# `for` Loop in Python

# Scratch loop

# Review from Friday: how would we write this using a while loop in Python?

## while loop

```
i = 0
while i < 5:
    print("Girls Who Code Rock!")
    i += 1
```

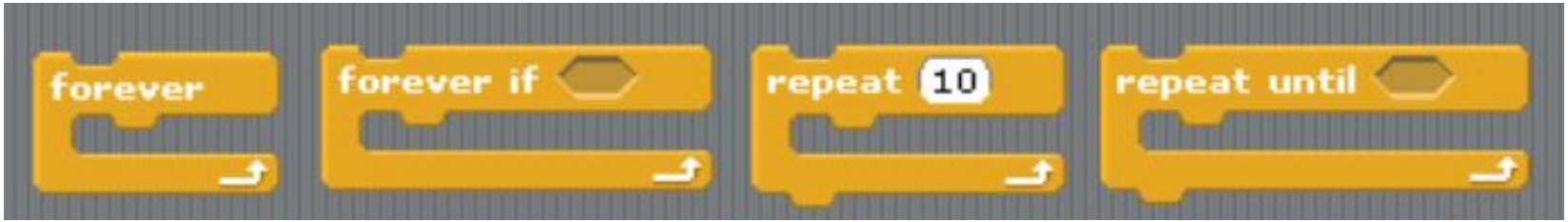
## for loop

```
for i in range(5):
  print("Girls Who Code Rock!")
```

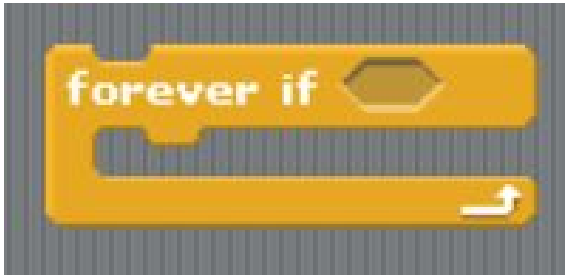
**Often we can accomplish the same thing with either loop.**

**So...when should you use one vs. the other?**

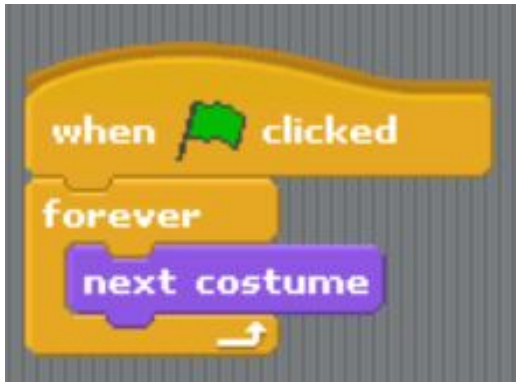
# Remember all the different looping blocks in Scratch?



# When do we repeat things forever?

# (Typically) When we have multiple tasks:




Scratch will do this for you.

Python won't.

So let's just worry about repeat blocks.


