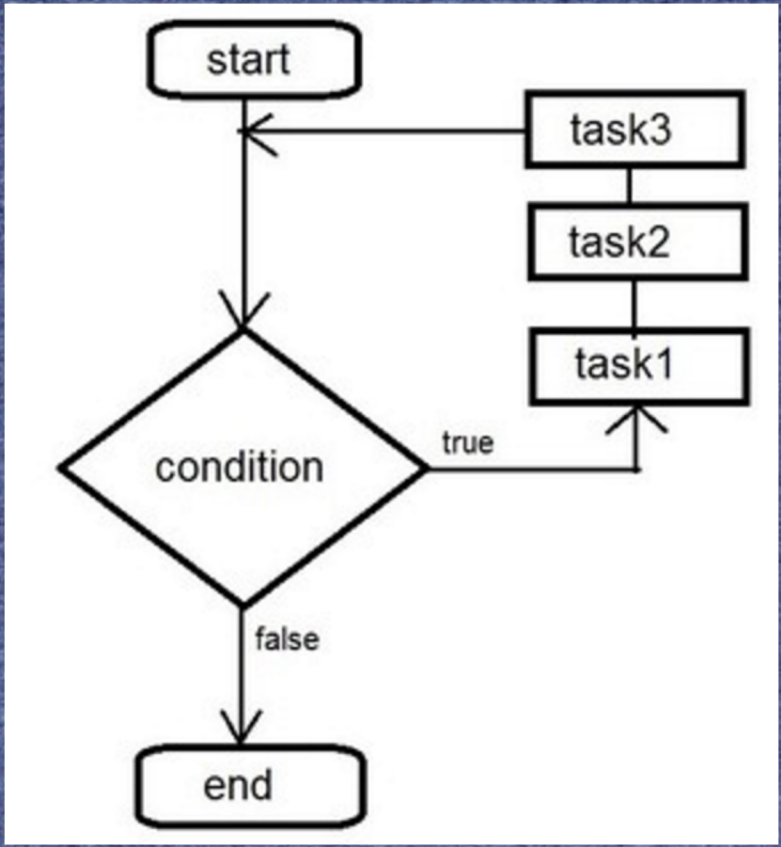
more Iteration



For & While Loops

- For Loop reloaded
- While Loop
- Randomly Walking Turtle
- For vs While

For Loop

- Iterating through items

```
for f in ["Joe", "Amy", "Brad", "Angelina"]:
    print("Hi", f, "Please come to my party on Saturday")
```

For Loop

- Iterating through items

```
for f in ["Joe", "Amy", "Brad", "Angelina"]:
    print("Hi", f, "Please come to my party on Saturday")
```

Works for Any Sequence variable

For Loop

- Iterating through items

```
text = "This is a string"
for f in text:
    print(f)
```

- Iterating through items

```
listofnumbers = [1, 2, 3, 4, 5]
for f in listofnumbers:
    print(f)
```

For Loop

- Accumulator pattern

```
def sumTo(aBound):
    theSum = 0
    for aNumber in range(1, aBound + 1):

    ret

print(sumTo(4))

print(sumTo(1000))
```

Sequence variable

```
for varA in varB     #varB must be
                     #  sequence type
```

For Loop

- Accumulator pattern

```
def sumTo(aBound):
    theSum = 0
    for aNumber in range(1, aBound + 1):

    ret

print(

print(sumTo(1000))
```

Sequence variable

```
for varA in varB     #varB must be
                     #  sequence type
```

Means

range(1,10) is a sequence type

For Loop

- Accumulator pattern

```
for varA in range(1,10):
    print(varA)
```

For Loop

- Accumulator pattern

```
for aNumber in range(1, 4):
        theSum = theSum + aNumber

print(theSum)
```

For Loop

- Accumulator pattern

```
theSum = 0

for aNumber in range(1, 4):
        theSum = theSum + aNumber

print(theSum)
```

