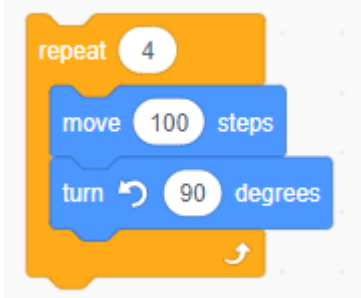



Lesson 6: *for* loop

In programming, a **loop** means repeating something multiple times. One of the main Python primitive loop commands is *for* loop:

Summary:

Code Instruction	What it does
<pre data-bbox="204 730 512 824">for m in range (4): t.forward(100) t.left (90)</pre> <p data-bbox="229 887 571 1149">Keep in mind that command t.fd(100) is equivalent command t.fd(100), t.forward is the same as t.fd(100)</p> <p data-bbox="204 1167 571 1301">t.left(90) is the same as t.lt(90) and t.right(90) is the same as t.rt(90)</p>	<p data-bbox="608 730 1378 864">Here <i>for</i> loop is used for a block of two lines code [t.forward(100) and t.left(90)] which you want to repeat 4 times</p> <p data-bbox="632 875 1370 920">This short program is equivalent to the following</p> 
<pre data-bbox="204 1317 571 1451">for m in range (2): for n in range (4): t.fd(100) t.left (90)</pre>	<p data-bbox="608 1317 1353 1536">Nested Loop, equivalent to the following block-based code. Nested loop is loop inside a loop, the outer loop only repeats (here 2 times) after inner loop has gone round its required number of times (here 4)</p> 

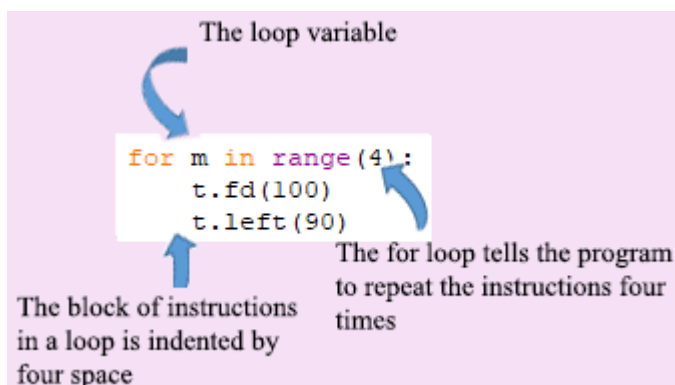
Let's learn how to write more efficient Python code by using loop structure
Let's simplify code that draws simple square. Code looks as follows:

```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.color('red')
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
```



What we can see from the structure of this code. The program above gives the turtle the same commands **t.forward(100)** and **t.left(90)** four times – once for each side of the square. An alternative to this is to use these two commands inside a *for* loop. Whenever we need to do something over and over again in a program, loops allow us to repeat those steps without having to type each one separately.

To build our own loop, we first need to identify the repeated steps. The instructions that we're repeating in the preceding code are `t.forward(100)` to draw a turtle line with a length of 100 pixels and `t.left(90)` to turn the turtle left 90 degrees before drawing the next circle. Second, we need to figure out how many times to repeat those steps. We want four circles to build a square, so let's start with four.



We start with the keyword *for* and then we give an integer number *m* that will be our counter. Here variable *m*(integer number) is something you want the computer to remember while your program is running. We use a name of this variable as an *m*, however it could be any name as you want, for example, *a*, *b*, *x*, *y* ... Name of the variable does not matter, the key point that this variable *m* successively takes the following values: first step value *m* equal 0, second step

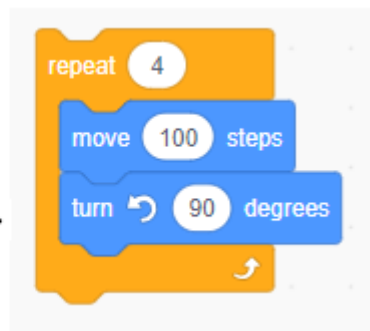
$m=1$, third step $m=2$, and fourth step $m=3$. For each step computer draws line `t.forward(100)` and turn turtle by 90 degrees `t.left(90)`. So, code with a loop is as following

```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.color('red')
for m in range (4):
    t.forward(100)
    t.left(90)
```



This *for* loop is equivalent to the block-based code shown below

```
for m in range (4):
    t.forward(100)
    t.left(90)
```



Keep in mind that there are four additional spaces at the beginning lines of the loop block as shown below (when you compare with the line `for m in range (4):`). Code that is not indented, which comes after the loop, will be run once just like regular code (Example #4).

```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.color('red')
for m in range (4):
    |   t.forward(100)
    |   t.left(90)
|
4 spaces
```

Python editor will have automatically indented this for you.

