



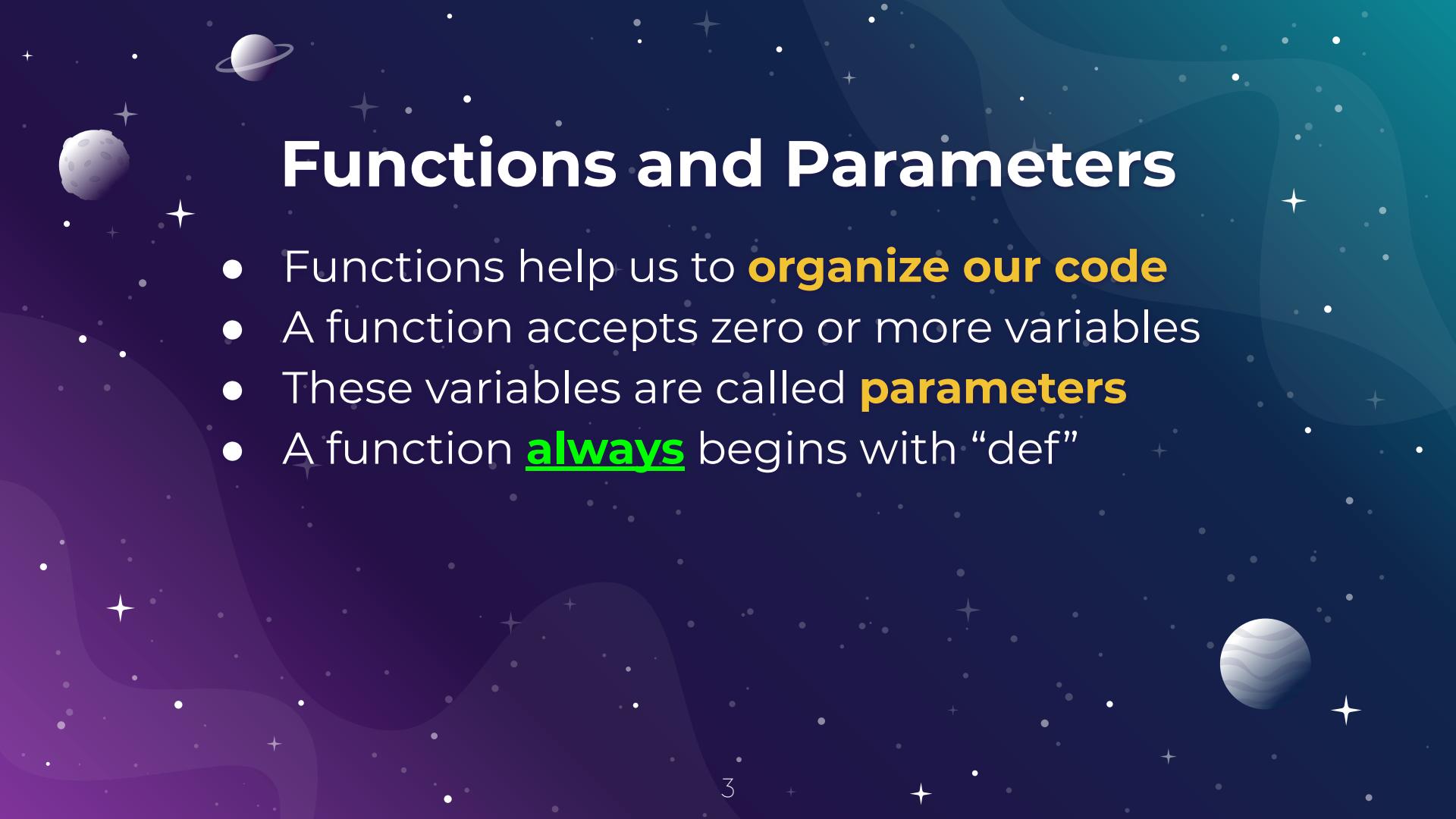
# CODE FOR CHINA

## LESSON 4



# Review of Lesson 3

- Functions
- Event Listeners



# Functions and Parameters

- Functions help us to **organize our code**
- A function accepts zero or more variables
- These variables are called **parameters**
- A function **always** begins with “def”



