

Foundations of Programming

The Turtle

Importing Libraries

```
import math
```

```
a = math.sqrt(2)
```

```
from math import sqrt
```

```
a = sqrt(2)
```

```
from math import *
```

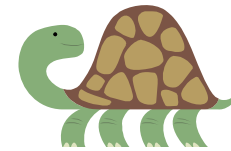
```
a = sqrt(2)
```

Python's Turtle Package

```
import turtle  
bob = turtle.Turtle()  
bob.forward(100)
```

Object manipulation

bob



bob



A turtle is an **Object**

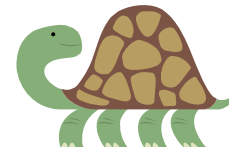
Python's Turtle Package

```
import turtle

bob = turtle.Turtle()

bob.forward(100)
bob.penup()
bob.right(90)
bob.backward(50)
bob.pendown()
bob.left(45)
bob.forward(100)
bob.setpos(-50,50)
print(bob.position())
```

bob



bob

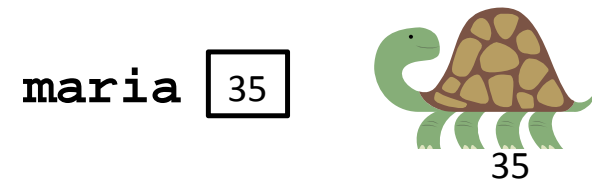
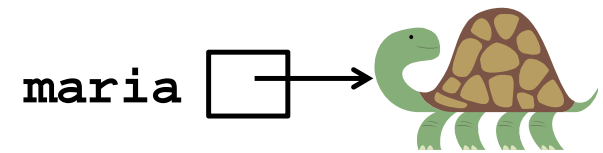


What happens if I replace the last line with the following?
`print(turtle.position)`

References

```
import turtle  
maria = turtle.Turtle()  
  
maria.forward(100)
```

The arrows are just a graphical representation of the reference (i.e., location of) the object in memory.

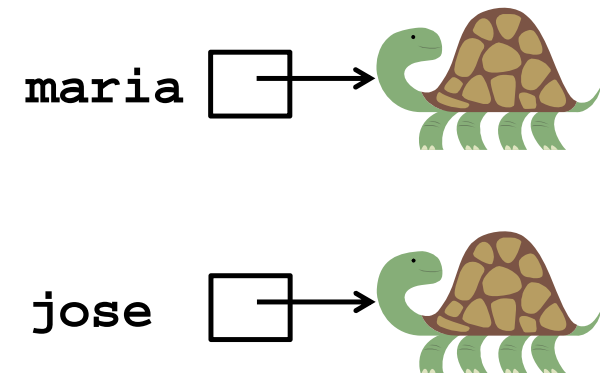


References

```
import turtle

maria = turtle.Turtle()
jose  = turtle.Turtle()

maria.forward(100)
jose.left(90)
jose.forward(100)
```



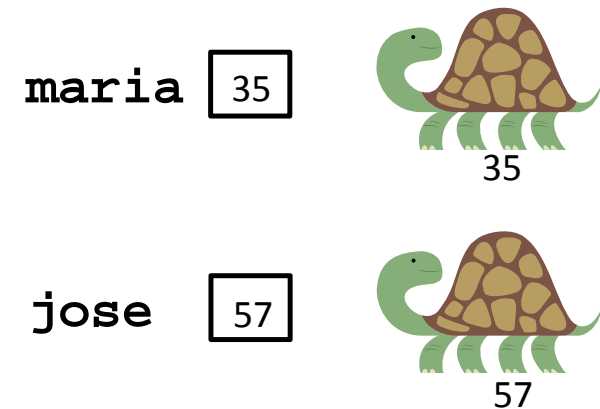
Two different turtle "objects" with the same capabilities
The variables store "references" (location of the Turtle in memory) to each Turtle.

References

```
import turtle

maria = turtle.Turtle()
jose  = turtle.Turtle()

maria.forward(100)
jose.left(90)
jose.forward(100)
```



Two different turtle "objects" with the same capabilities
The variables store "references" (location of the Turtle in memory) to each Turtle.

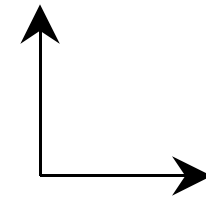
References

```
import turtle

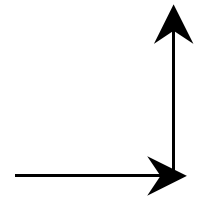
maria = turtle.Turtle()
jose = turtle.Turtle()
jose = maria

maria.forward(100)
jose.left(90)
jose.forward(100)
```

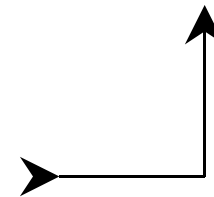
What do you expect the end state to be?



