

Student workbook
EduBlocks
Turtle Race

Code Playground



EduBlocks Turtle Race

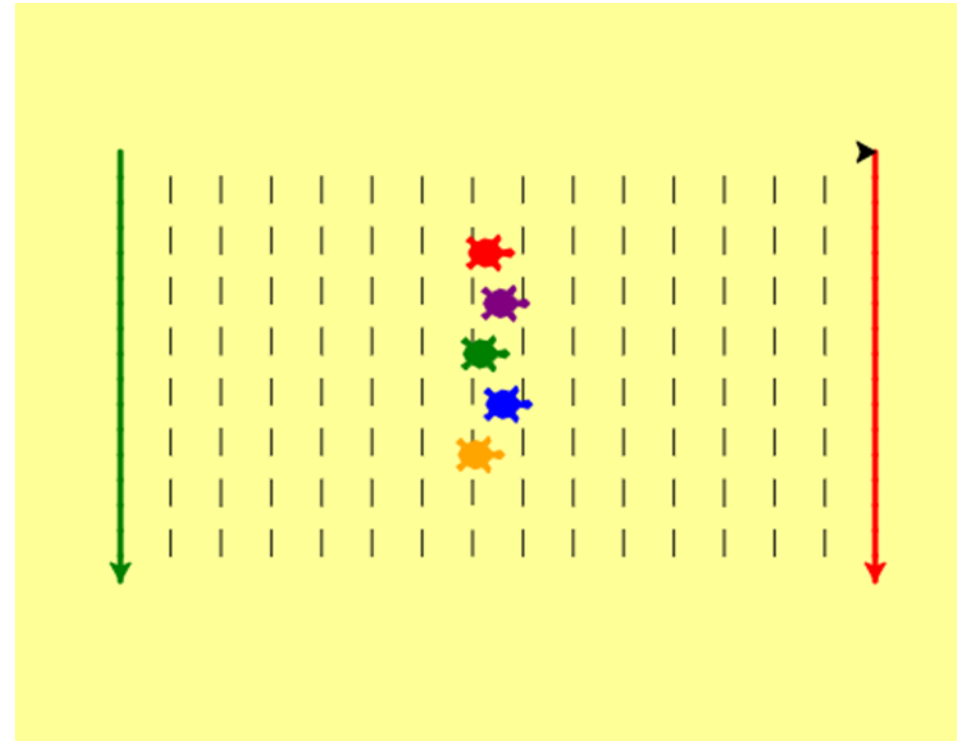
EduBlocks project

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

This session will allow you to use blocks to create a turtle race and be able to see what block coding looks like in Python coding language.

This project is designed for EduBlocks 4.0.

Hope you enjoy the project!



EduBlocks Turtle Race

EduBlocks project

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 background and create a list of colours for our pen, or turtle as we will now call it.

Import the turtle, then import random as we will need this later to move our turtles randomly, so any turtle could win the race.

The rest of the code in this step uses turtle blocks to set up the background screen and position our turtle, ready to start drawing the race track. We have added a new variable called start to identify where we would like the start line to be drawn, using x coordinates of -140 and y coordinates of 140 on the screen.

Python code

```
1 # Start Code Here
2 from turtle import *
3 import random
4 screen = Screen()
5 screen.bgcolor(255, 255, 152)
6 start = Turtle()
7 start.penup()
8 start.goto(-140, 140)
9 start.width(3)
10 start.right(90)
11 start.pendown()
```

Block code

The image shows a sequence of code blocks in a block-based programming environment. The blocks are as follows:

- A yellow block with the text "# Start Code Here".
- A pink block with the text "from turtle import *".
- A pink block with the text "import random".
- A blue block with a dropdown menu labeled "screen" and the text "= Screen()".
- A blue block with a dropdown menu labeled "screen" and the text ".bgcolor(255, 255, 152)".
- A blue block with a dropdown menu labeled "start" and the text "= Turtle()".
- A blue block with a dropdown menu labeled "start" and the text ".pen up (0)".
- A blue block with a dropdown menu labeled "start" and the text ".goto(-140, 140)".
- A blue block with a dropdown menu labeled "start" and the text ".width(3)".
- A blue block with a dropdown menu labeled "start" and the text ". right (90)".
- A blue block with a dropdown menu labeled "start" and the text ".pen down (0)".

