Draw a snowflake

Code Playground

December 2022

BARCLAYS

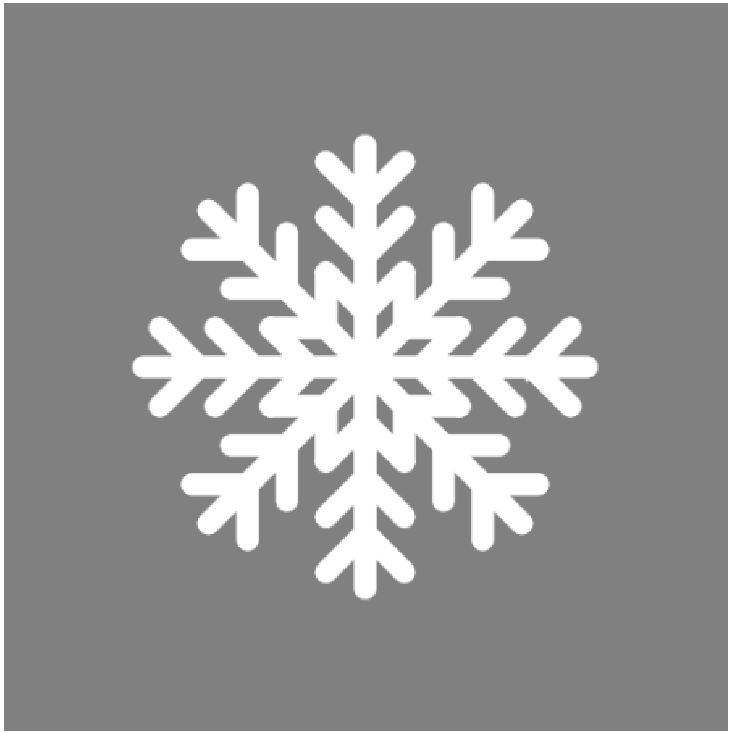
Introduction to EduBlocks

EduBlocks is a free tool that helps anyone learn how to code with text-based languages like Python or HTML, using familiar block coding as used in Scratch.

This session will allow you to use blocks to create a snowflake and see what block coding looks like in the Python coding language.

The project is designed for EduBlocks 4.0 and will show you what code is required to draw a simple snowflake pattern.

All you need to get started is an internet browser and access to EduBlocks.



◀ Previous Next ▶

Step 1

"Turtle" is a Python feature like a drawing board, which lets us command a pen to draw and move around the board using set functions.

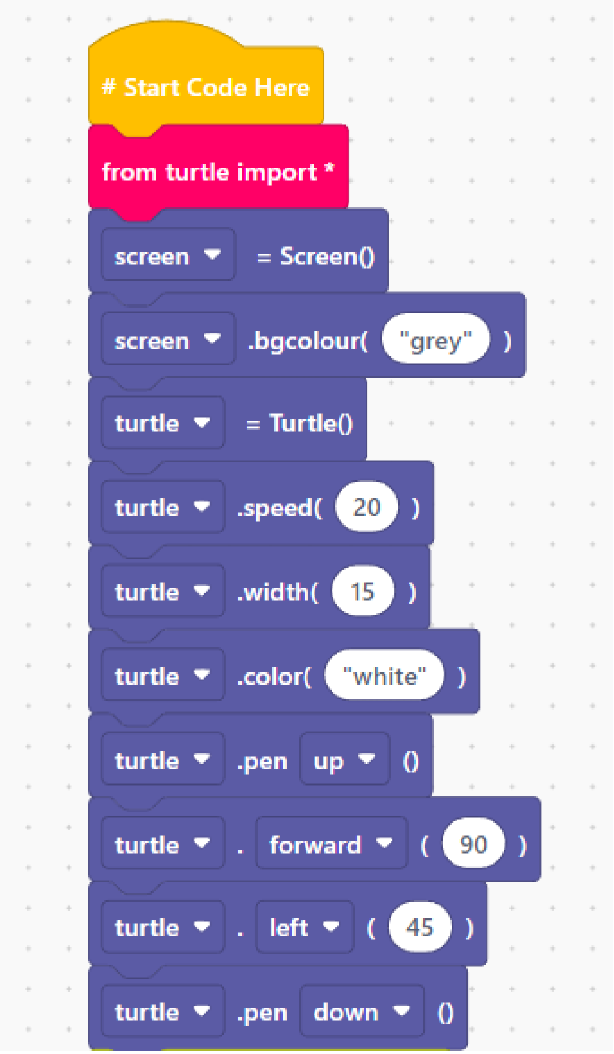
To get started, we need to build the screen background for our pen, or turtle as we'll now call it.

Import the turtle, then we need to identify the turtle and screen settings.