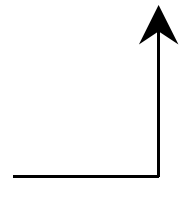
[A]



[B]



[C]



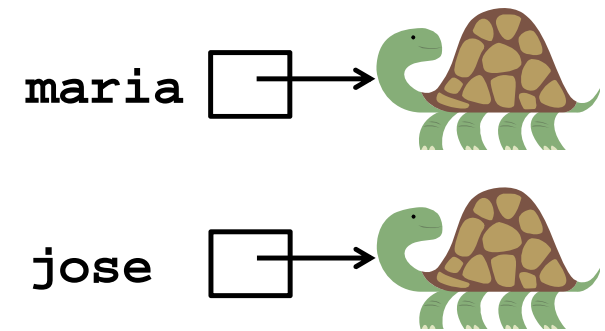
[D]

[E] Something else

References

```
import turtle  
  
maria = turtle.Turtle()  
jose  = turtle.Turtle()
```

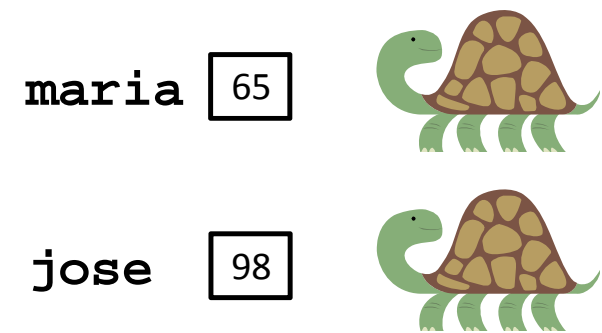
```
jose = maria
```



References

```
import turtle  
  
maria = turtle.Turtle()  
jose  = turtle.Turtle()
```

```
jose = maria
```

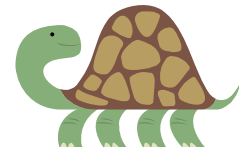


References

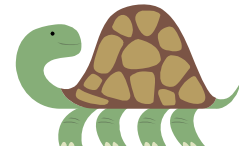
```
import turtle  
  
maria = turtle.Turtle()  
jose  = turtle.Turtle()
```

```
jose = maria
```

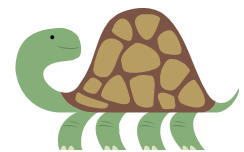
maria [65]



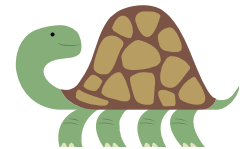
jose [98]



maria [65]



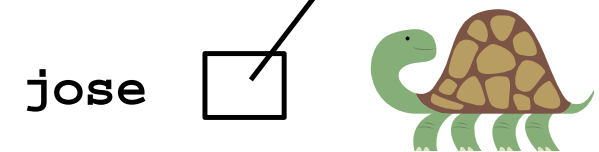
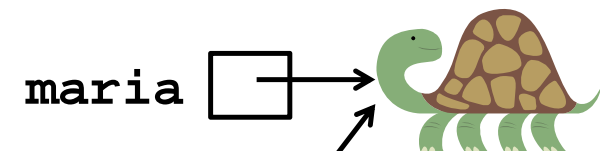
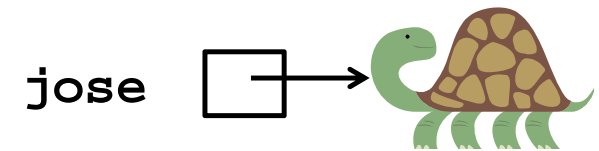
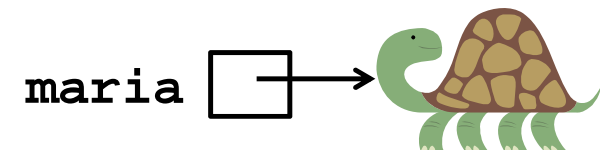
jose [65]



References

```
import turtle  
  
maria = turtle.Turtle()  
jose  = turtle.Turtle()
```

```
jose = maria
```