Loop variable (in our code loop variable named as m counts the number of times a loop has repeated itself. It starts at the first value in the range (0) and stops one before the last value (in our loop \rightarrow 3).

Nested Loops

Can the body of a loop have another loop within it? Yes!! This is called a nested loop. It's like Russian dolls, where each doll fits inside a larger doll. In a nested loop, an inner loop runs inside an outer loop.



Let's give an example. We will draw several squares, each of which is rotated 45 degrees from the previous one.

```
import turtle
t = turtle.Turtle()
wn=turtle.Screen()
wn.bgcolor('lightgrey')
t.hideturtle()
t.color('blue')
t.pensize(5)

for m in range(2):
    for n in range(4):
        t.fd(100)
        t.left(90)
        t.left(45)
```

In this example the inner loop (variable n goes around 4 times (we draw square). Code `t.left(45)` is a body of outer loop (variable m) and turns turtle by 45 degrees, and then we again go to the inner loop. Let's detail an operation of the code.

Step #1: The first time through the outer loop, counter m has a starting value of 0, turtle goes to the inner loop, variable n takes value equal 0, turtle draws first line 100 px, then turns left by 90 degrees.

Step #2 Python goes back to the beginning of the inner loop and sets n to 1. Then it draws second line and turns left by 90 degrees.

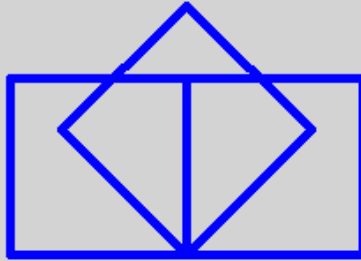
Step#3 Python goes back through the loop again, increasing n to 2. It draws third line and turn by 90 degrees.

Step #4 Python goes back through the loop again, increasing n to 3. It draws third line and turn by 90 degrees.

Step#5 Python goes out of inner loop, turns turtle to 45 degrees, go back to the beginning of the outer loop and sets outer variable m to 1. Then Python enters to the inner loop and repeats all steps #1 to #4. Because, outer variable changes in the range from 0 to 1 Python draws only 2 squares. Let's see what will happen, if outer loop looks like: `for m in range(3):` or `for m in range(4):`

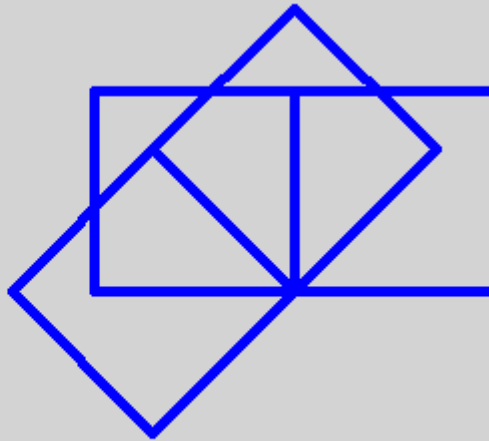
```
import turtle
t = turtle.Turtle()
wn=turtle.Screen()
wn.bgcolor('lightgrey')
t.hideturtle()
t.color('blue')
t.pensize(5)

for m in range(3):
    for n in range (4):
        t.fd(100)
        t.left(90)
        t.left(45)
```



```
import turtle
t = turtle.Turtle()
wn=turtle.Screen()
wn.bgcolor('lightgrey')
t.hideturtle()
t.color('blue')
t.pensize(5)

for m in range(4):
    for n in range (4):
        t.fd(100)
        t.left(90)
        t.left(45)
```





Python + Math

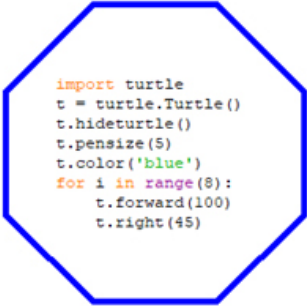
Code

Output

for Kids



```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.pensize(5)
t.color('blue')
for i in range(8):
    t.forward(100)
    t.right(45)
```



```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.pensize(5)
t.color('red')
t.begin_fill()
for i in range(5):
    t.forward(150)
    t.right(144)
t.end_fill()
```

