

Card 1 - Turtle Basics

Start your turtle program

★ What this does

- Import turtle before you use it
- Write one command on each line
- Run your code often to test it



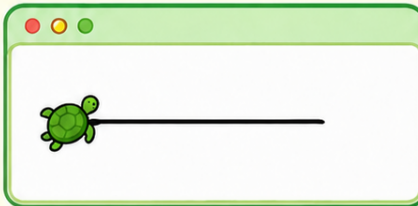
</> CODE EXAMPLE

```
import turtle

turtle.shape("turtle")
turtle.pensize(3)
```



Tip: Start small and test one line at a time.



You are ready to draw!

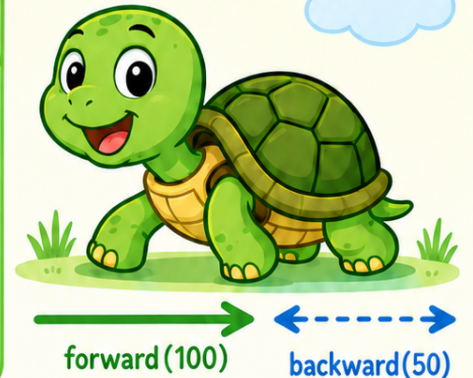
© James Abela 2024

Card 2 - Move the Turtle

Draw lines with forward and backward

★ What this does

- forward(n) moves forwards
- backward(n) moves backwards
- The number is the distance in pixels



</> CODE EXAMPLE

```
turtle.forward(100)
turtle.backward(50)
```



Tip: Start with short distances so you can see what happens.



The turtle can move!

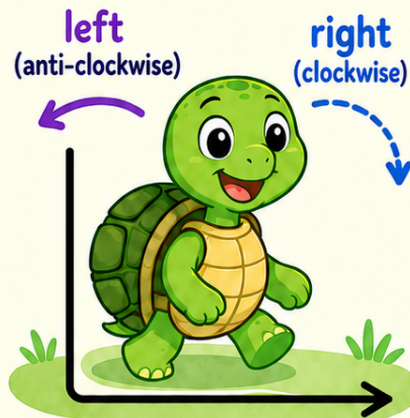
© James Abela 2024

Card 3 - Turn Corners

Change direction with left and right

★ What this does

- `right(angle)` turns clockwise
- `left(angle)` turns anti-clockwise
- A right angle is 90 degrees

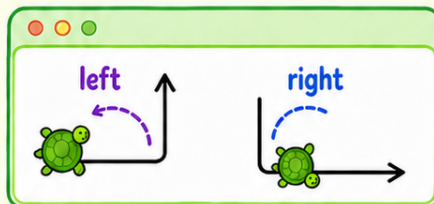


</> CODE EXAMPLE

```
turtle.right(90)
turtle.forward(80)

turtle.left(120)
```

Tip: If your shape goes the wrong way, try changing `right` to `left`.



★ Turning makes shapes! ➦

© James Abela 2024

Card 4 - Repeat with a Loop

Use a loop for shapes

★ What this does

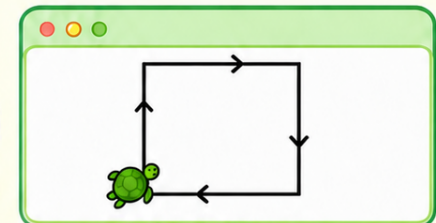
- A square repeats 4 times
- Indent the lines inside the loop
- Regular polygons use equal turns



</> CODE EXAMPLE

```
for side in range(4):
    turtle.forward(100)
    turtle.right(90)
```

Tip: The indented lines happen again and again.