# Functions and Parameters

- Functions help us to **organize our code**
- A function accepts zero or more variables
- These variables are called **parameters**
- A function **always** begins with “def”

Quiz: How many parameters does this function have?

```
def rgb(low,high):  
    color = random.randint(low,high)  
    return color
```





# Functions and Parameters

- Functions help us to **organize our code**
- A function accepts zero or more variables
- These variables are called **parameters**
- A function **always** begins with “def”

```
def rgb(low,high):  
    color = random.rand  
    return color
```

Quiz: How many parameters does this function have?

Answer: 2

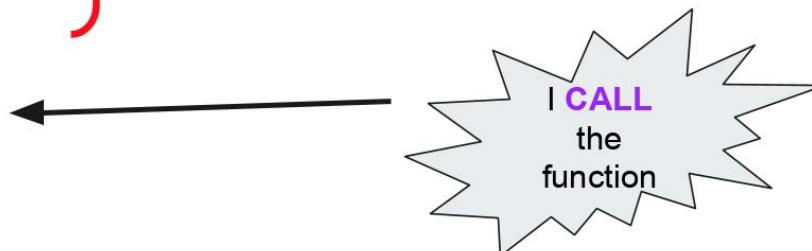
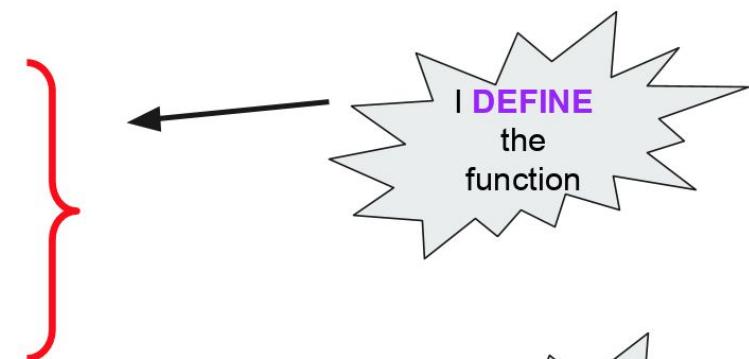
# Functions and Parameters

```
import turtle  
screen = turtle.Screen()  
turtle = turtle.Turtle()
```

```
def drawSquare(size):  
    angle = 90  
    for i in range(4):  
        turtle.forward(size)  
        turtle.right(angle)
```

```
length = 150  
drawSquare(length)
```

```
screen.exitonclick()
```



# Event Listeners

```
def forward():
    turtle.forward(100)

screen.onkey(forward, "Up")
screen.listen()
```

Step 2:  
response

Step 1:  
action

- Event listeners **capture user input**
- Think about how you want users to **interact with your game!**

# Today's Overview

- Function Warmups
- If / Else Statements

# Function Warmups

- 1) **multiply:** Takes 2 parameters and returns the product
- 2) **countToOneHundred:** Takes 1 parameter and prints out values from that value counting up to 100
- 3) **countToZero:** Takes 1 parameter and prints out values from that value counting down to zero

# if / else Statements

```
def foo(x):  
    if x > 0:  
        print(x)  
    else:  
        print(-x)  
  
foo(-15)
```

- a) 15
- b) x
- c) -15
- d) -x

## NEW CONCEPT

# if / else Statements

```
def foo(x):  
    if x > 0:  
        print(x)  
    else:  
        print(-x)  
  
foo(-15)
```

- a) 15
- b) x
- c) -15
- d) -x

## NEW CONCEPT

# if / else Statements

```
def blah(x):  
    if x < 0:  
        x = -x  
    print(-x)
```

```
blah(-3)
```

- a) 3
- b) x
- c) -3
- d) -x

## NEW CONCEPT

# if / else Statements

```
def blah(x):  
    if x < 0:  
        x = -x  
    print(-x)
```

```
blah(-3)
```

- a) 3
- b) x
- c) -3
- d) -x

# if / else Statements

```
def blah(x):  
    if x < 0:  
        x = -x  
    print(-x)
```

```
blah(-3)
```

How many lines of code in the function are read?