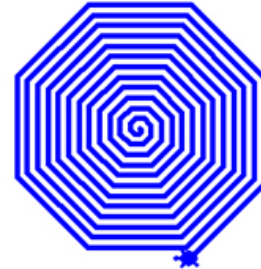


```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.pensize(5)
t.color('red')
for i in range(10):
    t.circle(50)
    t.left(36)
```



```
import turtle
import time
t = turtle.Turtle('turtle')
t.color('blue')
t.pensize(5)

for i in range(100):
    t.fd(1)
    t.right(45)
```



```
import turtle
import time
t = turtle.Turtle('turtle')
t.color('blue')
t.pensize(5)

for j in range(10):
    for i in range(20):
        t.down()
        t.fd(10)
        t.up()
        t.fd(10)
        t.down()
    t.up()
    t.left(90)
    t.fd(20)
    t.left(90)
    t.fd(400)
    t.left(180)
```



```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.pensize(5)
t.color('gold')
t.begin_fill()
for i in range(4):
    t.circle(50)
    t.left(90)
t.end_fill()
```



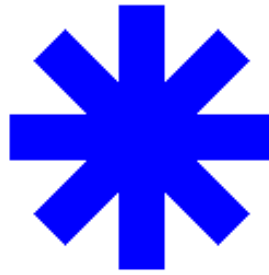
LESSON 6

To see examples, images, and challenges
www.python.kidsgo.ca



1. **Example #1** (Draw snowflake without and with loop)
 - a. (No loop)

```
import turtle
t=turtle.Turtle('square')
t.color('blue')
t.shapesize(6,1)
t.stamp()
t.right(45)
t.stamp()
t.right(45)
t.stamp()
t.right(45)
```

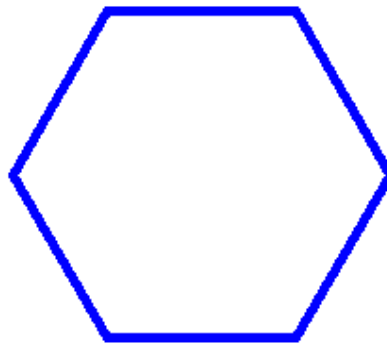


b. (With loop)

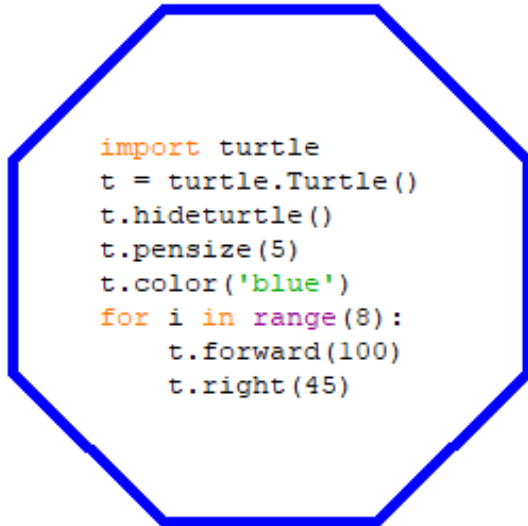
```
import turtle
t = turtle.Turtle('square')
t.color('blue')
t.shapesize(6,1)
for i in range(3):
    t.stamp()
    t.right(45)
```

2. Example #2 (Hexagon)

```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.pensize(5)
t.color('blue')
for i in range(6):
    t.forward(100)
    t.right(60)
```



3. **Example #3** (Octagon)



4. **Example #4** (Star)

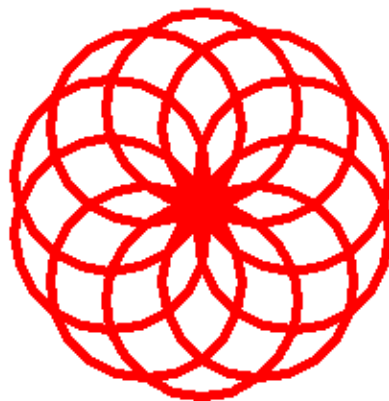
```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.pensize(5)
t.color('red')
t.begin_fill()
for i in range(5):
    t.forward(150)
    t.right(144)
t.end_fill()
```



5. **Example #5** (Number of circles)

a.

```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.pensize(5)
t.color('red')
for i in range(10):
    t.circle(50)
    t.left(36)
```



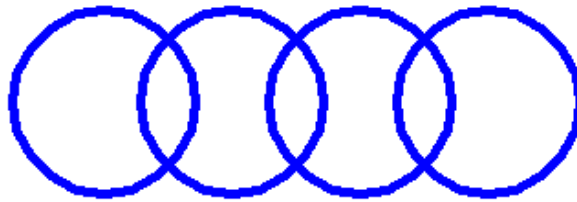
b.

```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.pensize(5)
t.color('gold')
t.begin_fill()
for i in range(4):
    t.circle(50)
    t.left(90)
t.end_fill()
```

