

## Ques. What is If Else?

- 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`

```
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")
output:- a is greater than b
```

- **Short Hand If**

```
a = 200
b = 33

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

- **Short Hand If ... Else**

```
a = 2
b = 330

print("A") if a > b else print("B")
output:- B
```

- One line if else statement, **with 3 conditions:**

```
a = 330
b = 330

print("A") if a > b else print("=") if a == b else print("B")
Output:- =
```

- The **and** keyword is a logical operator, and is used to combine conditional statements.

```
a = 200
b = 33
c = 500
if a > b and c > a:
    print("Both conditions are True")
output:- Both conditions are True
```

- The **Or** keyword is a logical operator, and is used to combine conditional statements.

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

- **Nested If** You can have if statements inside if statements, this is called nested if statements.

```
x = 41

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

```
a = 33
b = 200

if b > a:
    pass
Output:-
```

## Ques. Python While Loops?

- With the while loop we can execute a set of statements as long as a condition is true.

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

output:-

```
1
2
3
4
5
```

- **The break Statement:-** With the 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
```

Output:-

```
1
2
3
```

- **The continue Statement:-** With the continue statement we can stop the current iteration, and continue with the next.

```
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```

Output:-

```
1
2
4
5
6
```