- a) 1
- b) 2
- c) 3
- d) 4

# if / else Statements

```
def blah(x):  
    if x < 0:  
        x = -x  
    print(-x)
```

```
blah(-3)
```

How many lines of code in the function are read?

- a) 1
- b) 2
- c) 3
- d) 4

# if / else Statements

```
def blah(x):  
    if x < 0:  
        x = -x  
    print(-x)
```

```
blah(5)
```

How many lines of code in the function are read?

- a) 1
- b) 2
- c) 3
- d) 4

# if / else Statements

```
def blah(x):  
    if x < 0:  
        x = -x  
    print(-x)
```

```
blah(5)
```

How many lines of code in the function are read?

- a) 1
- b) 2
- c) 3
- d) 4

## CODING TIME!

# Write a Function

- Parameter 1: money
- Parameter 2: tooMuchMoney
- if money greater than tooMuchMoney,  
divide money by 10
- else multiply money by 4
- print money

**Call function with:**

```
money = 120  
tooMuchMoney = 100
```

# 15 min break



## NEW CONCEPT

# if / elif / else Statements

```
def activity(time):  
    if time < 7:  
        return "sleeping"  
    elif time < 8:  
        return "eating"  
    elif time < 12:  
        return "hungry"  
    else:  
        return "reading"  
  
activity(8)
```

- a) "sleeping"
- b) "eating"
- c) "hungry"
- d) "reading"

## NEW CONCEPT

# if / elif / else Statements

```
def activity(time):  
    if time < 7:  
        return "sleeping"  
    elif time < 8:  
        return "eating"  
    elif time < 12:  
        return "hungry"  
    else:  
        return "reading"  
  
activity(8)
```

- a) "sleeping"
- b) "eatina"
- c) "hungry"
- d) "reading"

## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age):  
    driveAge = 18  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    else:  
        print("You can drive")
```

```
canYouDrive(14)
```



- a) "You cannot drive"
- b) "Take Lessons"
- c) "You can drive"



## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age):  
    driveAge = 18  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    else:  
        print("You can drive")
```

```
canYouDrive(14)
```



- a) "You cannot drive"
- b) "Take Lessons"
- c) "You can drive"

## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age) :  
    driveAge = 18  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    else:  
        print("You can drive")  
  
canYouDrive(14)
```



How many lines of code in the function are read?

- a) 1
- b) 2
- c) 3
- d) 4
- e) 5
- f) 6

## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age) :  
    driveAge = 18  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    else:  
        print("You can drive")  
  
canYouDrive(14)
```



How many lines of code in the function are read?

- a) 1
- b) 2
- c) 3
- d) 4
- e) 5
- f) 6

## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age):  
    driveAge = 18  
    maxDriveAge = 75  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    elif age <= maxDriveAge:  
        print("You can drive")  
    else:  
        print("You are too old")
```

```
canYouDrive(75)
```



- a) "You cannot drive"
- b) "Take Lessons"
- c) "You can drive"
- d) "You are too old"

## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age):  
    driveAge = 18  
    maxDriveAge = 75  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    elif age <= maxDriveAge:  
        print("You can drive")  
    else:  
        print("You are too old")
```

```
canYouDrive(75)
```



- a) "You cannot drive"
- b) "Take Lessons"
- c) "You can drive"
- d) "You are too old"

## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age):  
    driveAge = 18  
    maxDriveAge = 75  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    elif age <= maxDriveAge:  
        print("You can drive")  
    else:  
        print("You are too old")  
  
canYouDrive(75)
```

How many lines of code in the function are read?

- a) 5
- b) 6
- c) 7
- d) 8
- e) 9
- f) 10



## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age):  
    driveAge = 18  
    maxDriveAge = 75  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    elif age <= maxDriveAge:  
        print("You can drive")  
    else:  
        print("You are too old")  
  
canYouDrive(75)
```

How many lines of code in the function are read?