★ Loops save time! ➦

© James Abela 2024

Card 5 - Lift the Pen

Move without drawing

★ What this does

- `penup()` lifts the pen
- `pendown()` starts drawing again
- Useful for separate shapes

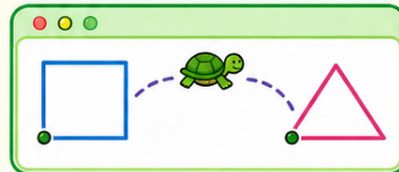


</> CODE EXAMPLE

```
turtle.penup()  
turtle.goto(80, 50)  
turtle.pendown()
```



Tip: Lift the pen before moving to a new place.



Now you can make separate shapes!

© James Abela 2024

Card 6 - Use Coordinates

Go to an exact position

★ What this does

- `goto(x, y)` moves to a point
- `x` changes left and right
- `y` changes up and down

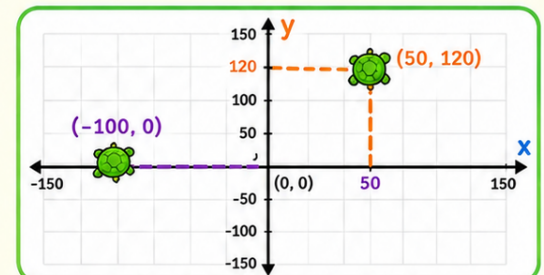


</> CODE EXAMPLE

```
turtle.goto(-100, 0)  
turtle.goto(50, 120)
```



Tip: Negative numbers go left or down.



Coordinates help you place shapes!

© James Abela 2024

Card 7 - Add Colour and Fill

Make your pictures brighter

★ What this does

- `pencolor()` changes the outline
- `fillcolor()` changes the inside colour
- Use `begin_fill()` and `end_fill()` around the shape



</> CODE EXAMPLE

```
turtle.pencolor("navy")
turtle.fillcolor("gold")
turtle.begin_fill()
# draw shape here
turtle.end_fill()
```



Tip: Start filling before you draw the shape.



Colour makes Turtle art pop!

© James Abela 2024

Card 8 - Draw Circles

Make circles and flower patterns

★ What this does

- `circle(radius)` draws a full circle
- Turn a little after each circle for patterns
- Positive radius works well for beginners

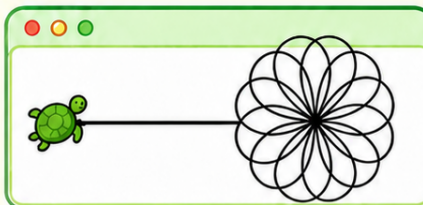


</> CODE EXAMPLE

```
for petal in range(12):
    turtle.circle(60)
    turtle.right(30)
```



Tip: Small turns can make beautiful patterns.



Circles can become art!

© James Abela 2024

Card 9 - Make a Procedure

Reuse code with a named block

★ What this does

- A procedure groups instructions
- A parameter lets you change the size
- Call the procedure again and again

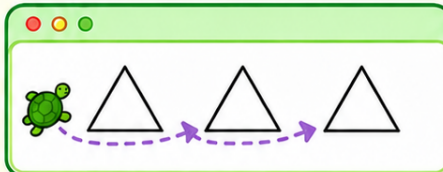


</> CODE EXAMPLE

```
def triangle(size):  
    for side in range(3):  
        turtle.forward(size)  
        turtle.right(120)  
triangle(60)
```



Tip: Write the procedure once, then call it many times.



Procedures make coding easier!

© James Abela 2024

Card 10 - Stuck? Check This

Quick debugging ideas

★ What this does

- Check your brackets and spelling
- Check your indentation in loops and procedures
- Read the error message from the top line first

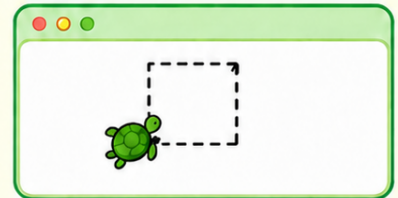


</> CODE EXAMPLE

```
for side in range(4):  
    turtle.forward(100)  
    turtle.right(90)
```



Tip: Change one thing, run it, then check again.



You can fix it!

© James Abela 2024