girls who code

## *Until [boolean]*
## do this


## *10 times*
## do this

## *Until [boolean]* do this

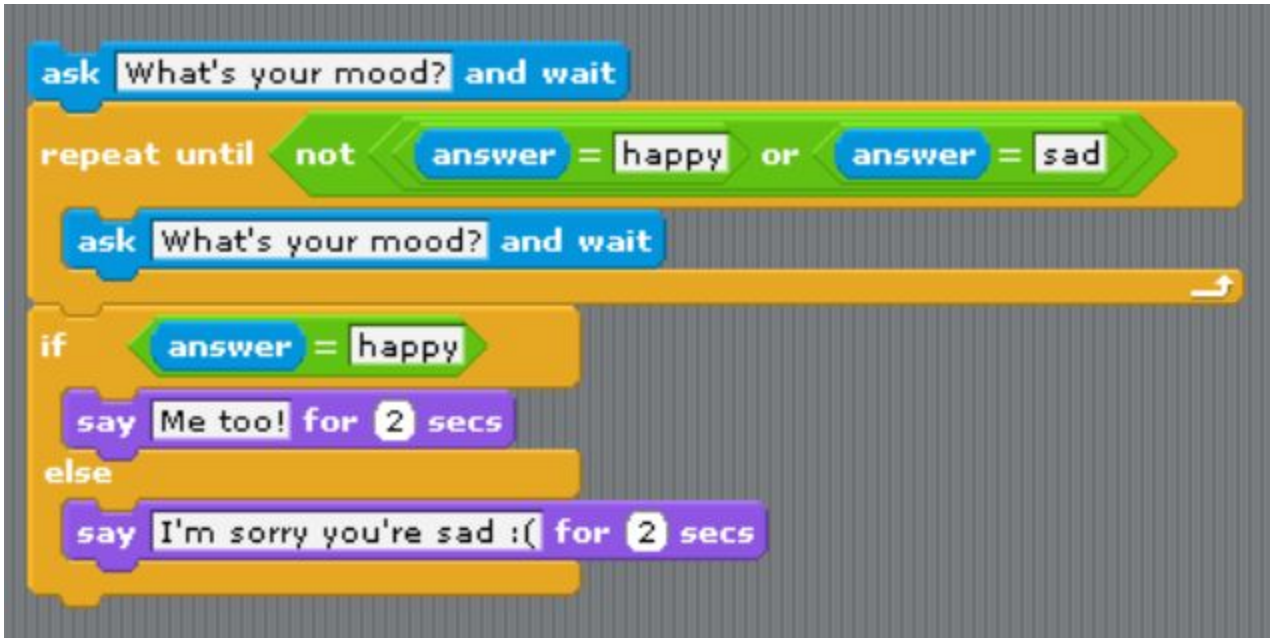

```
x_position = 0
x_edge = 100
while x_position < x_edge:
    x_position += 10
```

## *10 times* do this


```
for i in range(10):
    print("Hello, world!")
```

girls who code

# *Until [boolean]* do this



# *Until [boolean]* do this



```
answer = input("What is your mood?")
while(not(answer == "happy" or answer == "sad")):
    answer = input("What is your mood?")
if(answer == "happy"):
    print("Me too!")
else:
    print("I'm sorry you're sad :(")
```

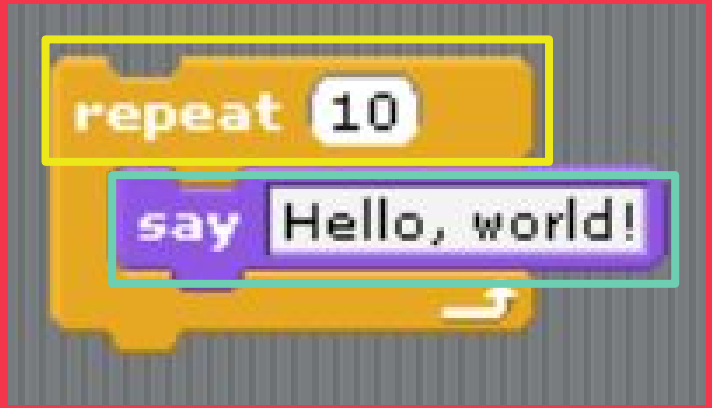
girls who code

## *Until [boolean]*
## do this


## *10 times*
## do this

```
for i in range(10):
    print("Hello, world!")
```

Same basic shape →



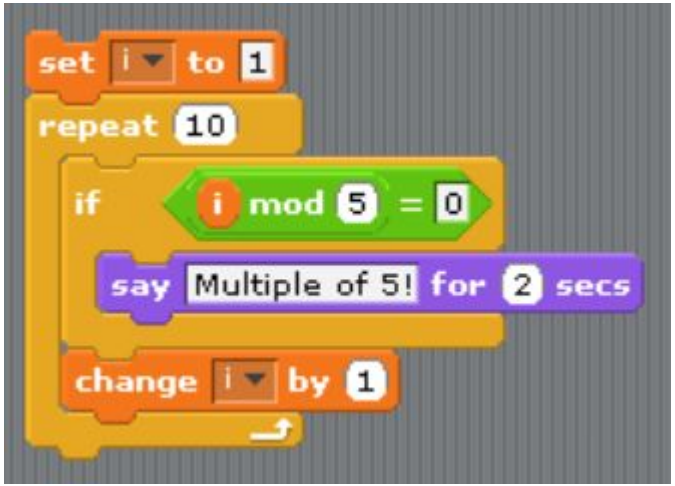

What's different?

