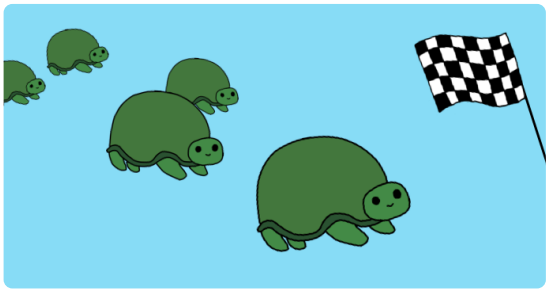
Projects

Turtle Race!

Race turtles against eachother!

Python

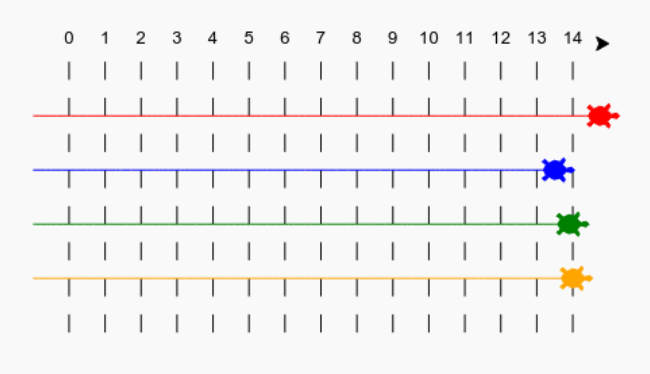


Step 1 Introduction

Use loops to draw a race track and create a racing turtle game.

What you will make

This project introduces for loops through a fun turtle race game. Loops are used to draw the race track and to make the turtles move a random number of steps each turn. If you have a group of people to play the game, each person pick a turtle and the one that gets the furthest is the winner.



What you will learn

By making your turtle race game, you will learn how to:

- Write `for` loops in Python
- Use random numbers in Python
- Draw lines in different colours with Python Turtle

This project covers elements from the following strands of the **Raspberry Pi Digital Making Curriculum** (http://rpf.io/curriculum):

- **Use basic programming constructs to create simple programs.**

(https://www.raspberrypi.org/curriculum/programming/creator/)

Completion of this project will earn you points towards your bronze "Digital Maker" iDEA badge. For more information, visit **idea.org.uk** (https://idea.org.uk).

Additional information for educators

If you need to print this project, please use the **printer-friendly version** (https://projects.raspberrypi.org/en/projects/turtle-race/print).

Use the link in the footer to access the GitHub repository for this project, which contains all resources (including an example finished project) in the 'en/resources' folder.

Additional information for club leaders

If you need to print this project, please use the **Printer friendly version** (https://projects.raspberry-pi.org/en/projects/turtle-race/print).

Step 2 What you will need

Hardware

- An internet-connected computer

Software

This project uses Python 3. We recommend using **Trinket** (https://trinket.io/), which allows you to write Python code online.

Step 3 Race track

You're going to create a game with racing turtles. First they'll need a race track.

- Open the blank Python template Trinket: **jumpto.cc/python-new** [(http://jumpto.cc/python-new)](http://jumpto.cc/python-new).
- Add the following code to draw a line using the 'turtle':

```
from turtle import *

forward(100)
```

- Now let's use the turtle to draw some track markings for the race.

  The turtle `write` function writes text to the screen.

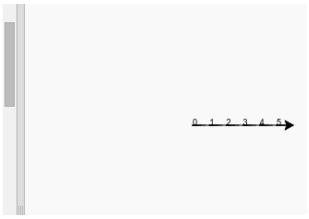
  Try it:

```
from turtle import *

write(0)
forward(100)
write(5)
```

- Now you need to fill in the numbers in between to create markings:

```
write(0)
forward(20)
write(1)
forward(20)
write(2)
forward(20)
write(3)
forward(20)
write(4)
forward(20)
write(5)
forward(20)
```



- Did you notice that your code is very repetitive? The only thing that changes is the number to write.

  There's a better way of doing this in Python. You can use a `for` loop.