EduBlocks Turtle Race

EduBlocks project

Step 2

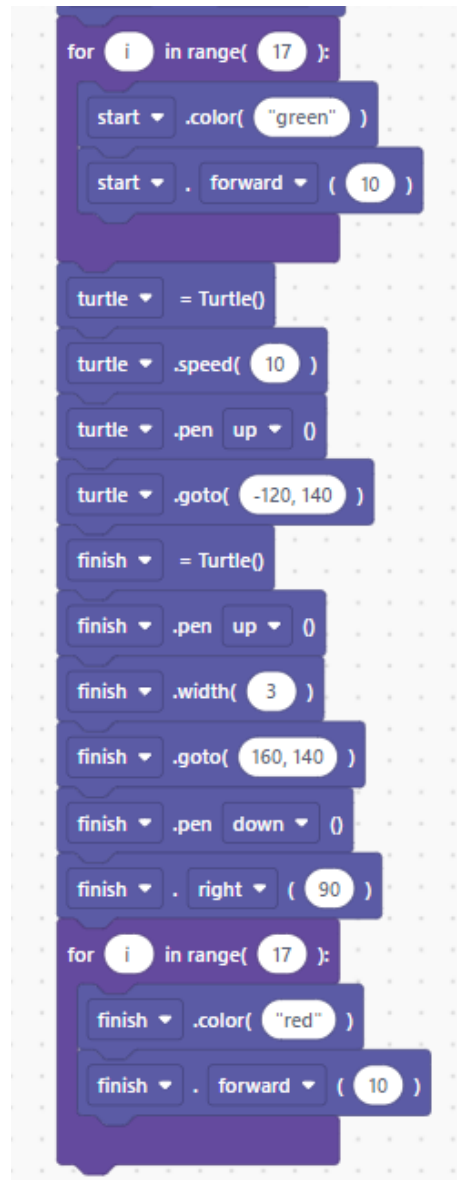
Now we are ready to draw the race track's Start and Finish lines, by introducing loops to set parameters and tell the turtle where to go and what to draw.

The first loop is the code needed to draw your start line. Don't forget to use the quotation marks around the word "green."

The following turtle blocks outside of the loop are used to set the position of the turtle to draw the finish line. We have introduced another new variable called finish to identify this code is for the finish line.

Another loop is then used to replicate the start line code, but using the finish variable and changing the colour to red.

Block code



```
for i in range( 17 ):
  start .color( "green" )
  start . forward ( 10 )

turtle = Turtle()
turtle .speed( 10 )
turtle .pen up 0
turtle .goto( -120, 140 )

finish = Turtle()
finish .pen up 0
finish .width( 3 )
finish .goto( 160, 140 )
finish .pen down 0
finish . right ( 90 )

for i in range( 17 ):
  finish .color( "red" )
  finish . forward ( 10 )
```

Python code

```
12 for i in range(17):
13     start.color("green")
14     start.forward(10)
15 turtle = Turtle()
16 turtle.speed(10)
17 turtle.penup()
18 turtle.goto(-120, 140)
19 finish = Turtle()
20 finish.penup()
21 finish.width(3)
22 finish.goto(160, 140)
23 finish.pendown()
24 finish.right(90)
25 for i in range(17):
26     finish.color("red")
27     finish.forward(10)
```

EduBlocks Turtle Race

EduBlocks project

Step 3

Next, we will draw the race track, again using loops to set parameters and tell the turtle where to go.

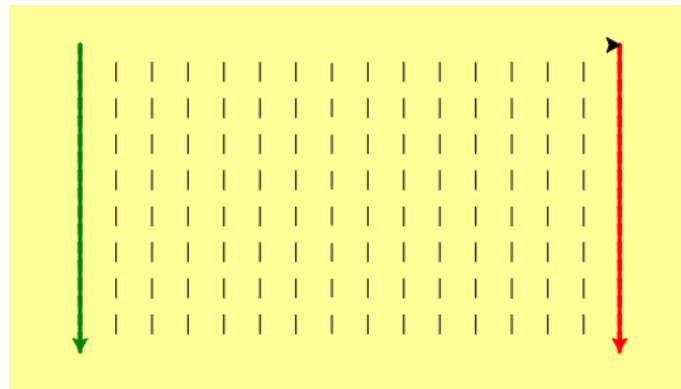
In the first loop, we have changed the `i` to the word `step` and added the number “14.” This will tell the turtle that we would like 14 columns drawing across the screen. The rest of the code on this page will be inside this loop.