We've changed the screen background colour to **"grey."**

Next, we'll define the settings of the turtle to control the:

- Speed of the turtle
- How thick we'd like the lines to be drawn
- The colour of the lines
- The direction of the turtle



```
# Start Code Here
from turtle import *
screen = Screen()
screen .bgcolour( "grey" )
turtle = Turtle()
turtle .speed( 20 )
turtle .width( 15 )
turtle .color( "white" )
turtle .pen up ()
turtle . forward ( 90 )
turtle . left ( 45 )
turtle .pen down ()
```

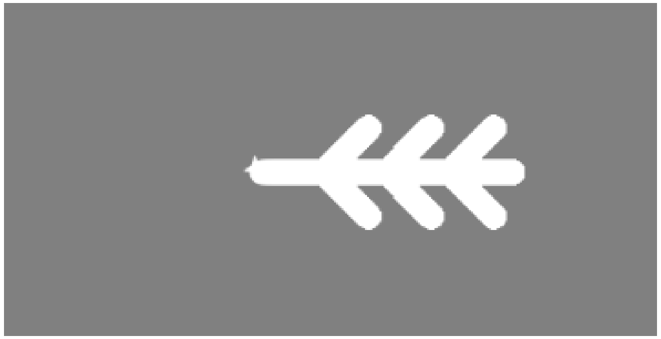
◀ Previous Next ▶

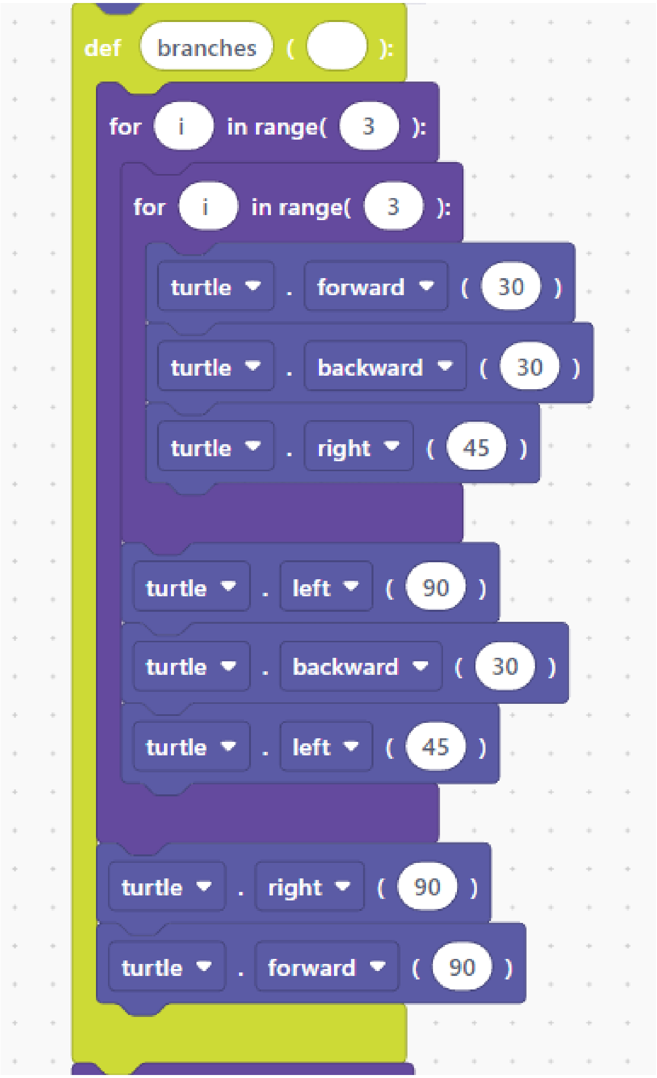
Step 2

We're now ready to code the branches of our snowflake, using definition, loop and turtle movement blocks.

This section of code will tell the turtle how to draw 1 branch of our snowflake structure.

Start with a **definition** block, and change the word **function** to **snowflake.** We can then code loops to control the turtle's direction to draw a branch of our snowflake, again coding the size and directions of the lines.





```
def branches ( ):
    for i in range( 3 ):
        for i in range( 3 ):
            turtle . forward ( 30 )
            turtle . backward ( 30 )
            turtle . right ( 45 )
        turtle . left ( 90 )
        turtle . backward ( 30 )
        turtle . left ( 45 )
    turtle . right ( 90 )
    turtle . forward ( 90 )
```

