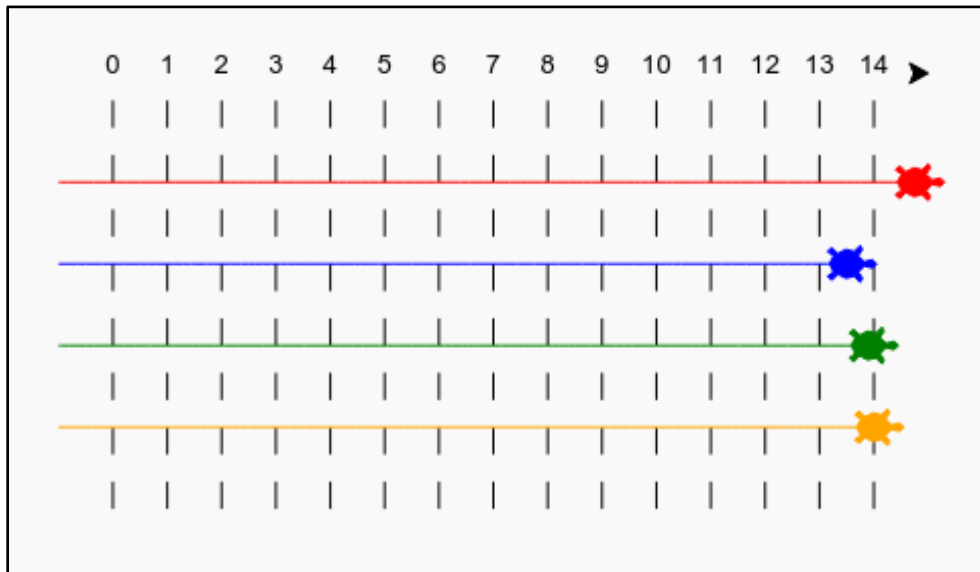


Turtle Race!

Introduction

In this project you will use loops to create a racing turtle game and draw a racetrack. You will be coding in **Python** and using the **Raspberry Pi**.



Step 1: Racetrack

You're going to create a game with racing turtles, but first they need a track to race on.

1. Launch **Thonny** (Simple Mode) on the Raspberry Pi
 - a. Click on the Raspberry icon in the top left corner
 - b. Go to Programming and select Thonny (Simple Mode) from the menu
2. Click on the **New** icon to create a new file
3. Using Python **commenting**, type your name, the course you're enrolled in and today's date e.g. Sheldon Cooper, ITCS1140 Winter 2019, 4/9/19
 - a. Single-line comments are created simply by beginning a line with the hash (#) character and are automatically terminated by the end of line.
 - b. For example: **# Sheldon Cooper, ITCS1140 Winter 2019, 4/9/19**
4. Click the **Save** icon to save your file to the **home/pi directory** and name it turtle_extracredit_**sc**.py. Please replace "sc" in the example with your own initials.

NOTE: The python file must reside within the **home/pi directory** on your Pi in order to have access to the imported libraries for this project. You can ask the lab aide for assistance if you are unsure where to locate the pi directory.

Below the comments you added within your Python file, **add the following code to your document 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.

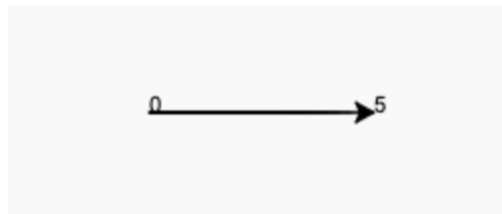
The turtle **write function** writes text to the screen.

Modify the code you just added above to include the highlighted write functions shown below:

```
from turtle import *  
write(0)  
forward(100)  
write(5)
```

Now it's time to test what we just wrote.

Click the **Run** button in Thonny located in the top menu bar to run and execute your code. When you click the Run button, your file is automatically **saved**. A new window will open that displays the output. The output should look like the image below.



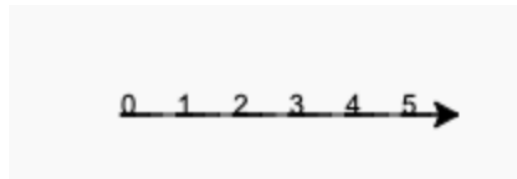
Return to Thonny and click the **Stop** button within the top menu bar.

Now you need to fill in the numbers in between to creating markings.

Modify the code you have written so far to match the code below. Please note that you will be removing `forward(100)` and replacing it with `forward(20)` as seen below:

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

Click the **Run** button in Thonny located in the top menu bar to run and execute your code. When you click the Run button, your file is automatically **saved**. A new window will open that displays the output. The output should look like the image below.