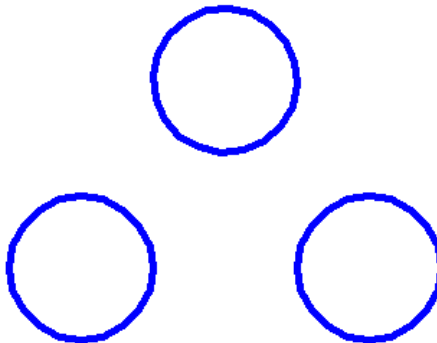


c.

```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.pensize(5)
t.color('blue')
for i in range(4):
    t.circle(50)
    t.up()
    t.fd(70)
    t.down()
```



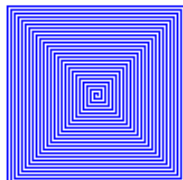
```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.pensize(5)
t.color('blue')
for i in range(1):
    t.circle(50)
    t.up()
    t.fd(200)
    t.down()
    t.circle(50)
    t.up()
    t.setheading(105)
    t.fd(200)
    t.down()
    t.circle(50)
```



6. Example #6 (Square Spiral)

```
import turtle
t = turtle.Turtle()
t.hideturtle()

t.color('blue')
for i in range(100):
    t.forward(i)
    t.left(90)
```

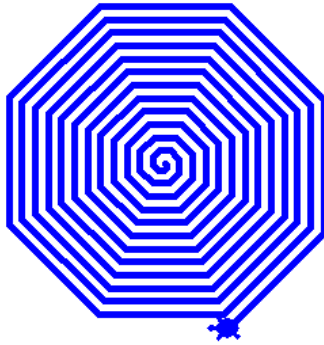


This particular loop sets up a range of 100 numbers from 0 to 99. (0,1,2,3,4... 99 totally 100 values). Because each time through the loop, our counter i changes value by 1, a length of the line also increases by 1

7. Example #8 (Spiral)

```
import turtle
import time
t = turtle.Turtle('turtle')
t.color('blue')
t.pensize(5)

for i in range (100):
    t.fd(i)
    t.right(45)
```



8. Example #7 (Christmas tree)

```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.color('lightgreen')
t.pensize(5)

t.begin_fill()
for i in range(3):
    t.fd(150)
    t.left(120)
t.end_fill()

t.up()
t.goto(0,70)
t.down()
t.begin_fill()
for i in range(3):
    t.fd(150)
    t.left(120)
t.end_fill()

t.up()
t.goto(0,140)
t.down()

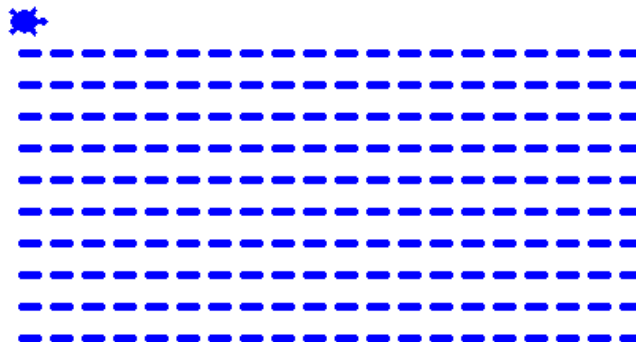
t.begin_fill()
for i in range(3):
    t.fd(150)
    t.left(120)
t.end_fill()
```



9. Example #8 (Dashed lines with nested loop)

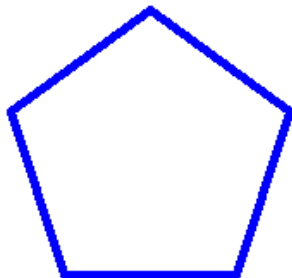
```
import turtle
import time
t = turtle.Turtle('turtle')
t.color('blue')
t.pensize(5)

for j in range (10):
    for i in range(20):
        t.down()
        t.fd(10)
        t.up()
        t.fd(10)
        t.down()
    t.up()
    t.left(90)
    t.fd(20)
    t.left(90)
    t.fd(400)
    t.left(180)
```

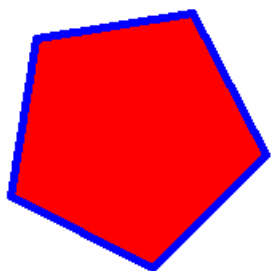


Challenges: write codes to create the following geometry shapes with circle code:

1. Expected output



2. Expected output



3. Expected output