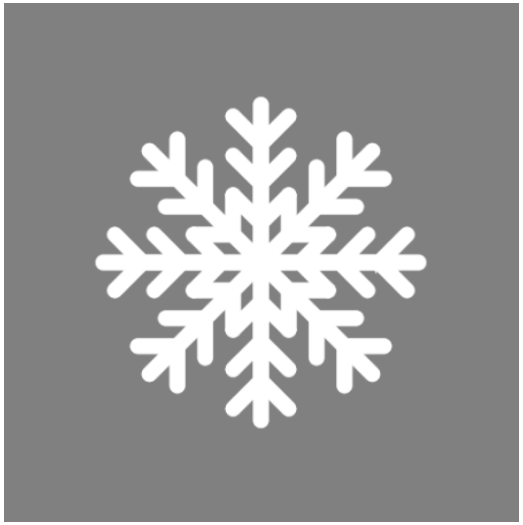
◀ Previous Next ▶

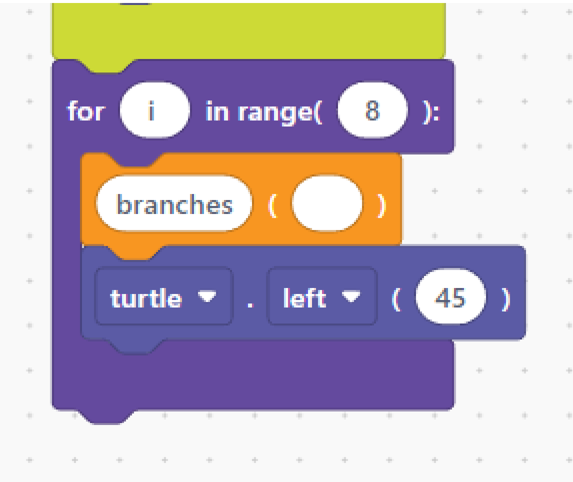
Step 3

Now we've defined the code for 1 branch of our snowflake, we now need to add more to make our final snowflake pattern.

For the final piece of code, we add a loop and repeat the branch code 8 times, moving the turtle 45 degrees after each branch.

If you run your code now, your turtle should draw a snowflake!

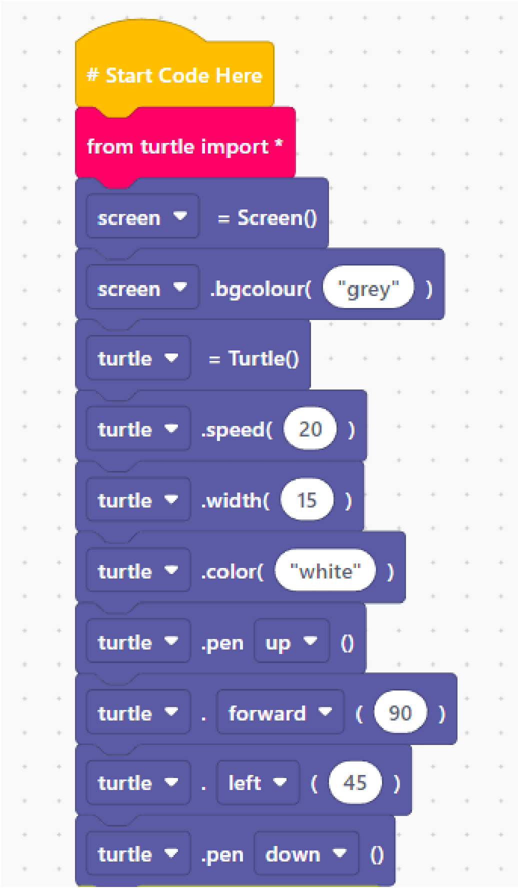




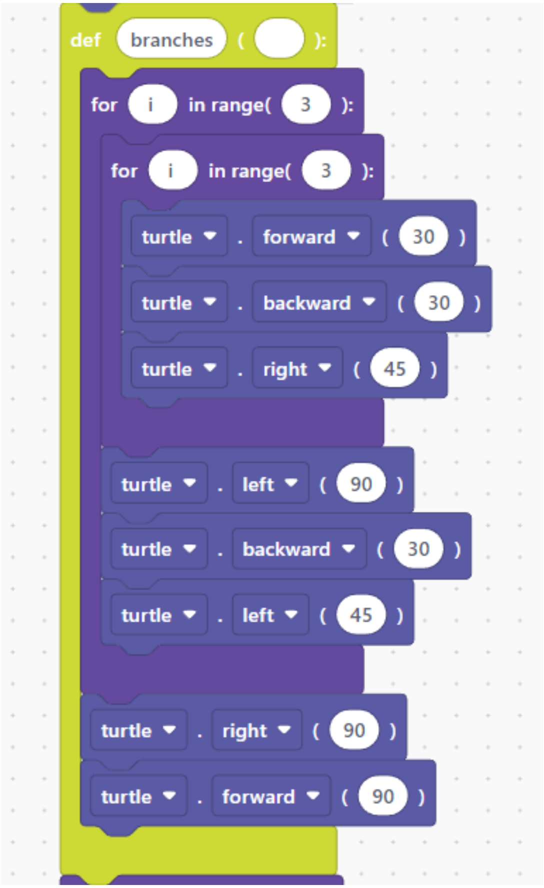
```
for i in range( 8 ):
    branches ( )
    turtle . left ( 45 )
```

