Code For China
Lesson 5

Review of Lesson 4

- If / Elif / Else Statements
- And Statements
- Mouse Events
- Writing Text on Screen
- Turtle Coordinates
- Turtle Shapes

REVIEW

if / elif / else statements

```
def doMath(firstNum, secondNum):
    if firstNum - secondNum >= 0:
        if firstNum - secondNum == 0:
            print(“they are equal”)
        else:
            print("firstNum is greater than secondNum")
    else:
        print("firstNum is smaller than secondNum")
```

Notice the number of tabs

REVIEW

and statements

```
def doMath(firstNum, secondNum):
    if firstNum > 0 and secondNum > 0:
        print("Both numbers are positive")
    elif firstNum > 0 and secondNum < 0:
        print("firstNum is positive only")
    elif firstNum < 0 and secondNum > 0:
        print("secondNum is positive only")
```

and requires both sides of the expression to be **True**

REVIEW

Mouse Click Event

https://repl.it/@jessicae5/mouseClick

```
def mouseClicked(x, y):
    ### CODE HERE ###

screen.onclick(mouseClicked)
screen.listen()
```

REVIEW

Writing on the Screen

```
turtle.write("Score: " + str(score))
```

```
turtle.clear()
```

REVIEW

Checking Turtle Position

```
x = turtle.xcor()
y = turtle.ycor()
```

REVIEW

Turtle Shapes

```
turtle.shape("turtle")
```

Other shapes: “arrow”, “turtle”, “circle”, “square”, “triangle”, “classic”

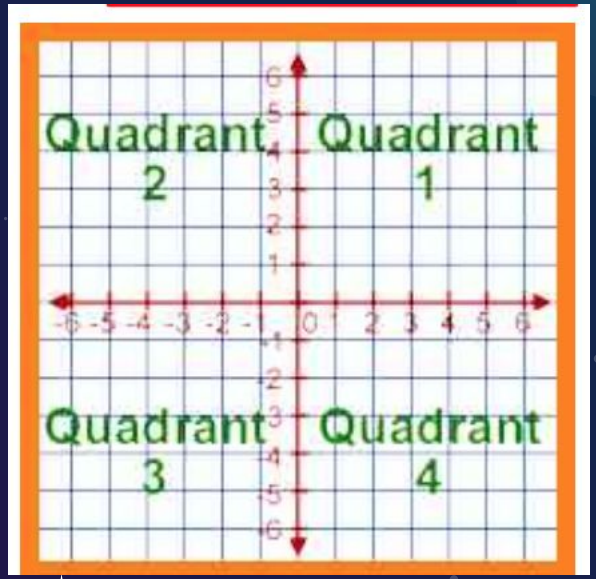
CODING TIME!

Functions Warmup

- Write function **whichQuadrant(x,y)** that decides which quadrant a click was in: 1,2,3 or 4
- the function returns 1,2,3 or 4
- from the **clickedFunction(x,y)** call **whichQuadrant(x,y),** save the return value
- print the value to the console

Remember to TEST your function!

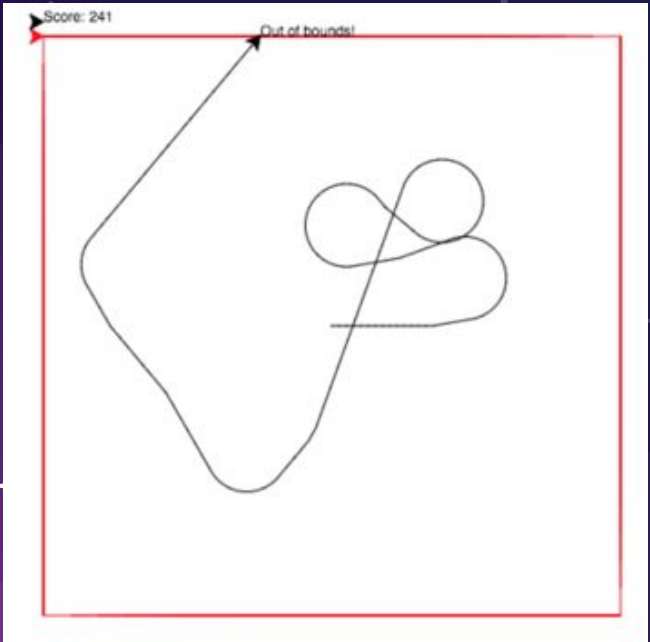
Starter code: shorturl.at/bwJKQ



CODING TIME!

