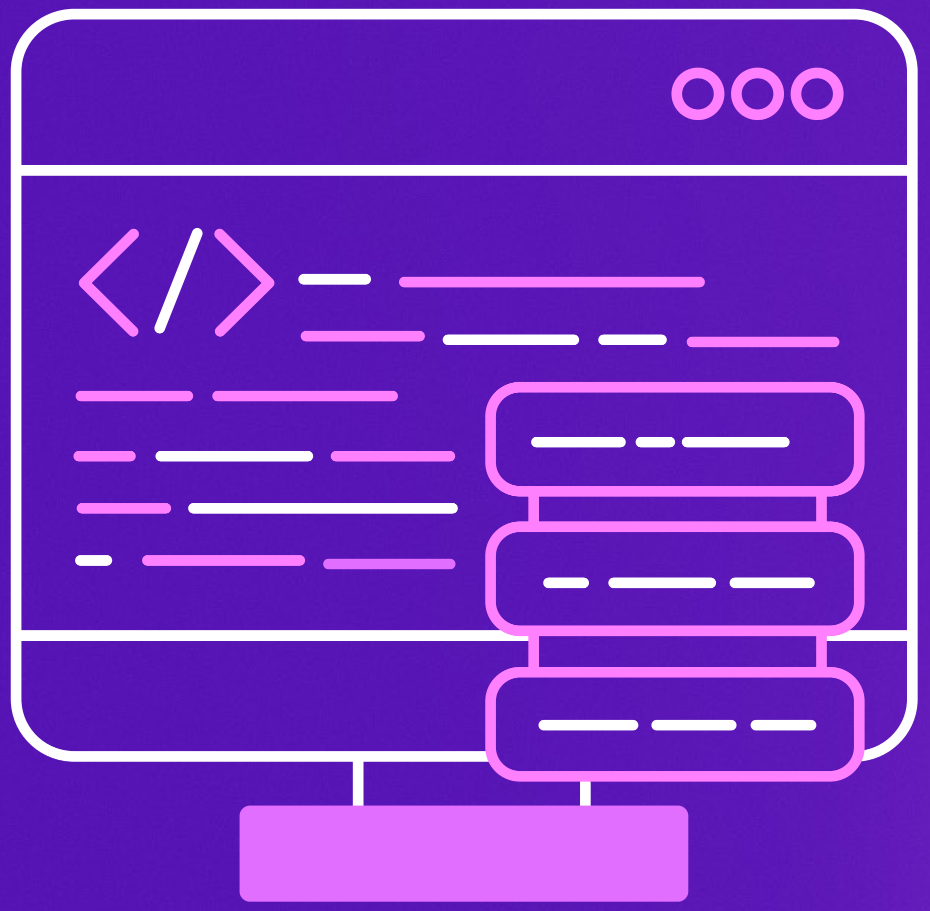
Technology

Software & Programming

```
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

#selection at the end -add back the deselected
mirror_ob.select= 1
modifier_ob.select=1
y.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob
#mirror_ob.select = 0
one = bpy.context.selected_objects[0]
bpy.data.objects[one.name].select = 1
```

Code That Keeps The World Turning



Software makes the world go round. It's increasingly used across more and more of our everyday items, from mobile phones to TVs and even cars - they all have software that control how they work.

To make that software you need to programme it by writing code, as computers can't understand our human language. A bit like if you tried to talk to an Alien who didn't speak English… you wouldn't be able to communicate. So computers have their own language instead, called code. There are lots of different types of computer code known as programming languages. Therefore, in order for us to get the computer to do what we want it to do – we have to translate, or change, our instructions into the computer's language so they understand it.

Code is basically a set of algorithms, which are rules or instructions that are followed in a sequence to complete a task, and that task could be anything. We all use algorithms on a daily basis as humans, such as following a recipe to make a meal. When the computer runs the code, it follows the 'recipe', step-by-step.

Have you ever wondered how your games console works? Well they are also computers which run software. Think about the controller. When you play your game, you tell it what to do by pressing buttons on the controller. The controller then acts as the translator and coverts your instructions (algorithm) into code that the console can understand. That way you can move the player on the game – it's super clever!

There might seem like there are too many programming languages out there, and you might wonder which one you should learn? But don't worry about learning the 'right' one. There isn't a 'wrong' one to learn. If you know the principles of programming then it doesn't matter how technology changes or which language you are asked to use, you will be able to read the code and program.

Today's activity will get you thinking all about code and the various programming languages that exist. Notice how they all look different, but follow the same key fundamentals.