In the second loop, we changed the `i` to the word `num` and added the number 8. This will tell the turtle that we would like 8 rows drawing.

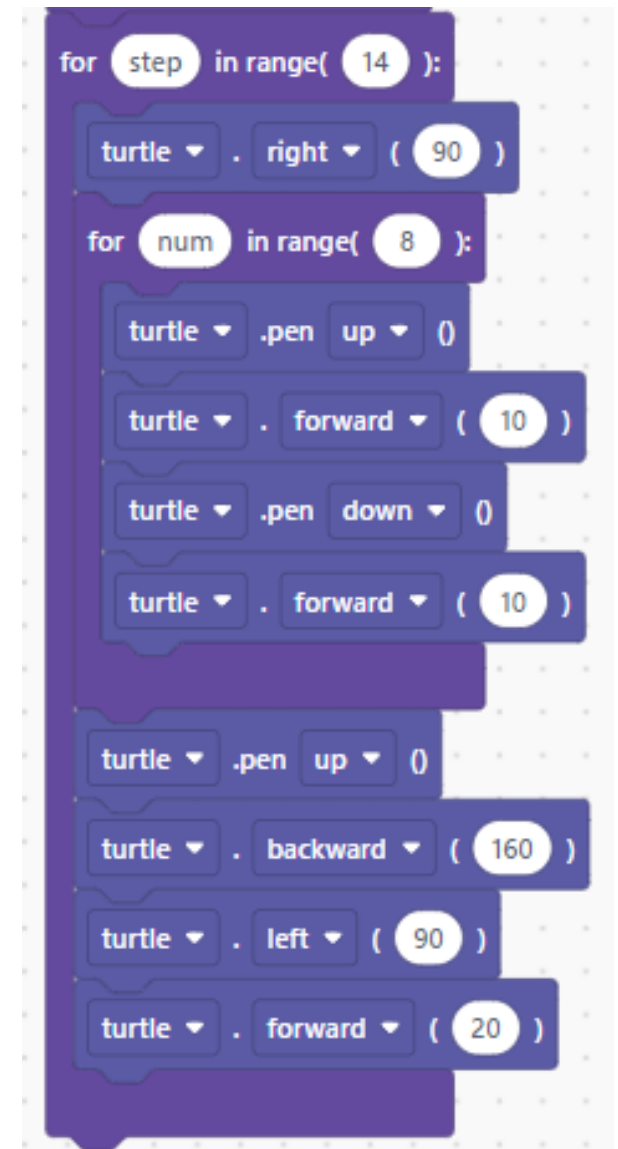
The last turtle blocks will tell the turtle to draw a dotted line, to start building our race track. When you run the code now, you should see the turtle draw the start and finish line, then the rest of the track; 8 rows down and 14 columns across.

Python code

```
28 for step in range(14):
29     turtle.right(90)
30     for num in range(8):
31         turtle.penup()
32         turtle.forward(10)
33         turtle.pendown()
34         turtle.forward(10)
35     turtle.penup()
36     turtle.backward(160)
37     turtle.left(90)
38     turtle.forward(20)
```



Block code



EduBlocks Turtle Race

EduBlocks project

Step 4

The next step is to create the 5 racing turtles, by giving them a name, setting their colour and their shape.