Stay Inbounds

Solution code:
shorturl.at/ehEOW



Knowing the screen boundaries is important for creating your game!

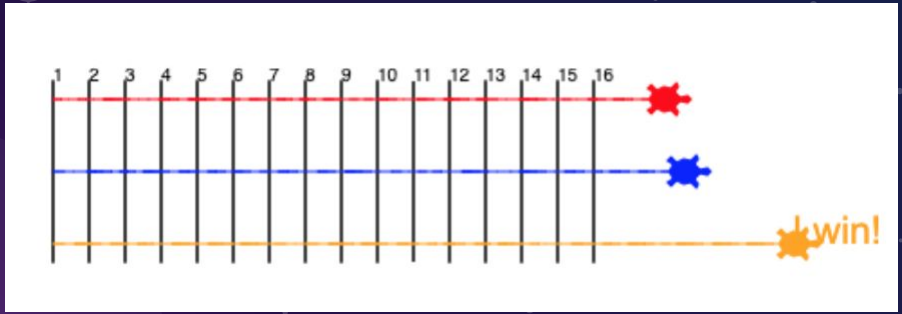
Left Key: turn left by 10 deg
Right Key: turn right by 10 deg

Hint: Use if/else, turtle.xcor(), turtle.ycor()

CODING TIME!

Turtle Race

Solution code:
shorturl.at/bhqsW



- Game starts when **mouse is clicked**
- 3 turtles, each moves forward a **random distance** every step (for 25 steps)
- Furthest turtle wins

NEW CONCEPT

% : the remainder / mod operator

5 % 3 = 2

1 remainder 2

NEW CONCEPT

% : the remainder / mod operator

```
5 % 3 = 2
```

1 remainder 2

```
10 % 7 = ?
7 % 8 = ?
18 % 2 = ?
```

NEW CONCEPT

% : the remainder / mod operator

```
5 % 3 = 2
```

1 remainder 2

```
10 % 7 = 3
7 % 8 = 7
18 % 2 = 0
```

NEW CONCEPT

% : the remainder / mod operator

```
5 % 3 = 2
```

1 remainder 2

```
10 % 7 = 3
7 % 8 = 7
18 % 2 = 0
```

How would you use the % operator to check if a number is even or odd?

NEW CONCEPT

While Loops

```
now = 1
while now < 6:
    print(now)
    now = now + 1
```

Console

1

now

NEW CONCEPT

While Loops

```
now = 1
while now < 6:
    print(now)
    now = now + 1
```

Console

1

2

now

NEW CONCEPT

While Loops

```
now = 1
while now < 6:
    print(now)
    now = now + 1
```

Console

1
2

3

now

NEW CONCEPT

While Loops

```
now = 1
while now < 6:
    print(now)
    now = now + 1
```

Console

```
1
2
3
```

4

now

NEW CONCEPT

While Loops

```
now = 1
while now < 6:
    print(now)
    now = now + 1
```

Console

1
2
3
4

5

now

NEW CONCEPT

While Loops

```
now = 1
while now < 6:
    print(now)
    now = now + 1
```

Console

```
1
2
3
4
5
```

6

now

NEW CONCEPT

While Loops

```
now = 1
while now < 6:
    print(now)
    now = now + 1
```

For Loops

```
max = 6
now = 1
for i in range(max):
    print(now)
    now = now + 1
```

What is the difference?

NEW CONCEPT

while True / break Pattern

```
x = 0
while True:
    print(x)
    x = x + 1
    if x > 10:
        break
```

CODING TIME!

While Loops

Remember to TEST your functions!

Problem One:

- Write a function **printWhile(min,max)** that prints the numbers max-min, inclusive
- printWhile(0,5)
  - 5...4...3...2...1...0
- printWhile(6,7)
  - 7...6

Starter code: shorturl.at/sBG67

Problem Two:

The Hailstone sequence takes a number n

- Stop when n = 1
- If n is even, n = n/2
- If n is odd, n = 3n + 1

Hint: Use the break statement to stop

While Loops Demo

https://repl.it/@jessicae5/whileSquares

I will write a function that draws 100 squares, using a while loop

```
done = False
squaresDrawn = 0
while(done == False)
    if squaresDrawn == 100:
    done = True
```

Here, we are checking for a condition - whether we have finished drawing all the squares