  Update your code to use a `for` loop:

```
from turtle import *

for step in range(5):
  write(step)
  forward(20)
```

- Hmm, that only prints numbers up to 4. In Python `range(5)` returns five numbers, from 0 up to 4. To get it to also return 5 you'll need to use `range(6)`:

```
for step in range(6):
  write(step)
  forward(20)
```

- Now we can draw some track markings. The turtle starts at coordinates (0,0) in

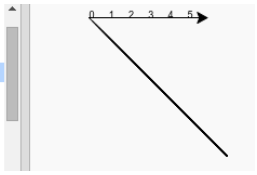
the middle of the screen.

Move the turtle to the top left instead:

```
from turtle import *

goto(-140, 140)

for step in range(6):
  write(step)
  forward(20)
```



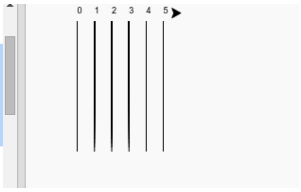
- Ah, you'll want to lift the pen up first!

```
penup()
goto(-140, 140)

for step in range(6):
  write(step)
  forward(20)
```

- Instead of drawing a line horizontally, let's draw vertical lines to create a track:

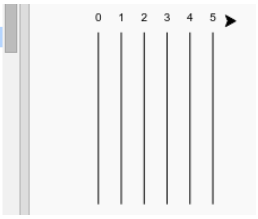
```
for step in range(6):
  write(step)
  right(90)
  forward(10)
  pendown()
  forward(150)
  penup()
  backward(160)
  left(90)
  forward(20)
```



`right(90)` makes the turtle turn right 90 degrees (a right angle.) Moving `forward(10)` before putting the pen down leaves a small gap between the number and the start of the line. After drawing the line you lift up the pen and go `backward(160)` the length of the line plus the gap.

- It looks neater if you centre the numbers:

```
for step in range(6):
  write(step, align='center')
  right(90)
  forward(10)
  pendown()
  forward(150)
  penup()
  backward(160)
  left(90)
  forward(20)
```



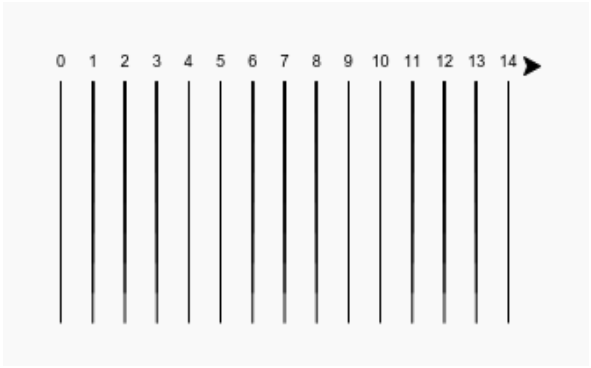
- And you can speed up the turtle so it draws faster:

```
from turtle import *

speed(10)
penup()
goto(-140, 140)
```

Step 4 Challenge: More lines

Can you change your code so that the track lines go right across the screen?

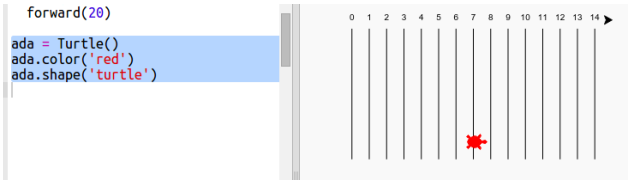


If you want to make the turtle go even faster you can use `speed(0)`.

Step 5 Racing turtles

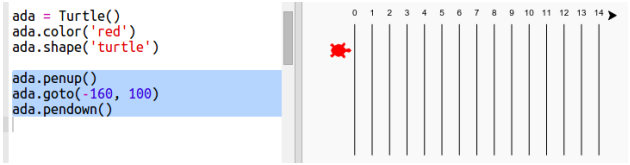
Now for the fun bit. Let's add some racing turtles. It would be really boring if the turtles did the same thing every time so they will move a random number of steps each turn. The winner is the turtle that gets the furthest in 100 turns.

- When you use commands like `forward(20)` you are using a single turtle. But you can create more turtles. Add the following code to the end of your script (but make sure it's not indented):



  The first line creates a turtle called 'ada'. The next lines set the colour and shape of the turtle. Now it really looks like a turtle!

