Even More turtle Methods

Here are a few more things that you might find useful as you write programs that use turtles.

- You can speed up or slow down the turtle's animation speed. (Animation controls how quickly the turtle turns and moves forward). Speed settings can be set between 1 (slowest) to 10 (fastest). But if you set the speed to 0, it has a special meaning — turn off animation and go as fast as possible.

```
sarah.speed(10)
```

- A turtle can "stamp" its footprint onto the canvas, and this will remain after the turtle has moved somewhere else. Stamping works even when the pen is up.

Let's do an example that shows off the stamping feature.

```
from turtle import *
wn = Screen()
wn.bgcolor("lightgreen")
tess = Turtle()
tess.color("blue")
tess.shape("turtle")

print(range(5,60,2))
tess.up()                       # this is another way to pick up the pen
for dist in range(5,60,2):      # start with dist = 5 and grow by 2
    tess.stamp()                # leave an impression on the canvas
    tess.forward(dist)          # move tess along
    tess.right(24)              # and turn her

wn.exitonclick()

```

ActiveCode: 1 (a_9)

Run

Save Load

One more thing to be careful about. All except one of the shapes you see on the screen here are footprints created by `stamp` . But the program still only has *one* turtle object — can you figure out which one is the real tess? (Hint: if you're not sure, write a new line of code after the `for` loop to change tess' color, or to put her pen down and draw a line, or to change her shape, etc.)

One new thing that we used in the program above was the **range** function. The range function returns a list of integers. You can specify the starting integer, the ending integer, and the amount to change by. Notice that the program above also printed the list returned from the range function using `print(range(5,60,2))` . Does the list include the starting integer? Does the list include the ending integer? You can just give one value to the range function. What do you think that would return?

Here is a program that uses the range function and has two loops with one inside of the other. The inner loop draws a square. What do you think the outer loop does? Why do you think the turtle turns 36 degrees? Try changing the outer range and amount to turn left to create a cool drawing.

```
from turtle import *
wn = Screen()
sarah = Turtle()
sarah.color("orange")

for outer in range(10):     #repeat 10 times
    for inner in range(4):  #repeat four times
        sarah.forward(50)
        sarah.left(90)
    sarah.left(36)          # turn left 36 degrees

wn.exitonclick()

```

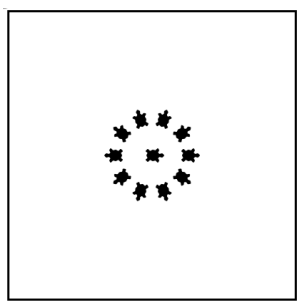
ActiveCode: 2 (a_10)

Run

Save Load

Mixed up program

trl-1: The following program uses the stamp method to create a circle of turtle shapes as shown to the left, but the lines are mixed up. The program should do all necessary set-up, create the turtle, set the shape to "turtle", and pick up the pen. Then the turtle should repeat the following ten times: go forward 50 pixels, leave a copy of the turtle at the current position, reverse for 50 pixels, and then turn right 36 degrees. After the loop, set the window to close when the user clicks in it.



Drag the blocks of statements from the left column to the right column and put them in the right order with the correct indention. Click on *Check Me* to see if you are right. You will be told if any of the lines are in the wrong order or are incorrectly indented.

Drag from here

Drop blocks here

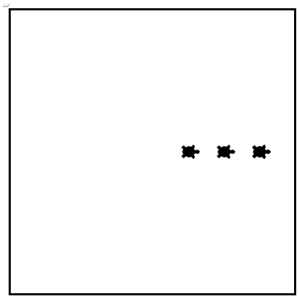
```
from turtle import *
wn = Screen()
jose = Turtle()
jose.shape("turtle")
jose.penup()
for size in range(10):
    jose.forward(50)
    jose.stamp()
    jose.forward(-50)
    jose.right(36)
wn.exitonclick()
```

Check Me | Reset

Perfect!

Mixed up program

trl-2: The following program uses the stamp method to create a line of turtle shapes as shown to the left, but the lines are mixed up. The program should do all necessary set-up, create the turtle, set the shape to "turtle", and pick up the pen. Then the turtle should repeat the following three times: go forward 50 pixels and leave a copy of the turtle at the current position. After the loop, set the window to close when the user clicks in it.



Drag the blocks of statements from the left column to the right column and put them in the right order with the correct indention. Click on *Check Me* to see if you are right. You will be told if any of the lines are in the wrong order or are incorrectly indented.

Drag from here

Drop blocks here

```
from turtle import *
wn = Screen()
```

```
nikea = Turtle()
nikea.shape("turtle")
nikea.penup()
for size in range(3):
    nikea.forward(50)
    nikea.stamp()
wn.exitonclick()
```

Check Me Reset

Perfect!

If you enjoyed this quick introduction to Python you might want to learn more by going to http://runestoneinteractive.org (http://runestoneinteractive.org). This short introduction started as one of the chapters in the book “How to Think Like a Computer Scientist”.