15 min break 🥳

NEW CONCEPT

Lists

```
groceries = ["Candy", "Milk", "Durian"]
groceries.append("Pizza")
print(groceries)
```

NEW CONCEPT

Lists

```
groceries = ["Candy", "Milk", "Durian"]
groceries.append("Pizza")
print(groceries)
```

Console: [“Candy”, “Milk”, “Durian”, “Pizza”]

NEW CONCEPT

Lists

```
students = ["Cindy", “Jake”, “Peter”]
students.append("Ray")
students.append(“Derek”)
print(students)
```

NEW CONCEPT

Lists

```
students = ["Cindy", “Jake”, “Peter”]
students.append("Ray")
students.append(“Derek”)
print(students)
```

Console: ["Cindy", “Jake”, “Peter”, “Ray”, “Derek”]




NEW CONCEPT

Adding to Lists

Option 1: Manually set values in a list:

```
list1 = [10, 20]
```

Option 2: Using the “append” function

```
list1.append(30)
```

NEW CONCEPT

Getting Every Item in a List

```
students = [“Frank”, “Alex”, “Charles”,
   “Steven”, “Allen”, “Sunny”, “Albert”]
for student in students:
   print(student)
```

NEW CONCEPT

Getting Every Item in a List

```
students = [“Frank”, “Alex”, “Charles”,
   “Steven”, “Allen”, “Sunny”, “Albert”]
for student in students:
   print(student)
```






NEW CONCEPT

What does this print?

```
teachers = [“Matt”, “Cat”, “Sophia”]
teachers.append(“Jessica”)
for t in teachers:
    print(t)
```

NEW CONCEPT

What does this print?

```
teachers = [“Matt”, “Cat”, “Sophia”]
teachers.append(“Jessica”)
for t in teachers:
    print(t)
```

Console: ["Matt", “Cat”, “Sophia”, “Jessica”]




CODING TIME!

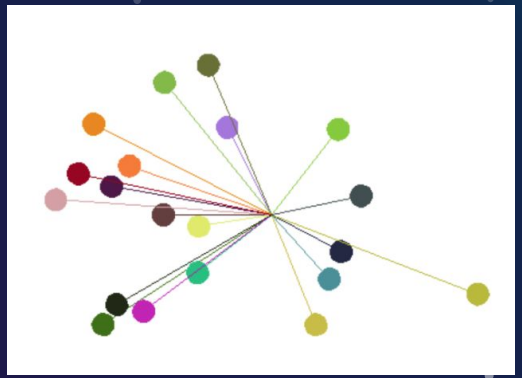
Lists

https://repl.it/@jessicae5/turtleCircle

- write the function **def makeTurtles(numberTurtles):**
- make a for loop that repeats **numberTurtles** times
- Inside the for loop make and save a new turtle each time
- Append the new turtle to the global list **turtles**

Remember to TEST your function!

Starter code: shorturl.at/nEMR6



Look at function moveTurtles()

- How does it talk to every turtle?
- What does it do to every turtle?
- What comments should we add?

NEW CONCEPT

List Size and Values

- Use `len()` to get the number of items in a list:

  ```
  len(list)
  ```

- Can access specific values using:
  `list[0]`

  Why? As computers start counting from 0!
  To access 1st element: `list[0]`
  To access 2nd element: `list[1]`
  To access 3rd element: `list[2]`

CODING TIME!

More Lists 😁

1) Write a function that takes in a list and returns the number of odd numbers

Ex. list1 = [1, 2, 6, 3, 8, 10, 11, 13, 15, 29] → return 6

Ex. list2 = [9, 3, 10, 11, 15] → return 4

2) Write a function that takes in a list and returns the sum of the list

Ex. list1 = [1, 2, 3, 4, 5] → return 15

Ex. list2 = [3, 8, 2, 1, 6, 3] → return 23

3) Write a function that takes in a list of numbers and returns the max value

Ex. list1 = [1, 3, 9, 10, 4, 6, 2] → return 10

Ex. list2 = [8, 2, 10, 3, 16, 17, 4, 23, 5] → return 23

4) Write a function that takes in a list of words and returns a list of only the words that start with 's'

Ex. list1 = ["apple", "strawberry", "banana", "stars", "snake", "bed", "shoe"]
→ return ["strawberry", "stars", "snake", "shoe"]

Today's Overview

- **%** operator
- **while** loops
- lists