*What is the point of i?*

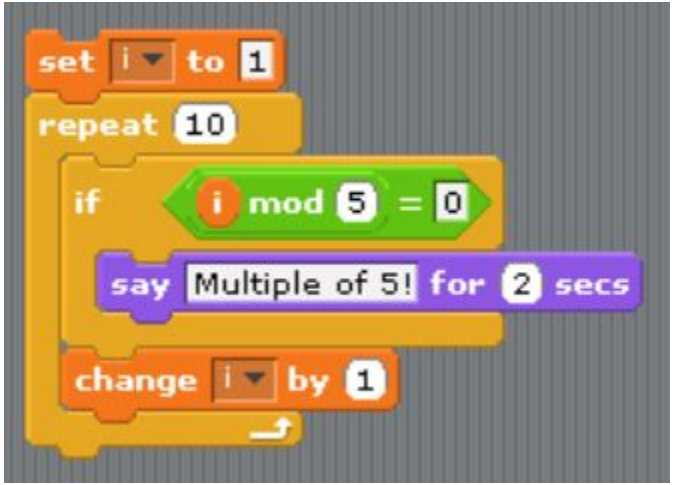
***10 times*** **do this**

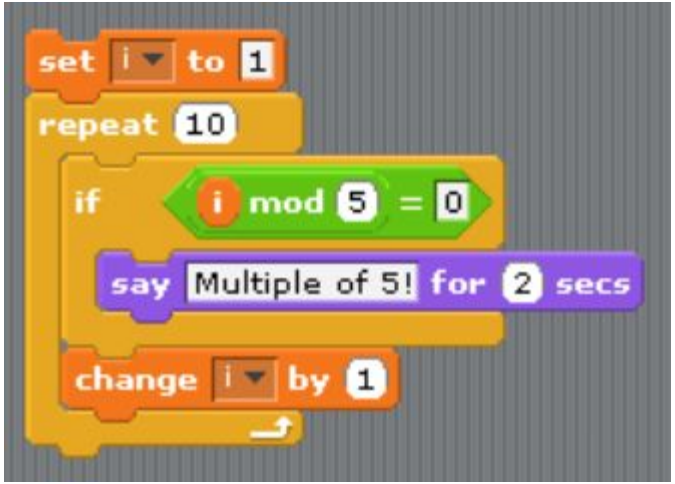

```
for i in range(10):
    print("Hello, world!")
```

# Use *i* to change the loop!

# How would we write this using a Python while loop?






```
i = 1
while i < 11:
    if(i % 5 == 0):
        print("Multiple of 5!")
    i += 1
```

girls who code

**For loops make it easier to *iterate* over a known set of data.**

***That's it.***

```
i = 1
while i < 11:
    if(i % 5 == 0):
        print("Multiple of 5!")
    i += 1
```

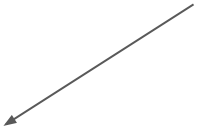
```
for i in range(1, 11):
    if(i % 5 == 0):
        print("Multiple of 5!")
```

# Python has all kinds of shortcuts

```
i = 1
while i < 11:
    if(i % 5 == 0):
        print("Multiple of 5!")
    i += 1
```



```
for i in range(1, 11):
    if(i % 5 == 0):
        print("Multiple of 5!")
```

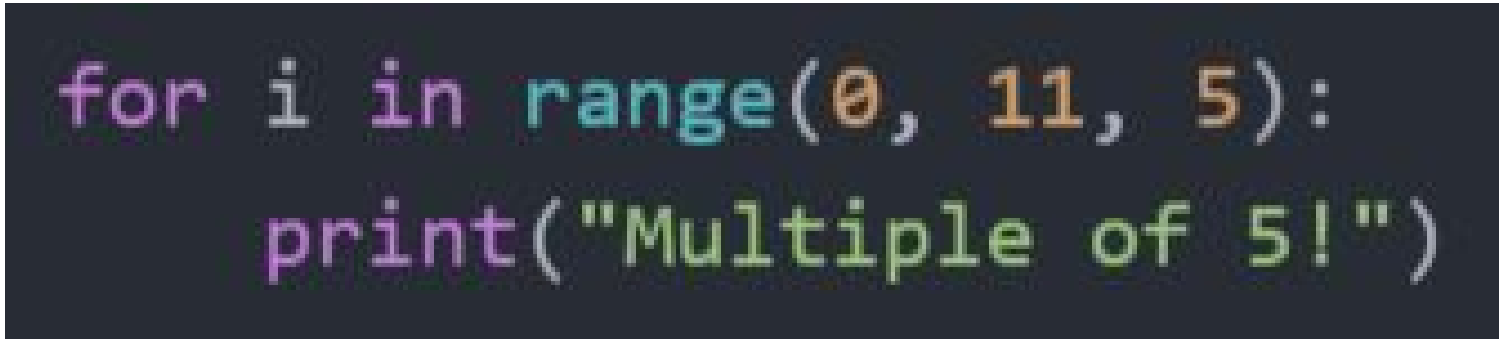


```
for i in range(0, 11, 5):
    print("Multiple of 5!")
```