◀ Previous Next ▶

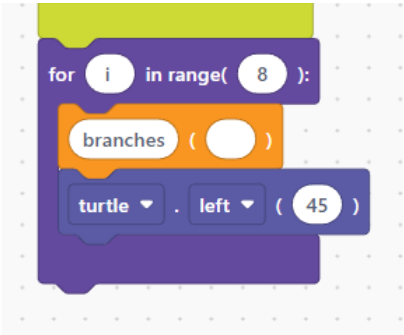
Full block code



```
# Start Code Here
from turtle import *
screen = Screen()
screen .bgcolour( "grey" )
turtle = Turtle()
turtle .speed( 20 )
turtle .width( 15 )
turtle .color( "white" )
turtle .pen up ()
turtle . forward ( 90 )
turtle . left ( 45 )
turtle .pen down ()
```



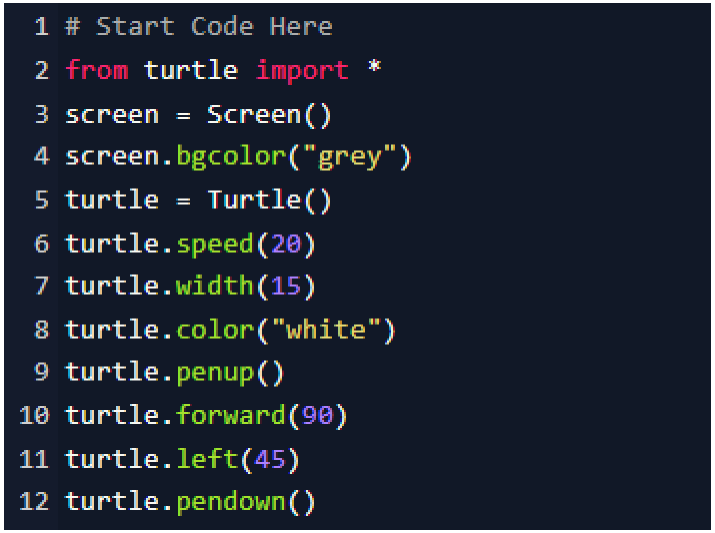
```
def branches ( ):
    for i in range( 3 ):
        for i in range( 3 ):
            turtle . forward ( 30 )
            turtle . backward ( 30 )
            turtle . right ( 45 )
        turtle . left ( 90 )
        turtle . backward ( 30 )
        turtle . left ( 45 )
    turtle . right ( 90 )
    turtle . forward ( 90 )
```



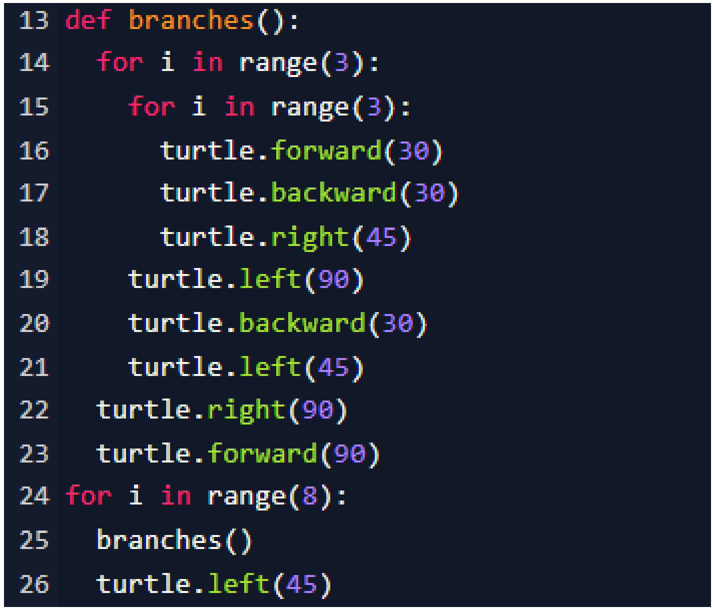
```
for i in range( 8 ):
    branches ( )
    turtle . left ( 45 )
```

◀ Previous Next ▶

Full Python code



```python
# Start Code Here
from turtle import *
screen = Screen()
screen.bgcolor("grey")
turtle = Turtle()
turtle.speed(20)
turtle.width(15)
turtle.color("white")
turtle.penup()
turtle.forward(90)
turtle.left(45)
turtle.pendown()
```



```python
def branches():
  for i in range(3):
    for i in range(3):
      turtle.forward(30)
      turtle.backward(30)
      turtle.right(45)
    turtle.left(90)
    turtle.backward(30)
    turtle.left(45)
  turtle.right(90)
  turtle.forward(90)
for i in range(8):
  branches()
  turtle.left(45)
```

◀ Previous