtle.fillcolor("red")
turtle.begin_fill()
turtle.circle(30, 180)
turtle.end_fill()
```

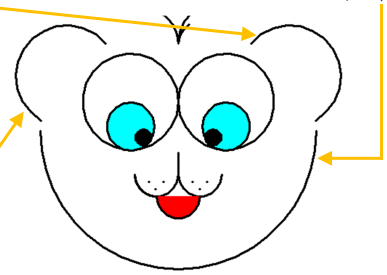


```
# Dots
turtle.up()
turtle.goto(0, 0)
turtle.backward(110)
turtle.left(90)
turtle.forward(20)
turtle.down()
turtle.forward(1)
turtle.up()
turtle.forward(19)
turtle.down()
turtle.forward(1)
turtle.up()
turtle.right(180)
turtle.forward(60)
turtle.down()
```

```
turtle.forward(1)
turtle.up()
turtle.forward(19)
turtle.down()
turtle.forward(1)
turtle.up()
turtle.right(90)
turtle.forward(10)
turtle.right(90)
turtle.forward(10)
turtle.down()
turtle.forward(1)
turtle.up()
turtle.forward(60)
turtle.down()
turtle.forward(1)
```

Teddy Bear Code 3/4

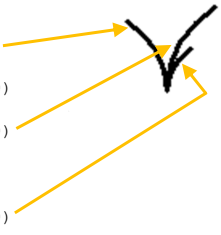
```
# Ears
turtle.up()
turtle.goto(0, 0)
turtle.right(90)
turtle.forward(80)
turtle.right(90)
turtle.forward(100)
turtle.down()
turtle.left(50)
turtle.circle(-70, 200)
turtle.up()
turtle.goto(0, 0)
turtle.right(120)
turtle.forward(80)
turtle.left(90)
turtle.forward(100)
turtle.down()
turtle.right(50)
turtle.circle(70, 200)
```



```
# Face
turtle.up()
turtle.goto(0, 0)
turtle.right(150)
turtle.forward(190)
turtle.left(90)
turtle.forward(40)
turtle.down()
turtle.circle(190, 180)
```

Teddy Bear Code 4/4

```
# Hair
turtle.up()
turtle.goto(0, 0)
turtle.forward(80)
turtle.down()
turtle.circle(50, 60)
turtle.right(180)
turtle.circle(-50, 60)
turtle.right(180)
turtle.circle(-60, 60)
turtle.left(180)
turtle.circle(60, 60)
turtle.left(180)
turtle.circle(-30, 60)
```



Hiding the Turtle

- After the turtle has finished drawing, the turtle is still shown
- It doesn't look good on top of the teddy bear image we have drawn
- We can make the turtle disappear:
  `turtle.hideturtle()`

Writing Text Using the Turtle

- You can display text using the turtle
- For example, you can use the code below to write the text 'COMP1021' in the turtle window:

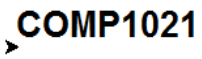
```
turtle.write("COMP1021")
```

- The text is shown all at once instead of being gradually drawn

Customising the Font

- If you want to, you can customise the font using the font option
- For example, a bigger and bold font can be used to write the text 'COMP1021' in the window

```
turtle.write("COMP1021",
    font=("Arial", 20, "bold"))
```



Using the bold 'Arial' font with a size of 20