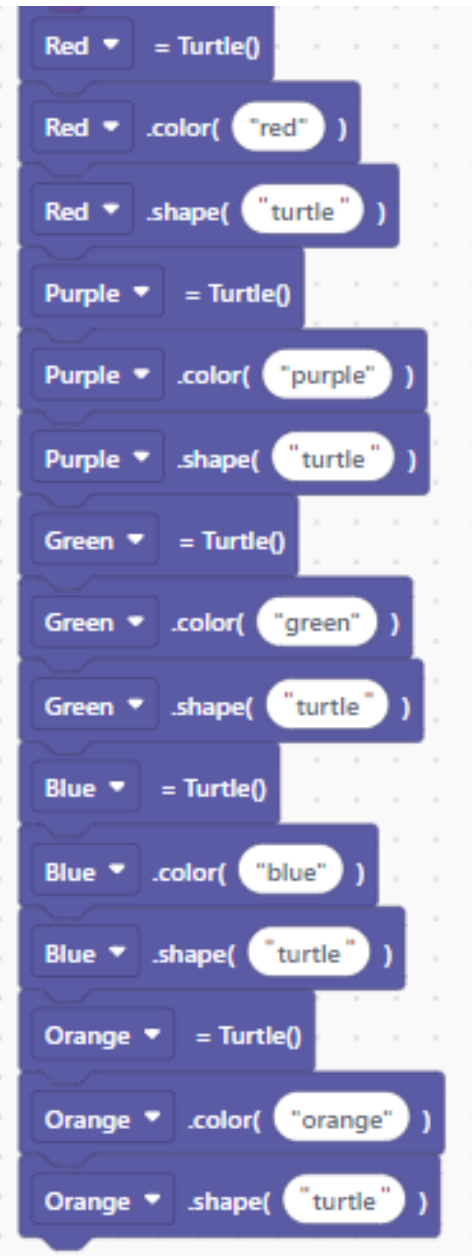
We need to add 3 blocks of code for each of our 5 racing turtles to identify it, change the colour and make the turtle an actual turtle shape.

Change the name of the turtle, by selecting the dropdown and adding a new variable. This will allow you to code each turtle differently, which is important when it comes to positioning them for the race. We have called ours by the name of the colours we are using for each turtle.

Repeat this code for the remaining 4 turtles, remembering to add a new variable for the different coloured turtles (purple, green, blue and orange).

Once you have completed this section of code, you should have a race track and 5 turtles on your screen. The turtles will all appear in the centre of your screen, so the next piece of code is to move them to the start line.

Block code



The image shows a vertical stack of 15 code blocks in a block-based programming environment. Each turtle is created with three blocks: a variable creation block, a color setting block, and a shape setting block. The turtles are named Red, Purple, Green, Blue, and Orange in that order from top to bottom.

```
Red = Turtle()
Red.color("red")
Red.shape("turtle")
Purple = Turtle()
Purple.color("purple")
Purple.shape("turtle")
Green = Turtle()
Green.color("green")
Green.shape("turtle")
Blue = Turtle()
Blue.color("blue")
Blue.shape("turtle")
Orange = Turtle()
Orange.color("orange")
Orange.shape("turtle")
```

Python code

```
39 Red = Turtle()
40 Red.color("red")
41 Red.shape("turtle")
42 Purple = Turtle()
43 Purple.color("purple")
44 Purple.shape("turtle")
45 Green = Turtle()
46 Green.color("green")
47 Green.shape("turtle")
48 Blue = Turtle()
49 Blue.color("blue")
50 Blue.shape("turtle")
51 Orange = Turtle()
52 Orange.color("orange")
53 Orange.shape("turtle")
```

EduBlocks Turtle Race

EduBlocks project

Step 5

Now we have our 5 racing turtles, we need to set their position on the start line.

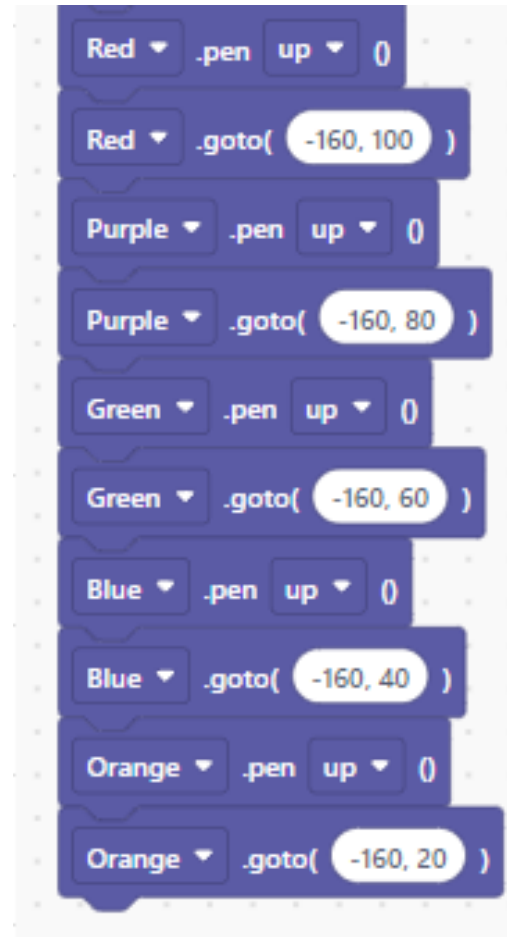
The following code is completed using the turtle blocks.