Find Out More

- Computer Science journeys
  https://atadastral.co.uk/go/bswtf1
- Women in Computing
  https://atadastral.co.uk/go/bswtf2
- Simple coding
  https://atadastral.co.uk/go/bswtf3
- Introduction to computational thinking
  https://atadastral.co.uk/go/bswtf4

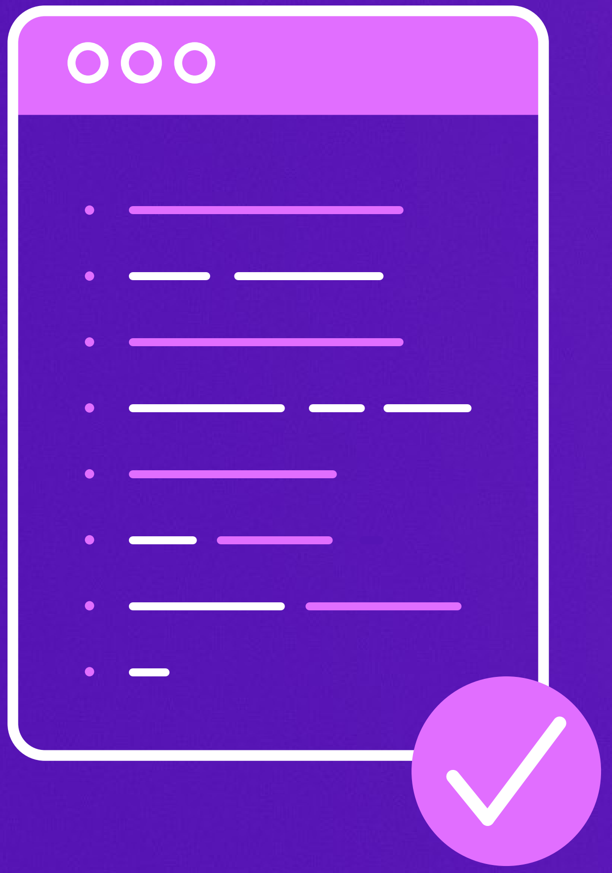
Have a Go

- Teach creative programming with Turtle power!
  https://atadastral.co.uk/go/bswth1
- Blockly Games
  https://atadastral.co.uk/go/bswth2
- Make things with code
  https://atadastral.co.uk/go/bswth3
- Write some code to run on the Space Station
  https://atadastral.co.uk/go/bswth4
- Can you rescue the diamond?
  https://atadastral.co.uk/go/bswth5
- GCSE transition - Programming Concepts
  https://atadastral.co.uk/go/bswth6
- Programming concepts
  https://atadastral.co.uk/go/bswth7
- Procedural programming
  https://atadastral.co.uk/go/bswth8
- The Jewels of Heuro
  https://atadastral.co.uk/go/bswth9

Teacher Links

- Raspberry Pi - Curriculum
  https://atadastral.co.uk/go/bswtt1
- Barefoot – Supporting primary school teaching
  https://atadastral.co.uk/go/bswtt2
- Teach Computing – Curriculum teaching resources
  https://atadastral.co.uk/go/bswtt3
- Teach Computing – Helping you teach computing
  https://atadastral.co.uk/go/bswtt4

Activity 1
Analysing Code Snippets



For this activity, we've pulled together lots of different code snippets from various programming languages. Each snippet of code makes a different shape, and your challenge is to work out what those shapes are by investigating the code.

You may have never programmed in these languages before, or even heard of them. But don't worry. Use the basic principles of programming and you'll be able to work out the shapes that these code snippets create.

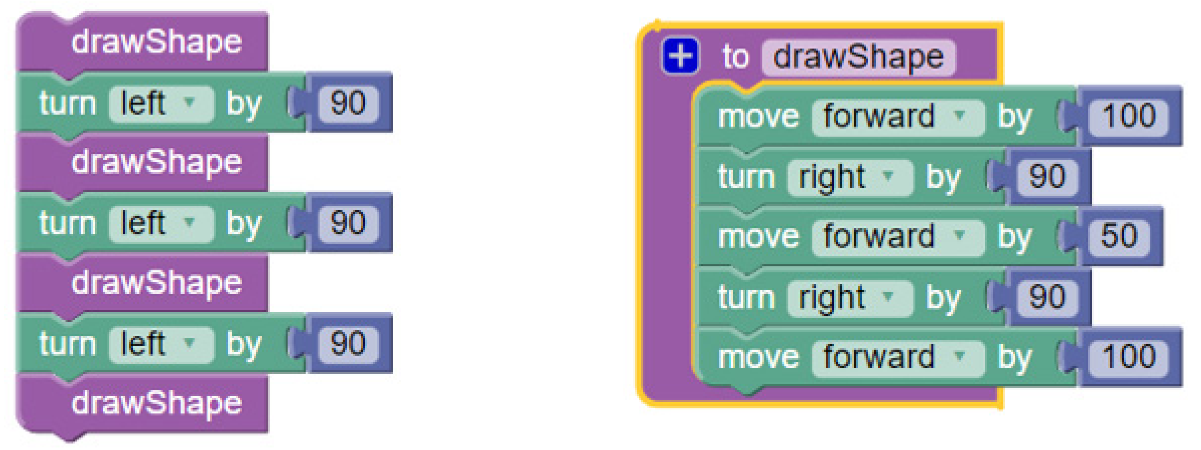
If it helps, try drawing the shapes out on some paper. If you go wrong, try troubleshooting and always go back to the basic fundamentals first. If you're completely stuck, try asking your partner or teacher for some help.