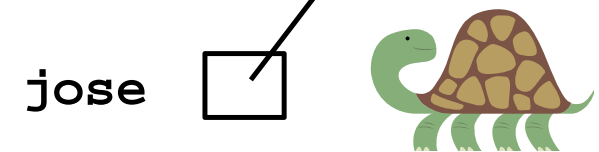
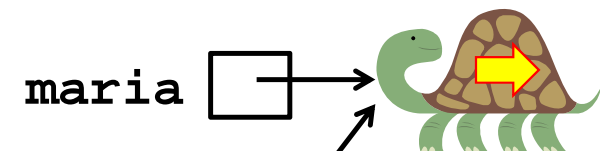
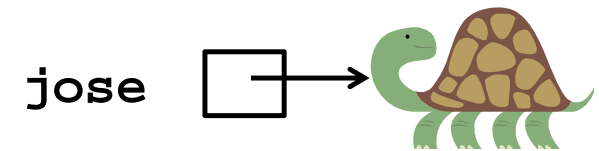
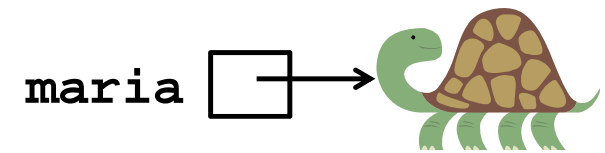


References

```
import turtle  
  
maria = turtle.Turtle()  
jose = turtle.Turtle()
```

```
jose = maria
```

```
maria.forward(100)
```



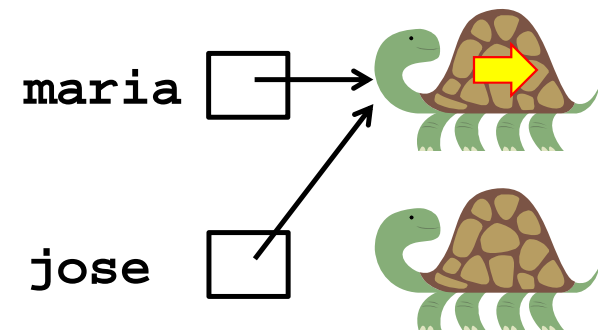
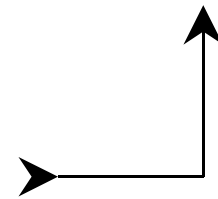
References

```
import turtle

maria = turtle.Turtle()
jose = turtle.Turtle()

jose = maria

maria.forward(100)
jose.left(90)
jose.forward(100)
```



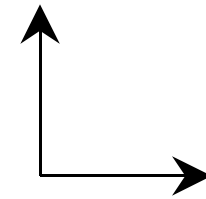
References

```
import turtle

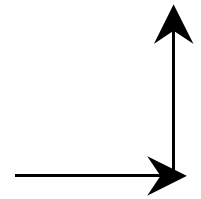
maria = turtle.Turtle()
#jose = turtle.Turtle()
jose = maria

maria.forward(100)
jose.left(90)
jose.forward(100)
```

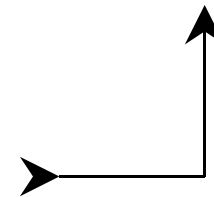
What do you expect the end state to be?



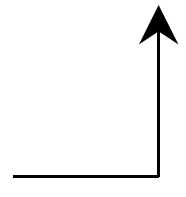
[A]



[B]



[C]



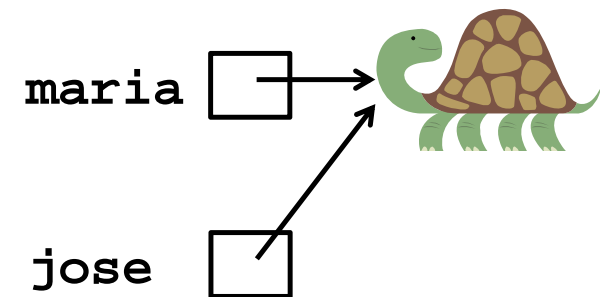
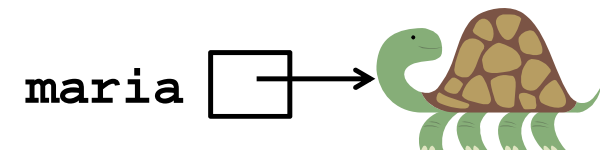
[D]

[E] Something else

References

```
import turtle  
  
maria = turtle.Turtle()  
#jose = turtle.Turtle()
```

```
jose = maria
```



What does this code draw?

```
import turtle
shaeli = turtle.Turtle()
chase = turtle.Turtle()
jaz = shaeli
shaeli.penup()
chase.setpos(0,-200)
jaz.forward(100)
jaz = chase
chase = shaeli
chase.pendown()
shaeli.backward(100)
jaz.forward(100)
```

Review

```
def turn(x):  
    x += 1  
  
a = 0  
turn(a)  
print(a)
```

What gets printed?