We need to change the variable and position for each of the turtles.

Repeat this code for the remaining 4 turtles, but reduce the last number by 20, so each of your turtles starts in a different position on the start line.

When you run your full code now, this will move all your turtles to start line.

Block code



Python code

```
54 Red.penup()  
55 Red.goto(-160, 100)  
56 Purple.penup()  
57 Purple.goto(-160, 80)  
58 Green.penup()  
59 Green.goto(-160, 60)  
60 Blue.penup()  
61 Blue.goto(-160, 40)  
62 Orange.penup()  
63 Orange.goto(-160, 20)
```

EduBlocks Turtle Race

EduBlocks project

Step 6

The next step is tell the racing turtles which direction to move and how fast to go.

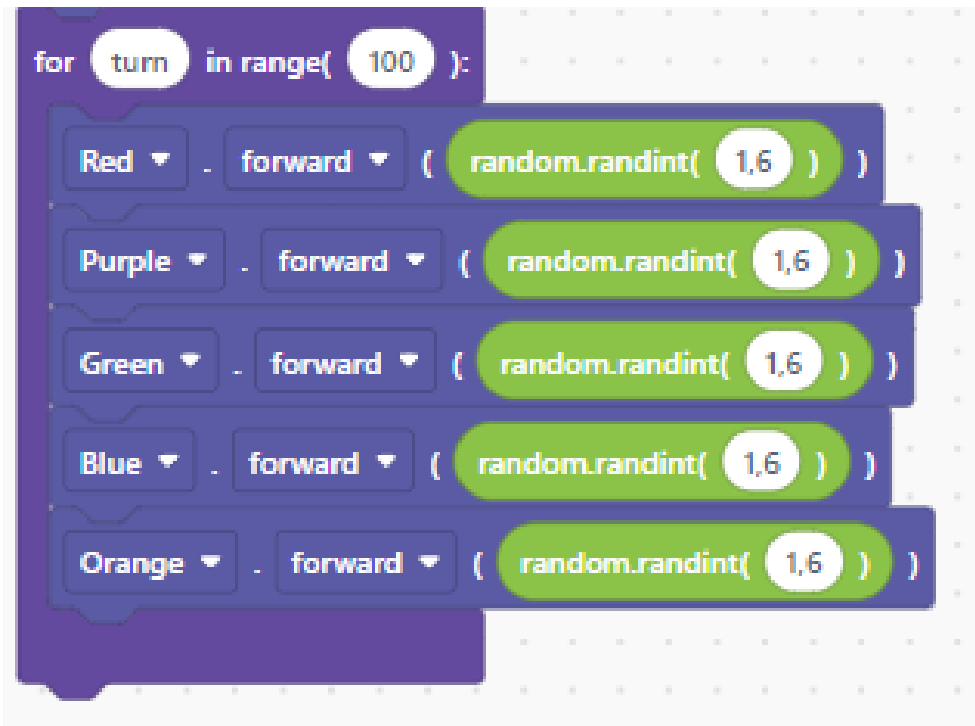
We are going to code the next part in another loop and tell each of our turtles to move forward. Change the name of each of your turtles to the variables you have added previously.

If you run the code now, you will see it is not a very good race, as each turtle is just moving forward in turn.

We will make the race better by introducing random movement. The numbers 1, 6 will tell the turtles to move randomly between 1 and 6 pixels across the screen, to make the race more exciting.

If we run the code now, the turtles race randomly, but do not stop at the finish line. The next piece of code will make the winning turtle stop at the finish line and declare the winner.

