

Data Analytics + Python

Logical Flow
Control

- If / Else
- And / Or
- Loops
- Range

IF Statements

- Used with the logical operators

==

!=

<

>

<=

>=

a = 45

b = 68

If b > a:

 print("b is greater")

Elif Statement

Python's way of saying,
"If the previous
conditions were not
true, then try this
condition"

```
A = 55
```

```
B = 55
```

```
If b > a:
```

```
    print("b is greater than a")
```

```
elif b == a:
```

```
    print("a and b are equal")
```

ELSE Statement

Else statements are used to catch anything that wasn't already caught in the previous two conditions.

```
A = 200
```

```
B = 55
```

```
if b > a:
```

```
    print("b is greater than a")
```

```
elif b == a:
```

```
    print("a and b are equal")
```

```
else:
```

```
    print("a is greater than b")
```

You can have an else
without the elif!

```
A = 200
```

```
B = 55
```

```
if b > a:
```

```
    print("b is greater than a")
```

```
else:
```

```
    print("a is greater than b")
```

Using “And”

- Considered a logical operator
- It combines conditional statements together

```
a = 200
```

```
b = 33
```

```
c = 500
```

```
if a > b and c > a:
```

```
    print("Both conditions are True")
```

Using “or”

- Considered a logical operator
- It combines conditional statements together

```
a = 200  
b = 33  
c = 500
```

```
if a > b or a > c:  
    print("At least one of the conditions is True")
```

Nested IF Statement

If a statement is inside of a statement, then that means it is considered nested

```
X = 55
```

```
if x > 10:
```

```
    print("It's above 10,")
```

```
    if x < 20:
```

```
        print("and also about 20!")
```

```
    else:
```

```
        print("but not above 20")
```


While Loop

You can execute a set of statements as long as a condition is true.

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

1
2
3
4
5

For Loops

Used for iterating over a sequence (like a list, tuple, dictionary, set or a string.)

```
fruits = ["apple", "banana",  
"grape"]
```

```
for x in fruits:  
    print(x)
```

Break Statement

We can stop the loop even if the while condition is true

```
i = 1
while i < 6:
    print(i)
    if i == 3:
        break
    i += 1
```

Break Statement

We can stop the loop before it goes through all the items.

```
fruits = ["apple", "banana",  
"grape"]
```

```
for x in fruits:  
    print(x)  
    if x == "banana":  
        break
```

Another Break Example

```
fruits = ["apple", "banana", "grape"]
```

```
for x in fruits:  
    if x == "banana":  
        break  
    print(x)
```

Nested Loops

Goal: print each adj for every fruit

```
adj = ["red", "big", "juicy"]
```

```
fruits = ["apple", "orange", "pineapple"]
```

```
for x in adj:
```

```
    for y in fruits:
```

```
        print(x,y)
```

Else Statement

We can run a block of code once when the condition is no longer true.

```
i = 1
while i < 6:
    print(i)
    i += 1
else:
    print("i is no longer less than 6")
```

range() function as a loop

- Used to loop through a set of code a specified number of times.
- It returns a sequence of numbers, starting from 0 by default, and increments of 1 (by default) and ends at a specified number

```
For x in range(6):  
    print(x)
```


range() function as a loop

It is possible to specify the starting value by adding a parameter, telling python to start at that first number and go to the second (but not include it)

```
for x in range (2,6):  
    print(x)
```

range() function as a loop

It is possible to specify the increment value by adding a third parameter:

```
for x in range (2, 50, 5):  
    print(x)
```

range() function as a loop

The else keyword in a for loop specifies a block of code to be executed when the loop is finished:

```
for x in range(10):  
    print(x)  
else:  
    print("Finally finished!")
```

range() function as a loop

Now, let's break the loop and see what happens.

```
for x in range(10):  
    if x == 5: break  
    print(x)  
else:  
    print("Finally finished!")
```

Practice

- Create a file and name it statements
 - Create an if statement
 - Create an if statement using else
 - Create a nested loop
- Create a for loop and include a break
 - Loop through a string
- Send the file to me via Slack