For Loop

- Accumulator pattern

```
def sumTo(aBound):
    theSum = 0
    for aNumber in range(1, aBound + 1):
        theSum = theSum + aNumber

    return theSum

print(sumTo(4))

print(sumTo(1000))
```

For Loop vs While Loop

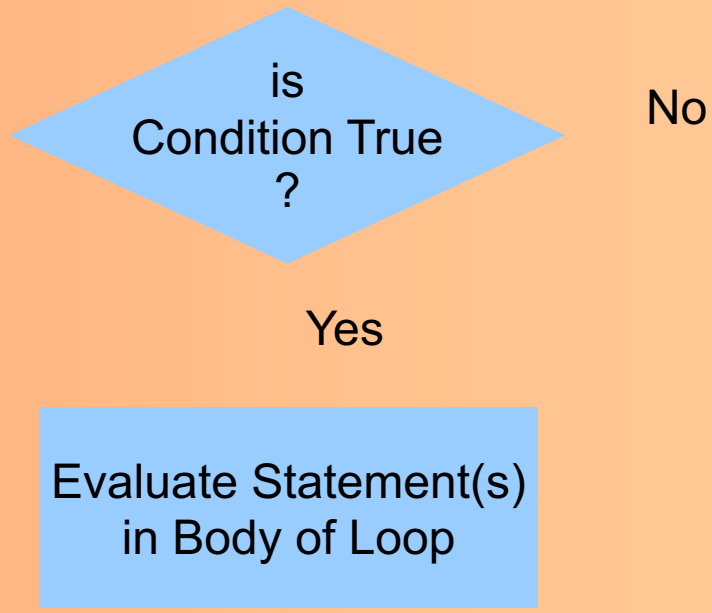
- For loops
  - –specific number of times
- While loops
  - –while condition is true

While vs For Loop

- average 10 students scores
- get input until user enters 'F'
- as long as value is less than 20
- simulate tossing a coin 100 times

- For loops
  - specific number of times
- While loops
  - while condition is true

While Loop


is
Condition True
?
No
Yes
Evaluate Statement(s)
in Body of Loop

Randomly Walking Turtle

- Turtle starts in middle of screen
- flip a coin
  - if coin is heads turn turtle left 90
  - if coin is tails turn turtle right 90
- go 50 steps forward
- if turtle is outside screen stop

Randomly Walking Turtle

- Turtle starts in middle of screen
- flip a coin
  - if coin is heads turn turtle left 90
  - if coin is tails turn turtle right 90
- go 50 steps forward
- if turtle is outside screen stop

What Type of Loop ?

Randomly Walking Turtle

create a window and a turtle

while the turtle is still in the window:
    generate a random number between 0 and 1
    if the number == 0 (heads):
        turn left
    else:
        turn right
    move the turtle forward 50

Randomly Walking Turtle

create a window and a turtle

while **the turtle is still in the window**:
    generate a random number between 0 and 1
    if the number == 0 (heads):
        turn left
    else:
        turn right
    move the turtle forward 50

Randomly Walking Turtle

create a window and a turtle

while **the turtle is still in the window**:
    generate a random number between 0 and 1
    if the number == 0 (heads):
        turn left
    else:
        turn right
    move the turtle forward 50

Create function
  returns true or false

```
def isInWindow():
  return True
```

Randomly Walking Turtle

create a window and a turtle

while **the turtle is still in the window**:
    generate a random number between 0 and 1
    if the number == 0 (heads):
        turn left
    else:
        turn right
    move the turtle forward 50

Generate random number

```
import random

....

coin = random.randrange(0, 2)
```

Randomly Walking Turtle

create a window and a turtle

while **the turtle is still in the window**:
    generate a random number between 0 and 1
    if the number == 0 (heads):
        turn left
    else:
        turn right
    move the turtle forward 5(

Create function
  returns true or false

```
def isInWindow(screen, turtle):

        return True
```

Randomly Walking Turtle

create a window and a turtle

while **the turtle is still in the window**:
    generate a random number between 0 and 1
    if the number == 0 (heads):
        turn left
    else:
        turn right
    move the turtle forward 50

```
def headstails():

coin = random.randrange(0, 2)

def isInWindow(screen, turtle):
```

Success Model!