Block code

A screenshot of the EduBlocks code editor showing a loop of 100 iterations. Inside the loop, five code blocks are stacked vertically, each representing a different turtle: Red, Purple, Green, Blue, and Orange. Each block contains a 'forward' block followed by a 'random.randint(1,6)' block. The blocks are connected by a vertical line on the left side of the loop structure.

```
for turn in range( 100 ):
  Red . forward ( random.randint( 1,6 ) )
  Purple . forward ( random.randint( 1,6 ) )
  Green . forward ( random.randint( 1,6 ) )
  Blue . forward ( random.randint( 1,6 ) )
  Orange . forward ( random.randint( 1,6 ) )
```

Python code

```
64 for turn in range(100):
65     Red.forward(random.randint(1,6))
66     Purple.forward(random.randint(1,6))
67     Green.forward(random.randint(1,6))
68     Blue.forward(random.randint(1,6))
69     Orange.forward(random.randint(1,6))
```


EduBlocks Turtle Race

EduBlocks project

Step 7

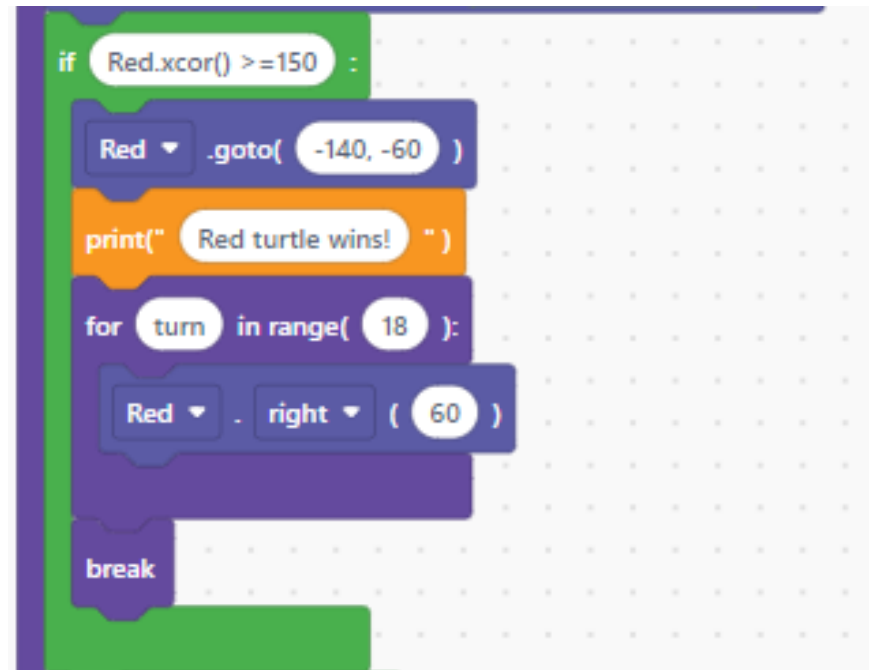
Now we need to tell the racing turtles, where the finishing line is, inform us of who has won the race and make the winning turtle do a little celebration! The remaining code is all inside the loop we created in Step 6.

We now need to add an if logic block to our code, to define what happens after a particular outcome. The `Red.xcor() >=150` code means that when the Red turtle passes the x coordinate at a position greater than or equal to 150 on the screen, it has won.

Next, we want a little celebration, so in the logic block, add code to change the variable name to Red and the position to move the turtle to just below the start line. To declare the winner, we will add a statement block and change the Hello World text to Red turtle wins!

We then need to add another loop under the statement, to make the turtle perform a little celebration. Add a break block to stop the code.

Block code



Python code

```
70  if Red.xcor() >=150:
71      Red.goto(-140, -60)
72      print("Red turtle wins!")
73      for turn in range(18):
74          Red.right(60)
75      break
```