Return to Thonny and click the **Stop** button within the top menu bar.

Did you notice the 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**.

Modify your code to use a for loop by replacing what you previously have written with the highlighted for loop code below:

```
from turtle import *

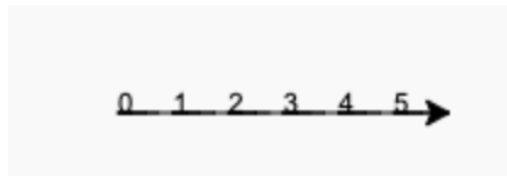
for step in range(5):
    write(step)
    forward(20)
```

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)`**.

Update your code to use `range(6)`:

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

Click the **Run** button in Thonny located in the top menu bar to run and execute your code. When you click the Run button, your file is automatically **saved**. A new window will open that displays the output. The output should look like the image below.



Return to Thonny and click the **Stop** button within the top menu bar.

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.

Update your code and add the following:

```
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! **Update your code and add the following:**

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

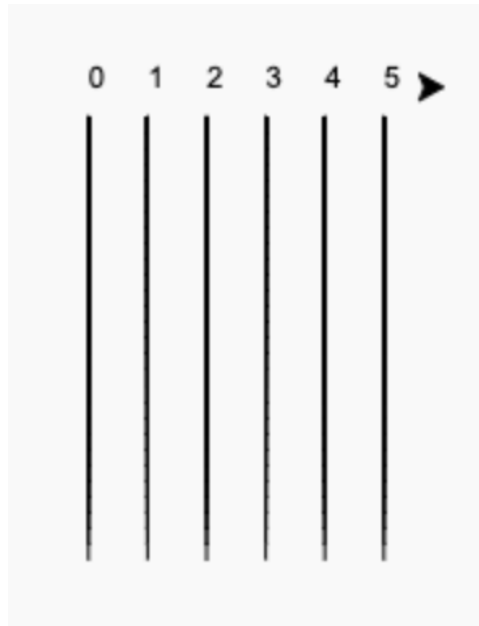
for step in range(6):
    write(step)
    forward(20)
```

Instead of drawing a line horizontally, let's draw vertical lines to create a track. **Update your code and add the following:**

```
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.

Click the **Run** button in Thonny located in the top menu bar to run and execute your code. When you click the Run button, your file is automatically **saved**. A new window will open that displays the output. The output should look like the image below.

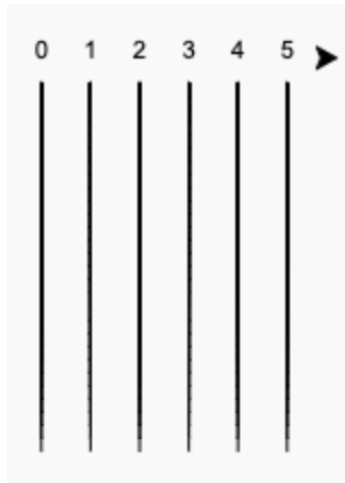


Return to Thonny and click the **Stop** button within the top menu bar.

It looks neater if you center the numbers. **Update your code to include the following within the write function:**

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

Click the **Run** button in Thonny located in the top menu bar to run and execute your code. When you click the Run button, your file is automatically **saved**. A new window will open that displays the output. The output should look like the image below.



Return to Thonny and click the **Stop** button within the top menu bar.

Want the turtle to go faster? Experiment by modifying the speed function and see what happens. **Update your code to include the following:**

```
from turtle import *
speed(10)
penup()
goto(-140, 140)
```

NOTE: If you want to make the turtle go even faster you can use **speed(0)**. Run your code to see the results. Is your turtle moving faster or slower?

Want to add more track lines? Adjust the number in the range function within the **for loop** you created. If I modified the for loop to this: **for step in range(15):** how many track lines would display on screen?

Step 2: Racing Turtles

Now for the fun part! 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 below the last line of your for loop (but make sure it's not indented) as shown below:

```
forward(20)
```

```
ada = Turtle()  
ada.color('red')  
ada.shape('turtle')
```

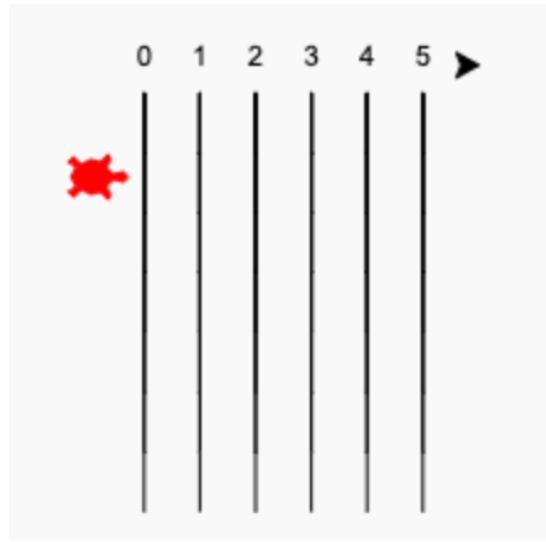
The first line of code above creates a turtle named 'ada'. The next lines set the color and shape of the turtle. Now it really looks like a turtle!

Let's send the turtle to the starting line. **Add the following highlighted code:**

