COMP1021
Introduction to Computer Science

Turtle Shapes

Gibson Lam and David Rossiter

Outcomes

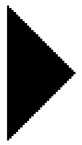
- After completing this presentation, you are expected to be able to:
  1. Change the shape of the turtle in turtle programming
  2. Adjust the size of the turtle

Turtle Shapes

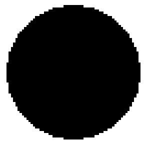
- There are several different shapes you can use for the turtle:
  - arrow, turtle, circle, square, triangle and classic
- You can also use any image in GIF format
- This means you can change the turtle shape according to the program you are creating
- For example, in a music program where the user see the turtle move, you could change the turtle to a musical note:

Turtle Shapes You Can Choose

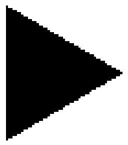
- Arrow



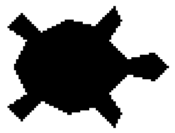
- Circle



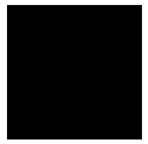
- Triangle



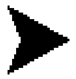
- Turtle



- Square



- Classic



The default shape of a turtle is "classic"

Changing the Turtle Shape

- To change the shape of the turtle you can use the following code:

  `turtle.shape(` *name of the shape* `)`

  where shape is one of the names of the shape listed in the previous slide

- For example:

```
turtle.shape("square")
```

changes the shape of the turtle to a square

Using Your Own Image

- Apart from the default turtle shapes you can also use any GIF image as your turtle shape
- For example, to use the GIF image on the right as the turtle shape you can use the following code:

ninja.gif

```
turtle.addshape("ninja.gif")
turtle.shape("ninja.gif")
```

Add the image to the turtle system so that it can then be selected as a turtle shape

Use the newly added shape (the image) as the turtle shape

GIF Images

- You have to use a GIF image, not other types
- GIF images have 256 different colours at most
- It has other limitations as well
- Usually these days you would choose the PNG image format instead of GIF format – but the PNG format isn't supported by `turtle.shape()`

- This program shows all the possibilities, one by one

```
import turtle

def draw():
    turtle.clear()
    for _ in range(4):
        turtle.forward(100)
        turtle.left(90)

def arrow_shape():
    turtle.shape("arrow")
    draw()

def circle_shape():
    turtle.shape("circle")
    draw()
```

```
def triangle_shape():
    turtle.shape("triangle")
    draw()

def turtle_shape():
    turtle.shape("turtle")
    draw()

def square_shape():
    turtle.shape("square")
    draw()

def classic_shape():
    turtle.shape("classic")
    draw()
```

```
def gif_shape():
    turtle.addshape("ninja.gif")
    turtle.shape("ninja.gif")
    draw()
```

Here the GIF file needs to be in the same directory as the Python program

```
# Start of the main program
print("Repeatedly press Enter to see a new shape")

arrow_shape()
input("Press Enter")
circle_shape()
input("Press Enter")
triangle_shape()
input("Press Enter")
turtle_shape()
input("Press Enter")
square_shape()
input("Press Enter")
classic_shape()
input("Press Enter")
gif_shape()
input("Press Enter")
turtle.done()
# End of
# program
```

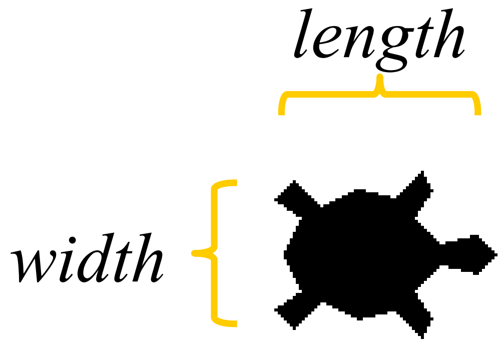
Changing the Size of Turtle Shapes

- Sometimes the turtle may look too small
- You can use `turtle.shapesize()` to make it bigger or smaller
- For example, you can double the size of a turtle shape using this code:

```
turtle.shapesize(2, 2)
```

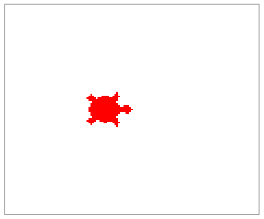
Multiply the width of the shape by 2

Multiply the length of the shape by 2

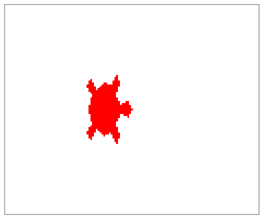


More Turtle Size Examples

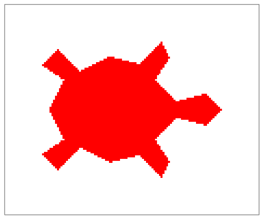
- Original turtle shape



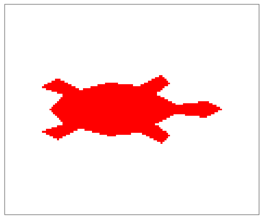
- `turtle.shapesize(2, 1)`



- `turtle.shapesize(4, 4)`



- `turtle.shapesize(2, 4)`



- `turtle.shapesize(3, 0.5)`