- a) 5
- b) 6**
- c) 7
- d) 8
- e) 9
- f) 10

## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age):  
    driveAge = 18  
    maxDriveAge = 75  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    elif age <= maxDriveAge:  
        print("You can drive")  
    else:  
        print("You are too old")
```

```
canYouDrive(77)
```



- a) "You cannot drive"
- b) "Take Lessons"
- c) "You can drive"
- d) "You are too old"

## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age):  
    driveAge = 18  
    maxDriveAge = 75  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    elif age <= maxDriveAge:  
        print("You can drive")  
    else:  
        print("You are too old")
```

```
canYouDrive(77)
```



- a) "You cannot drive"
- b) "Take Lessons"
- c) "You can drive"
- d) "You are too old"

## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age):  
    driveAge = 18  
    maxDriveAge = 75  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    elif age <= maxDriveAge:  
        print("You can drive")  
    else:  
        print("You are too old")
```

```
canYouDrive(77)
```

How many lines of code in the function are read?

- a) 5
- b) 6
- c) 7
- d) 8
- e) 9
- f) 10

## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age):  
    driveAge = 18  
    maxDriveAge = 75  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    elif age <= maxDriveAge:  
        print("You can drive")  
    else:  
        print("You are too old")  
  
canYouDrive(77)
```

How many lines of code in the function are read?

- a) 5
- b) 6
- c) 7
- d) 8
- e) 9
- f) 10

## NEW CONCEPT

# if / elif / else Statements

```
def canYouDrive(age):  
    driveAge = 18  
    maxDriveAge = 75  
    if age < driveAge:  
        print("You cannot drive")  
    elif age == driveAge:  
        print ("Take Lessons")  
    elif age <= maxDriveAge:  
        print("You can drive")  
    else:  
        print("You are too old")
```

```
canYouDrive(77)
```



What happens if we change the ordering of the if / elif / else statements?

## CODING TIME!

# Functions

- Parameter: `number`
- if `number` is smaller than zero:  
print “Smaller than zero”
- if `number` is equal to zero: print  
“Equal to zero”
- if a `number` is bigger than zero:  
print “Bigger than zero”

- 1) Remember to **TEST** your functions!
- 2) Give your functions **clear names**

- Parameter: `angle`
- if `angle` is 90, draw a square size 100
- if `angle` is 120, draw a triangle size 100
- else, draw a circle

# 15 min break



# Math with **if** statements

```
def doMath(firstNum, secondNum):  
    if firstNum - secondNum >= 0:  
        print("firstNum is greater than or equal to secondNum")  
    else:  
        print("firstNum is smaller than secondNum")
```

doMath(5, 10)

# Math with if statements

```
def doMath(firstNum, secondNum):  
    if firstNum - secondNum >= 0:  
        print("firstNum is greater than or equal to secondNum")  
    else:  
        print("firstNum is smaller than secondNum")
```

doMath(5, 10)

firstNum is smaller than secondNum

# Nested if 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")
```

doMath(10, 10)

## NEW CONCEPT

# Nested if 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")
```

doMath(10, 10)

they are equal

## NEW CONCEPT

# Nested if 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

doMath(10, 10)

they are equal

## NEW CONCEPT

# 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")
```

```
doMath(10, -10)
```

## NEW CONCEPT

# 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")
```

doMath(10, -10)

firstNum is positive only

# and statements: Same Result?

```
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")
```

doMath(10, -10)

```
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")  
    else:  
        print("secondNum is positive only")
```