EduBlocks Turtle Race

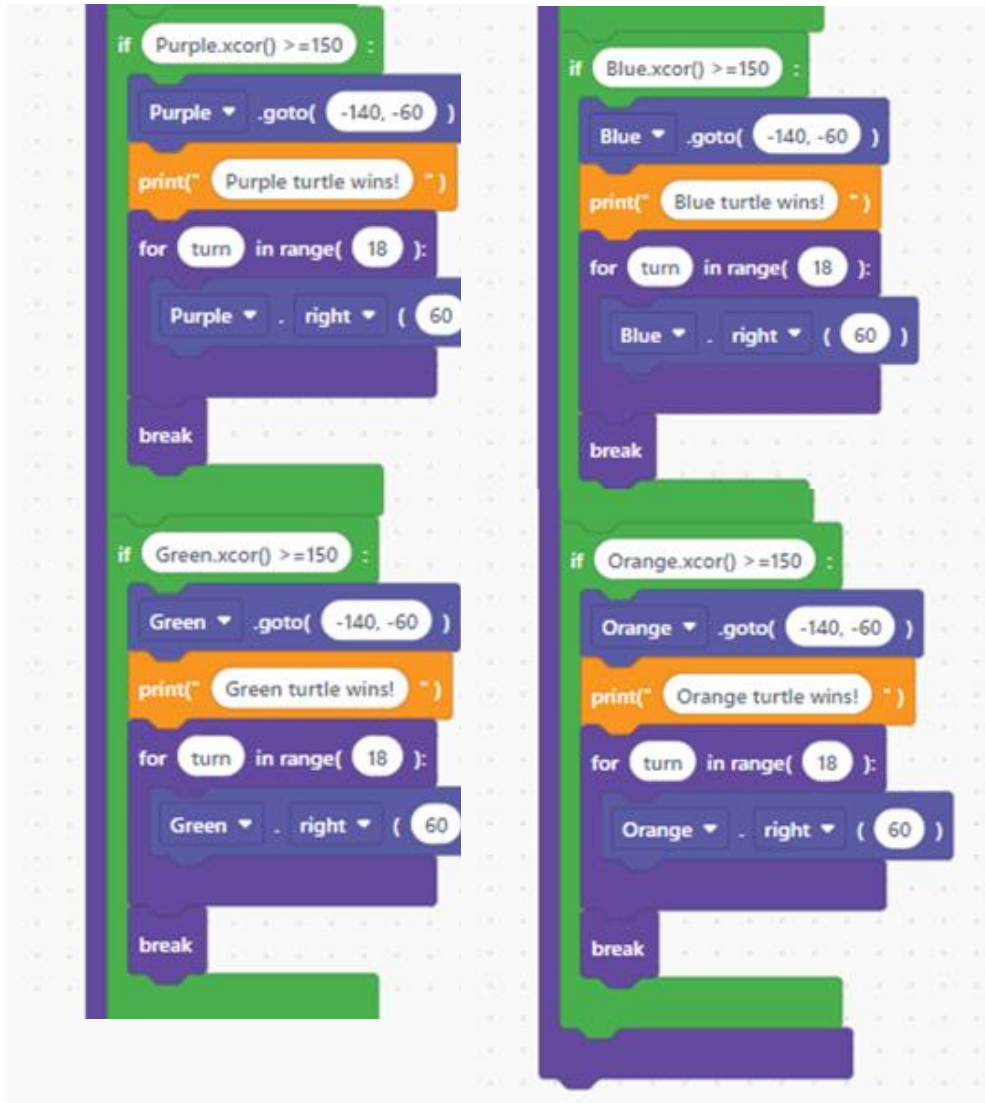
EduBlocks project

Step 7

Finally, we need to repeat the same code for the 4 remaining racing turtles.

Remember, this remaining code is all inside the loop we created in Step 6.