Randomly Walking Turtle

create a window and a tu

while **the turtle is still i**
    generate a random nu
    if the number == 0 (h
        turn left
    else:
        turn right
    move the turtle forwa

```
import turtle
import random

def headstails():
    return(True)

def isInWindow(win, turt):
    return(True)

wn = turtle.Screen()
rwt = turtle.Turtle()

while(inWindow(wn, rwt)):
        if headstails():
                rwt.left(90)        #turn left
        else:
                rwt.right(90)       #turn right

        rwt.forward(50)             #forward 50
```

Excellent!

Success Model!

Randomly Walking Turtle

create a window and a tu

while **the turtle is still i**
    generate a random nu
    if the number == 0 (h
        turn left
    else:
        turn right
    move the turtle forwa

```
import turtle
import random

def headstails():
    coin = random.randrange(0, 2)

    return(True)
```

Success Model!

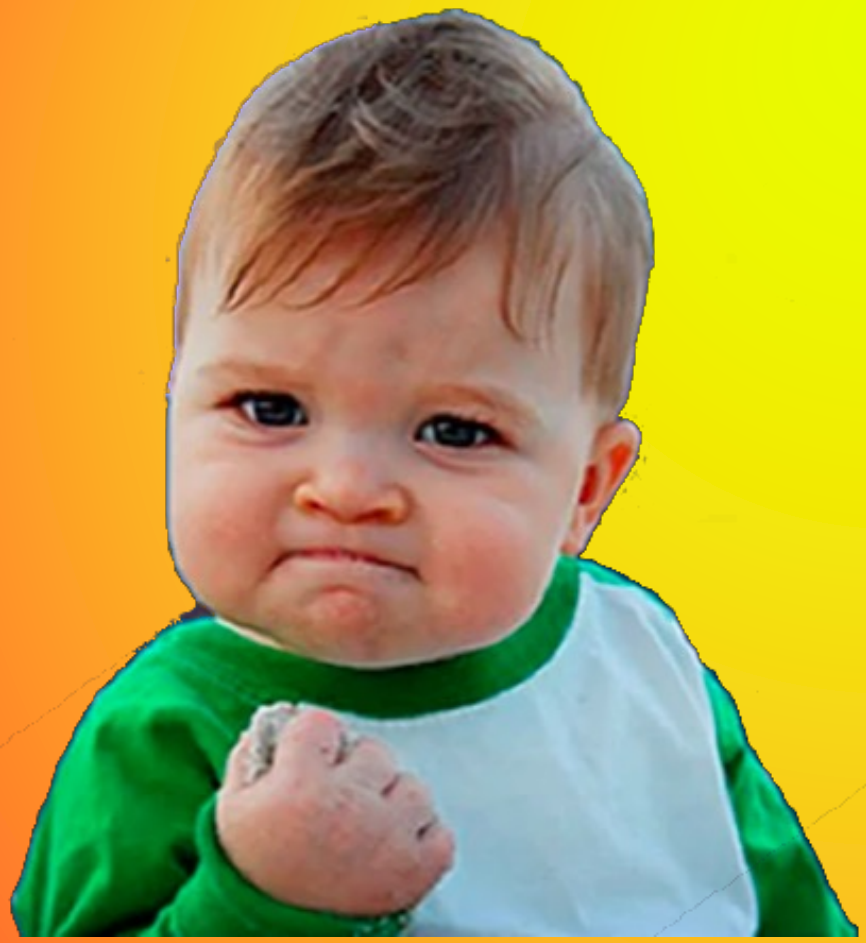
Randomly Walking Turtle

create a window and a tu

while **the turtle is still i**
    generate a random nu
    if the number == 0 (h
        turn left
    else:
        turn right
    move the turtle forwa

```
import turtle
import random

def headstails():
    coin = random.randrange(0, 2)

    if(coin == 0):
        return(True)
    else:
        return(False)
```

Most Excellent!