Now take a look at these code snippets and work out the shapes they create, good luck!

Task 1 - Logo

```
clearscreen
pendown
forward 100
right 90
forward 100
right 90
forward 100
Right 90
forward 100
penup
```

Task 2 - Blocky



```
drawShape
turn left by 90
drawShape
turn left by 90
drawShape
turn left by 90
drawShape
```

```
to drawShape
    move forward by 100
    turn right by 90
    move forward by 50
    turn right by 90
    move forward by 100
```

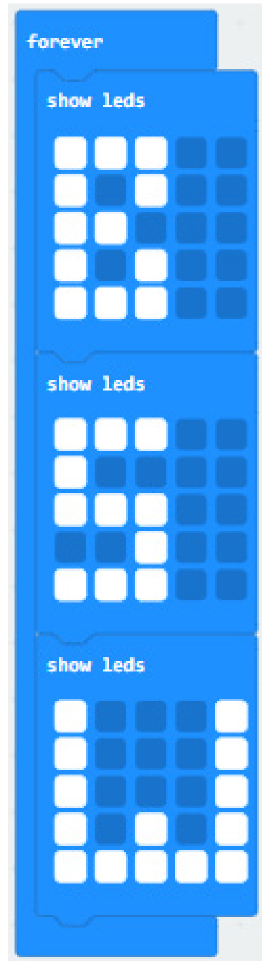
Task 3 - Python

```
import turtle
drawing = turtle.Turtle()
drawing.forward(100)
drawing.right(120)
drawing.forward(100)
drawing.right(120)
drawing.forward(100)
```

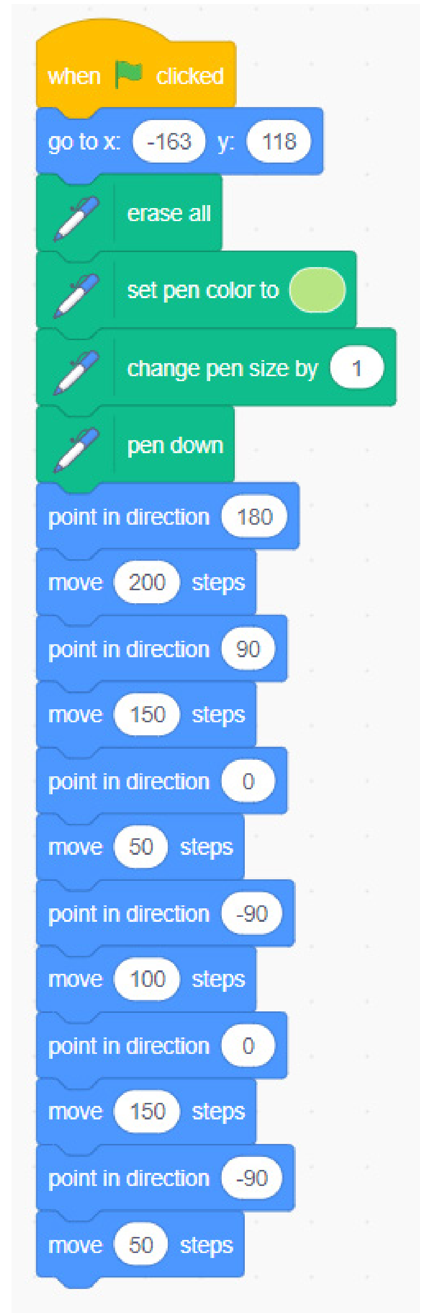
Task 4 - JavaScript

```
input.onButtonPressed(Button.A, function() {basic.showIcon(IconNames.Happy)})
input.onButtonPressed(Button.B, function(){basic.showIcon(IconNames.Sad)})
```

Task 3 - Micro:Bit



Task 6 - Scratch



```
when [flag] clicked
go to x: -163 y: 118
erase all
set pen color to
change pen size by 1
pen down
point in direction 180
move 200 steps
point in direction 90
move 150 steps
point in direction 0
move 50 steps
point in direction -90
move 100 steps
point in direction 0
move 150 steps
point in direction -90
move 50 steps
```

Each code snippet looks different, some are quite visual and some are more text-based. Which did you find easiest to work out? Which of the six programming languages do you prefer?

Challenge: Have a go at coding these snippets for real.

Extension: Have a go at making your own code snippets to challenge your partner with.

If you would like the answers to the activities, please email computerscience@bt.com stating your school and key stage.