- Let's send the turtle to the starting line:



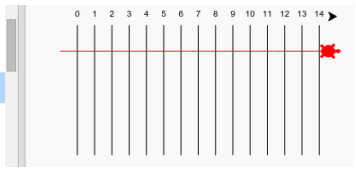
- Now you need to make the turtle race by moving a random number of steps at a time. You'll need the `randint` function from the Python `random` library. Add this `import` line to the top of your script:

```
from turtle import *
from random import randint
```

- The `randint` function returns a random integer (whole number) between the values chosen. The turtle will move forward 1, 2, 3, 4, or 5 steps at each turn.

```
ada.penup()
ada.goto(-160, 100)
ada.pendown()

for turn in range(100):
  ada.forward(randint(1,5))
```

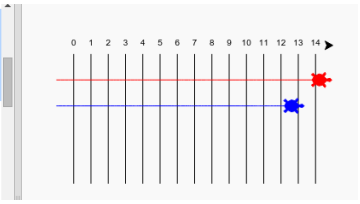


- One turtle isn't much of a race! Let's add another one:

```
bob = Turtle()
bob.color('blue')
bob.shape('turtle')

bob.penup()
bob.goto(-160, 70)
bob.pendown()

for turn in range(100):
  ada.forward(randint(1,5))
  bob.forward(randint(1,5))
```

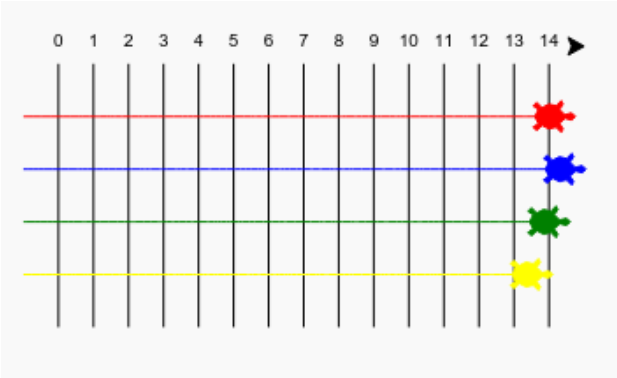


Note that the code for moving the blue turtle needs to be in **the same** `for` loop as the code for moving the red turtle so that they each make a move every turn.

Step 6 Challenge: Race time!

Now you're ready to race. Pick a turtle and an opponent and see who wins.

Can you add more turtles so you can race with more friends?



Colours include: orange, purple, violet, tomato, turquoise, magenta and brown - or you can go to **jumpto.cc/colours** [(http://jumpto.cc/colours)](http://jumpto.cc/colours) and pick any colour you like!

Step 7 Challenge: Do a twirl

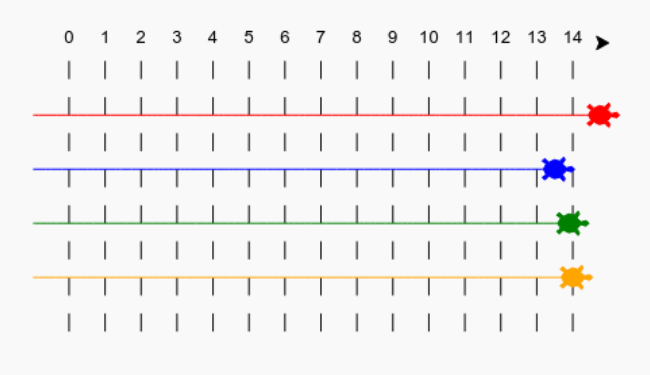
Can you use a `for turn in range():` loop to make each turtle do a 360 degree twirl after they get to the starting line? You'll need to make sure they are facing in the right direction at the start of the race!

`ada.right(36)` will turn the red turtle right by 36 degrees.

Hint: A full turn is 360 degrees. A turtle could turn right 10 degrees 36 times, or left 5 degrees 72 times, or any other numbers make 360!

Step 8 Challenge: Dashed lines

Can you use a loop to make the track lines dashed instead of solid?



Hint: Find the code that draws a straight line. Try using: `for`, `forward()`, `penup()` and `pendown()`