Success Model!

Randomly Walking Turtle

create a window and a tu

while **the turtle is still i**
    generate a random nu
    if the number == 0 (h
        turn left
    else:
        turn right
    move the turtle forwa

```
import turtle
import random

def headstails():
    return(True)

def isInWindow(win, turt):
    return(True)

wn = turtle.Screen()
rwt = turtle.Turtle()

while(inWindow(wn, rwt)):
        if headstails():
                rwt.left(90)        #turn left
        else:
                rwt.right(90)       #turn right

        rwt.forward(50)             #forward 50
```

Success Model!

Randomly Walking Turtle

create a window and a tu

while **the turtle is still i**
    generate a random nu
    if the number == 0 (h
        turn left
    else:
        turn right
    move the turtle forwa

```
import turtle
import random

def isInWindow(win, turt):
    return(True)


win.window_width()
win.window_height()

turt.xcor()
turt.ycor()
```

Success Model!

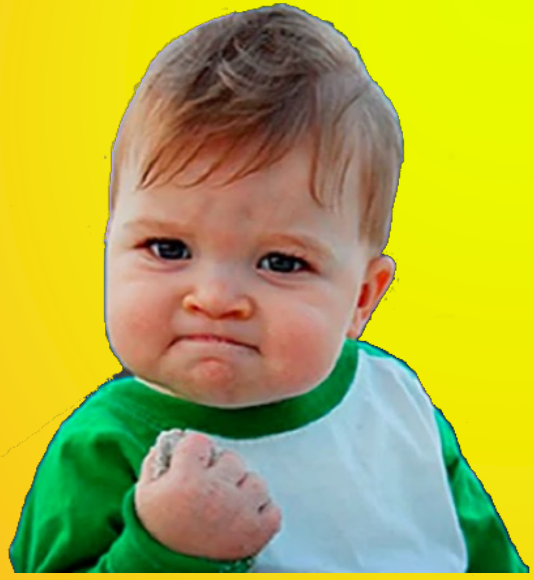
Randomly Walking Turtle

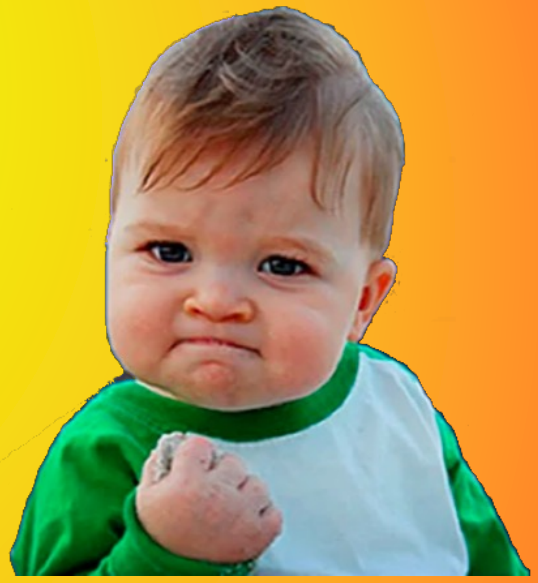
create a window and a tu

while **the turtle is still i**
    generate a random nu
    if the number == 0 (h
        turn left
    else:
        turn right
    move the turtle forwa

