If Statements (Conditional Statements) in Python

Just like other programming languages, Python supports the basic logical conditions from math:

- Equals: a == b
- Not Equals: a != b
- Less than: a < b
- Less than or equal to a <= b
- Greater than: a > b
- Greater than or equal to: a >= b

You can leverage these conditions in various ways. But most likely, you'll use them in "**if statements**" and **loops.**

If Statement Example

The goal of a conditional statement is to check if it's True or False.

```
if 5 > 1:
        print(“That’s True!”)
```

Output:

```
That’s True!
```

Nested If Statements

For more complex operations, you can create nested if statements. Here's how it looks:

```
x = 35

if x > 20:
  print(“Above twenty,”)
  if x > 30:
    print(“and also above 30!”)
```

Elif Statements

elif keyword prompts your program to try another condition if the previous one(s) was not true. Here's an example:

```
a = 45
b = 45
if b > a:
  print(“b is greater than a”)
elif a == b:
  print(“a and b are equal”)
```

If Else Statements

else keyword helps you add some additional filters to your condition clause. Here's how an if-elif-else combo looks:

```
if age < 4:
ticket_price = 0
elif age < 18:
ticket_price = 10
else: ticket_price = 15
```

If-Not-Statements

Not keyword let's you check for the opposite meaning to verify whether the value is NOT True:

```
new_list = [1, 2, 3, 4]
x = 10
if x not in new_list:
    print(“’x’ isn’t on the list, so this is True!”)
```

Pass Statements

If statements can't be empty. But if that's your case, add the **pass** statement to avoid having an error:

```
a = 33
b = 200

if b > a:
  pass
```