```
ada = Turtle()  
ada.color('red')  
ada.shape('turtle')
```

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


Click the **Run** button in Thonny located in the top menu bar to run and execute your code. When you click the Run button, your file is automatically **saved**. A new window will open that displays the output. The output should look like the image below.



Return to Thonny and click the **Stop** button within the top menu bar.

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 the highlighted import line to the top of your script:

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

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

Add the following highlighted code to the bottom your script:

```
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!

Add the following highlighted code to your script below the code for the turtle named 'ada' to create another turtle 'bob':

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

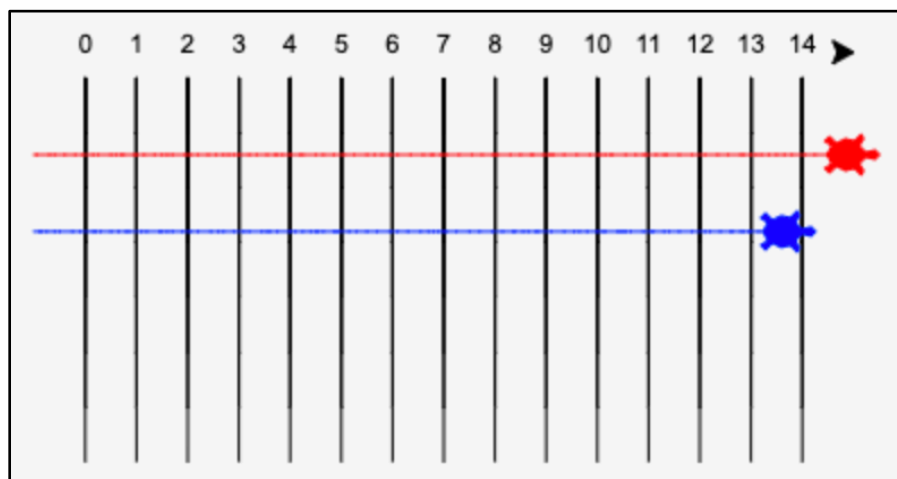
bob.penup()
bob.goto(-160, 70)
bob.pendown()

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

NOTE: Don't forget to update the code with the **for loop** to include 'bob'!

NOTE: 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. To show more racetrack lines, modify the **number** within the **range function** within the **for loop** you created on Page 4 in this document. **HINT: range(15)**

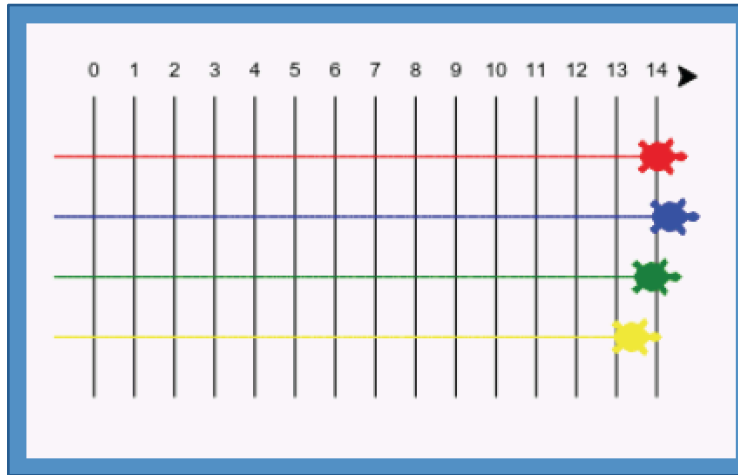
Click the **Run** button in Thonny located in the top menu bar to run and execute your code. When you click the Run button, your file is automatically **saved**. A new window will open that displays the output. The output should look like the image below.



Return to Thonny and click the **Stop** button within the top menu bar.

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?



Colors include: orange, purple, violet, tomato, turquoise, magenta and brown or you can go to <https://trinket.io/docs/colors> and pick any color you like.

Deliverable

1. Download the **Extra Credit worksheet** from the Makerspace website.
2. Print out the worksheet and fill in the appropriate information.
3. Sign the form.
4. Have the lab aid on duty sign the form.
5. Turn it in to your instructor.

Happy Coding!