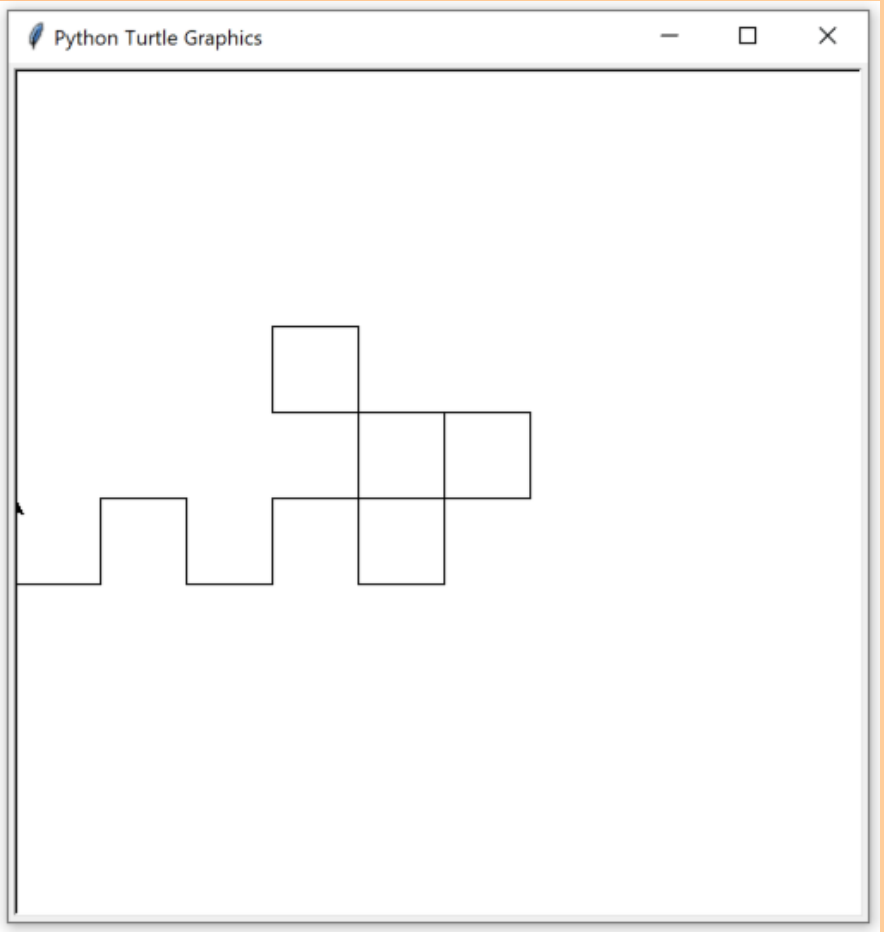
```
import turtle
import random

def inWindow(win, turt):
    leftBound = -(win.window_width() / 2)
    rightBound = win.window_width() / 2
    topBound = win.window_height() / 2
    bottomBound = -(win.window_height() / 2)

    tX = turt.xcor()
    tY = turt.ycor()

    stillIn = True
    if tX > rightBound or tX < leftBound:
        stillIn = False
    if tY > topBound or tY < bottomBound:
        stillIn = False

    return stillIn
```

Awesomly Excellent!



```
import turtle
import random

def headstails():
    coin = random.randrange(0, 2)
    print(coin)
    if coin == 0:
        return(True)
    else:
        return(False)

def inWindow(win, turt):
    leftBound = -(win.window_width() / 2)
    rightBound = win.window_width() / 2
    topBound = win.window_height() / 2
    bottomBound = -(win.window_height() / 2)

    tX = turt.xcor()
    tY = turt.ycor()

    stillIn = True
    if tX > rightBound or tX < leftBound:
        stillIn = False
    if tY > topBound or tY < bottomBound:
        stillIn = False

    return stillIn

wn = turtle.Screen()
wn.setup(width=500, height=500)

rwt = turtle.Turtle()

while(inWindow(wn, rwt)):
                    if headstails():
                                        rwt.left(90)        #turn left
                    else:
                                        rwt.right(90)       #turn right

                    rwt.forward(50)                         #forward 50
                    print(rwt.xcor(), rwt.ycor())

#turtle.bye()
print("Your Turtle has left the window")
```



For & While Loops

- For Loop reloaded
- While Loop
- Randomly Walking Turtle
- For vs While