Python Conditions and If statements

- Python supports the usual logical conditions from mathematics
  - 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
- These conditions can be used in several ways, most commonly in "if statements" and loops

- An "if statement" is written by using the if keyword
- Example
  - If statement

In [1]:
```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

```
b is greater than a
```

Elif

- The elif keyword is Python's way of saying "if the previous conditions were not true, then try this condition"
- Example

In [4]:
```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

```
a and b are equal
```

Else

- The else keyword catches anything which isn't caught by the preceding conditions
- Example

In [2]:
```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

```
a is greater than b
```

- You can also have an else without the elif
- Example

In [3]:
```
a = 200
b = 33
if b > a:
  print("b is greater than a")
else:
  print("b is not greater than a")
```

```
b is not greater than a
```

Short Hand If

- If you have only one statement to execute, you can put it on the same line as the if statement
- Example
  - One line if statement

In [5]:
```
a = 200
b = 33

if a > b: print("a is greater than b")
```

```
a is greater than b
```

Short Hand If ... Else

- If you have only one statement to execute, one for if, and one for else, you can put it all on the same line
- Example
  - One line if else statement

In [6]:
```
a = 2
b = 330
print("A") if a > b else print("B")
```

```
B
```

- This technique is known as Ternary Operators , or Conditional Expressions

- You can also have multiple else statements on the same line
- Example
  - One line if else statement, with 3 conditions

In [7]:
```
a = 330
b = 330
print("A") if a > b else print("=") if a == b else print("B")
```

```
=
```

And

- The and keyword is a logical operator, and is used to combine conditional statements
- Example
  - Test if a is greater than b , AND if c is greater than a

In [8]:
```
a = 200
b = 33
c = 500
if a > b and c > a:
  print("Both conditions are True")
```

```
Both conditions are True
```

Or

- The or keyword is a logical operator, and is used to combine conditional statements
- Example
  - Test if a is greater than b, OR if a is greater than c

In [9]:
```
a = 200
b = 33
c = 500
if a > b or a > c:
  print("At least one of the conditions is True")
```

```
At least one of the conditions is True
```

Not

- The not keyword is a logical operator, and is used to reverse the result of the conditional statement
- Example
  - Test if a is NOT greater than b

In [10]:
```
a = 33
b = 200
if not a > b:
  print("a is NOT greater than b")
```

```
a is NOT greater than b
```

Nested If

- You can have if statements inside if statements, this is called ***nested* if** statements
- Example

In [11]:
```
x = 41

if x > 10:
  print("Above ten,")
  if x > 20:
    print("and also above 20!")
  else:
    print("but not above 20.")
```

```
Above ten,
and also above 20!
```

The pass Statement

- if statements cannot be empty, but if you for some reason have an if statement with no content, put in the pass statement to avoid getting an error
- Example

In [ ]:
```
a = 33
b = 200

if b > a:
  pass
```