[A] 1 [B] 0

[E] Something else

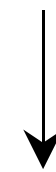
What does this code draw?

```
import turtle

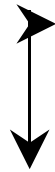
def turn(someTurtle):
    someTurtle.right(90)

gaby = turtle.Turtle()
turn(gaby)
gaby.forward(100)
```

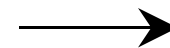
What do you expect the end state to be?



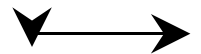
[A]



[B]



[C]



[D]

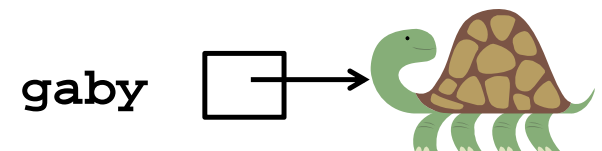
[E] Something else

What does this code draw?

```
import turtle

def turn(someTurtle):
    someTurtle.right(90)

gaby = turtle.Turtle()
turn(gaby)
gaby.forward(100)
```



What does this code draw?

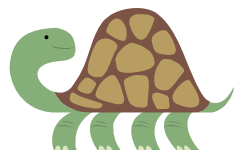
```
import turtle

def turn(someTurtle):
    someTurtle.right(90)

gaby = turtle.Turtle()
turn(gaby)
gaby.forward(100)
```

gaby

41



What does this code draw?

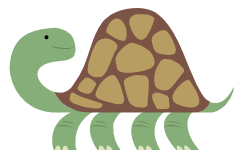
```
import turtle

def turn(someTurtle):
    someTurtle.right(90)

gaby = turtle.Turtle()
turn(gaby)
gaby.forward(100)
```

someTurtle 41

gaby 41

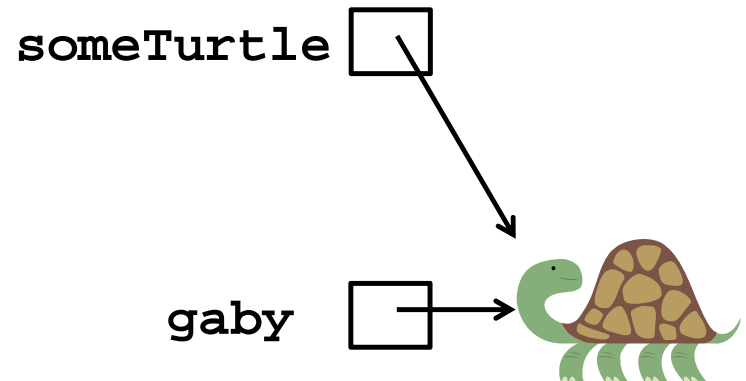


What does this code draw?

```
import turtle

def turn(someTurtle):
    someTurtle.right(90)

gaby = turtle.Turtle()
turn(gaby)
gaby.forward(100)
```



Turtles and Functions

```
import turtle

def drawShape(someTurtle):
    someTurtle.right(90)
    someTurtle.forward(100)
    someTurtle.left(30)
    someTurtle.forward(50)

gaby = turtle.Turtle()
drawShape(gaby)
gaby.forward(100)
drawShape(gaby)
```


Review - Loops

```
for x in [8, 1, 6, 3, 4, 2, 6]:  
    print(x)
```

```
sides = [ (0,4), (7,6), (8,3) ]  
for x in sides:  
    if (x[0]-x[1] < 0):  
        print(x)
```

Turtles and Functions

```
import turtle

# Draw a figure visiting the
# coordinates in coord
def drawFig(tt, coord):
    # Complete this function

coord = [(100,200), (300,0), (0,-100)]
nandini = turtle.Turtle()
drawFig(nandini, coord)
```

Complete the code.

Commands

```
forward(x)
backward(x)
right(x)
left(x)
setpos(x,y)
penup()
pendown()
...
```

Turtles and Functions

```
import turtle

# Draw a figure visiting the
# coordinates in coord
def drawFig(tt, coord):
    tt.penup()
    for x in coord:
        tt.setpos(x[0],x[1])
        tt.pendown()
    tt.setpos(coord[0][0],coord[0][1])

coord = [(100,200),(300,0),(0,-100)]
nandini = turtle.Turtle()
drawFig(nandini,coord)
```

Reading from a csv file

```
import csv
...
hurricaneFile = "data/irma.csv"

# The line below is a little magical. It opens the file,
# with awareness of any errors that might occur.
with open(hurricaneFile, 'r') as csvfile:

    # This line gives you an "iterator" you can use to get each line
    # in the file.
    pointreader = csv.reader(csvfile)

    for row in pointreader:
        # This code just prints out each row in the file
        # You'll need to change it
        for data in row:
            print(data, ' ', end='')
        print()
```