Stop before here

Start here

Change i by this much



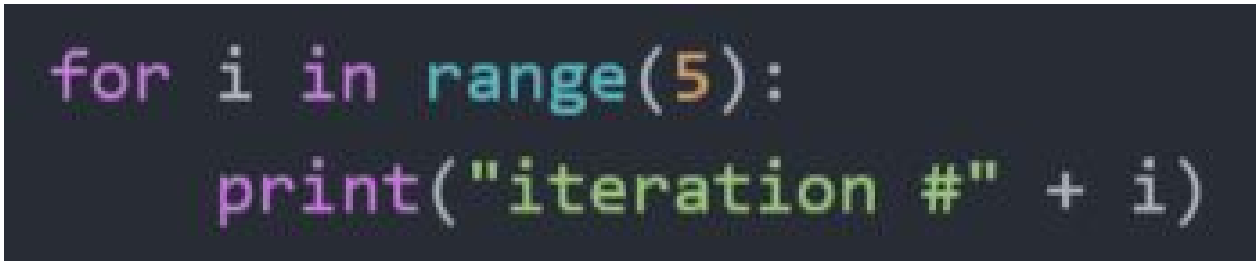

girls who code

# By default...

Stop before here

(Start at 0)

(Change i by 1)

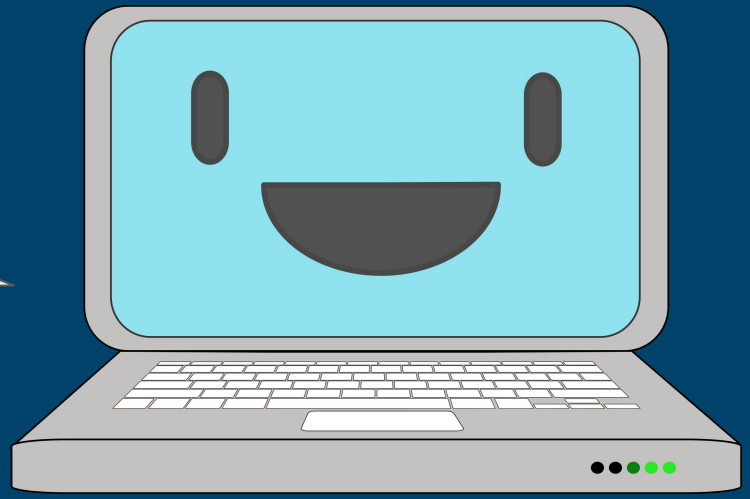


```
for i in range(5):
    print("iteration #" + i)
```

# Tips

```
for i in range(5):
    print("iteration #" + i)
```

When I see the colon I know that any following **INDENTED** lines is stuff I should repeat.



girls who code

```
for i in range(5):
    print("Girls Who Code Rock!")
print("I'm done repeating now!")
```

girls who code

# What happens when this code is run?

```
for i in range(2):
print("Print this twice please!")
```

# What happens when this code is run?

```
for i in range(100)
    print("I will always check the syntax, even the smallest thing!")
```

# What happens when this code is run?

```
for i in range(222):
    print("This might take a long time.")
print("It will stop eventually.")
```

# Bonus Question: What do you think will happen here?

```
for i in range(3):
    print("This is a loop, but...")
    for j in range(2):
        print("there is this other loop inside of it.")
```

girls who code