Block code



Python code

```

76 if Purple.xcor() >=150:
77     Purple.goto(-140, -60)
78     print("Purple turtle wins!")
79     for turn in range(18):
80         Purple.right(60)
81     break
82 if Green.xcor() >=150:
83     Green.goto(-140, -60)
84     print("Green turtle wins!")
85     for turn in range(18):
86         Green.right(60)
87     break
88 if Blue.xcor() >=150:
89     Blue.goto(-140, -60)
90     print("Blue turtle wins!")
91     for turn in range(18):
92         Blue.right(60)
93     break
94 if Orange.xcor() >=150:
95     Orange.goto(-140, -60)
96     print("Orange turtle wins!")
97     for turn in range(18):
98         Orange.right(60)
99     break

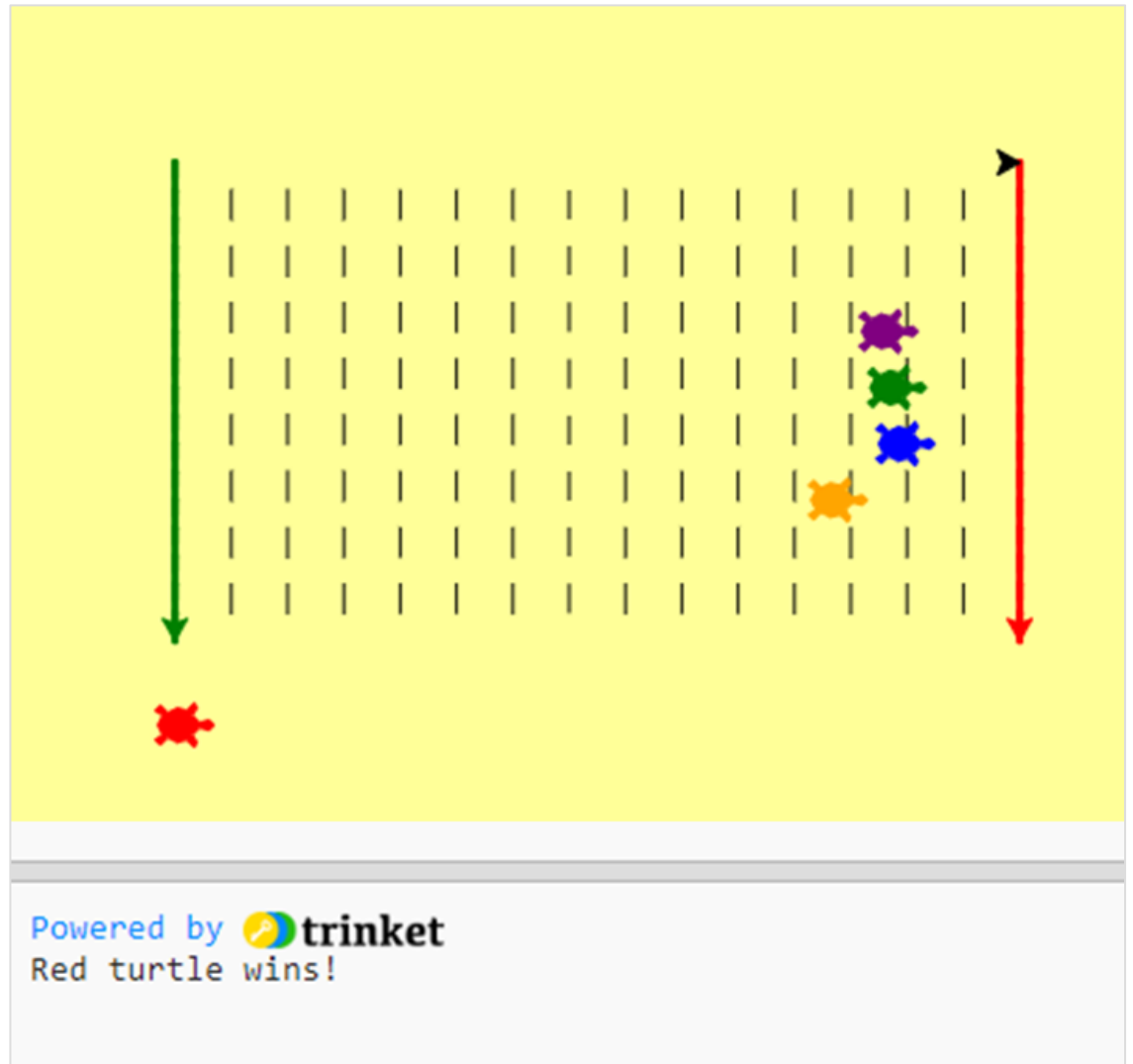
```

EduBlocks Turtle Race

EduBlocks project

Step 8

That's it! You have coded the complete Turtle Race! Now run your code to see which colour turtle wins.



Notes

Code Playground

Barclays Bank UK PLC is authorised by the Prudential Regulation Authority and regulated by the Financial Conduct Authority and the Prudential Regulation Authority (Financial Services Register No. 759676). Registered in England, Registered No. 9740322. Registered Office 1 Churchill Place, London E14 5HP.