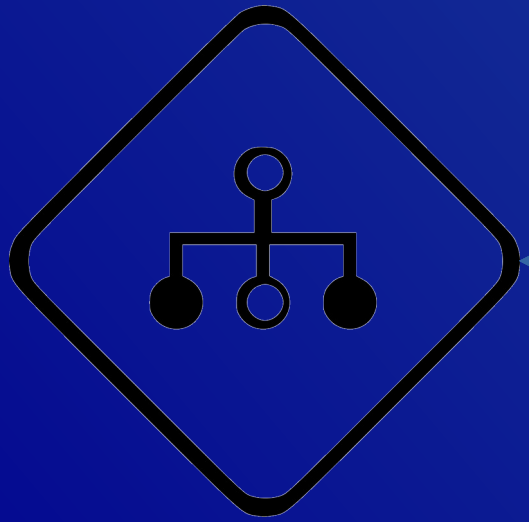
Boolean values

All value which are neither empty nor zero are **true**.

Empty collections & zero are False

Condition statements

It is used when you want to execute some code based on some condition.



1. If statement
2. If else statement
3. else If ladder statement
4. Nested if statements

if statement

It is used when you want to execute some code when condition is true.

If condition:
    true_statements

If n%2==0:
    print(n,“ is even”)

if else statement

It is used when you want to execute some code when condition is true or when condition is false.

```
If  condition:
    true_statements
else:
    false_statements
```

```
If  n%2==0:
    print(n,“ is even”)
else:
    print(n,“ is odd”)
```

else if statement

This is multi condition statement, when you want to check multiple condition on a single component

```
If  condition:
        true_statements_1
elIf  condition:
        true_statements_2
elif  condition:
        true_statements_3
…

else:
        false_statements
```

```
If  day==1:
        print(“Monday”)
elIf  day==2:
        print(“Tuesday”)
elif  day==3:
        print(“Wednesday”)
…

else:
        print(“Invalid input”)
```

Nested if statement

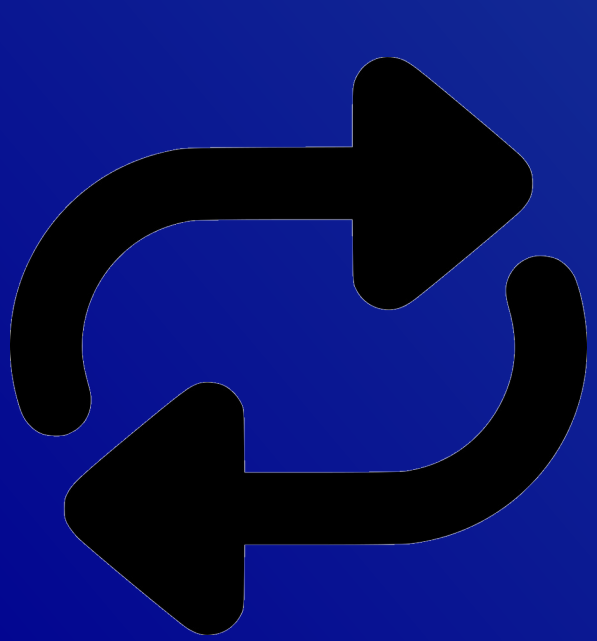
If statements in another if statement is known as nested statement.

```
If  condition:
    if condition2:
        true_part
else:
    false_statements
```

```
If  n%5==0:
    if n%2==0:
        print("Lucky number")
else:
    print("Better luck next time")
```

Iteration statements

It is used when you want to repeat execute some code based on some condition.



1. for
2. while
3. Nested loop statements

for loop

It is used when you know how many times you want to execute the statements.

```
for i in range(start,end,step):
    statements
```

```
for i in range(0,10,2):
    print(i, “is even”)
```

```
for i in collection:
    statements
```

```
for i in list:
    print(i, “is element”)
```

while loop

When you want to execute code only when condition & you don’t know how many times.

```
while condition:
    statements
```

```
while I<10:
    print(i)
    i+=1
```

else cause with loops

else cause is help full when you want to check condition on whole collection & make only one else part.

```
List = [2, 4, 5, 6, 6, 2, 8]
for i in List:
        if i %2!=0:
                break
else:
        print(“list contains all even..”)
```

pass statement

pass statements used when you want to leave empty block.

```
if n%3==0:
    pass
else:
    print(“welcome”)
```

break statement

To exit from loop or block

```
data = [1,4,3,8,1]
for i in data:
        if i%3==0:
                break
        print(“number ”,i)

print(“Final statement”)
```

continue statement

Used to skip an iteration of loop

```
data = [1,4,3,8,1]
for i in data:
    if i%3==0:
        continue
    print("number ",i)

print("Final statement")
```

return statement

To exit from a function, but it could return some values to where it has been called.

```
def code():
    print(“Hello world”)
    return 10
```

exit statement

exit from program

Statements
exit()
statements