## NEW CONCEPT

# and statements: Same Result?

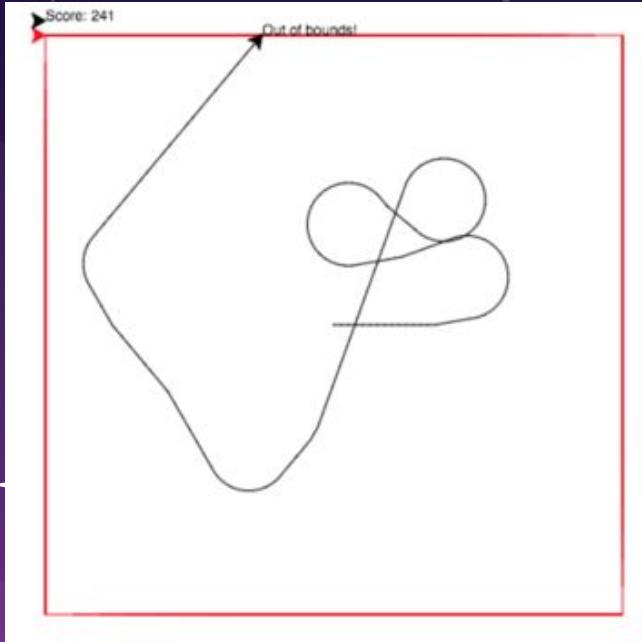
```
def doMath(firstNum, secondNum):  
    if firstNum > 0 and secondNum > 0:  
        print("Both numbers are positive")  
    elif firstNum > 0 and secondNum < 0:  
        print("firstNum is positive only")  
    elif firstNum < 0 and secondNum > 0:  
        print("secondNum is positive only")
```

doMath(10, -10)

```
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")  
    else:  
        print("secondNum is positive only")
```

No, the second function is wrong: does not account for when both numbers are negative

# Stay Inbounds



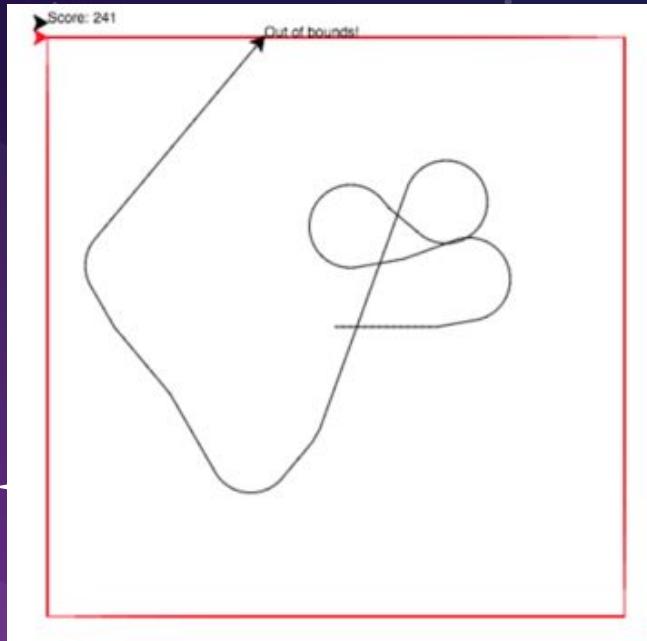
Volunteer to play  
game

## Checking Turtle Position

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

## CODING TIME!

# Stay Inbounds



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()

# Mouse Click Event

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

```
def mouseClicked(x, y):  
    ### CODE HERE ###  
  
screen.onclick(mouseClicked)  
screen.listen()
```

# Today's Overview

- **if / elif / else** statements
- nested **if** statements
- **and** statements
- Turtle Position