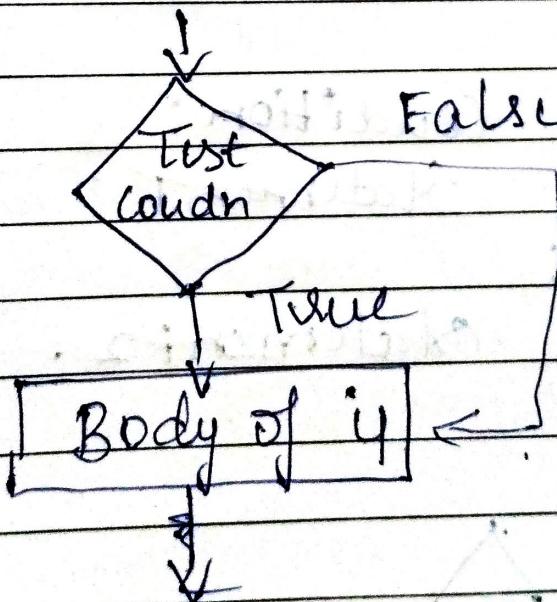


If statement

- > In Control statements, The If statement is the simplest form. It takes Condition and evaluates to either True or False.
- > If the Condition is True, then the True block of Code will be executed and if the Condition is False then the block of code is skipped.

Syntax:

```
if condition:
    Statement 1
    Statement 2
    Statement 3
```



Example write a program to calculate
the square of a number greater
than 5?

number = 6

if number > 5:

 print(number * number)

If - else Statement

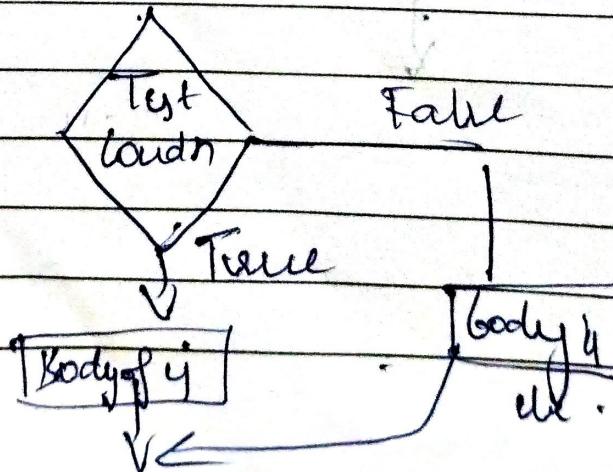
The If - else statement checks the condition and executes the block of code when the condition is True and if the condition is False it will execute the else block of code.

Syntax

If condition:
Statement 1

else:

Statement 2.



Ex

write a program to check the valid password.

Ex

```
⇒ password = input("Enter password")
if password == "Bdmg":
    print ("Valid")
else:
    print ("Invalid")
```

3) If - elif - else

In python the if - elif - else condition statement has an elif block to chain multiple conditions one after another.

> useful to check multiple conditions.

Syntax

```
If condition 1:
    Statement 1
```

```
elif condition 2:
    Statement 2
```

```
elif condition 3:
    Statement 3
```

```
else:
    Statement
```

Date _____ Page _____

Write a program to student result analysis

```
n = int(input("Enter a number"))
if n > 90 and n <= 100:
    print("A")
elif n > 80:
    print("B")
elif n > 70:
    print("C")
elif n > 50:
    print("D")
elif n < 35:
    print("Fail")
else:
    print("Invalid number")
```

Nested if else statement

In python the nested if - else Statement is an if statement inside another if - else Statement. It is allowed for python to put any number of if statement in another if statement.

Syntax

If condition:
If condition2:

Statement of outer loop

else:

Statement of inner

else:

Statement to outer

else:

Statement outside if block

Ex Find a great number of two numbers.

```
n = int(input("Enter first number"))
n1 = int(input("Enter second number"))
if num1 == num2:
    print(num1, "and", num2, "are equal")
else:
```

print("num1 is greater")

else:

print("num1 is smaller")

Loops

Loops are used to repeat a block of code.

Suppose if you want to print 100 messages in one shot, you can use for loop.

2 types

- > for loop
- > while loop.

for loop

For loop is used to iterate over sequences such as lists, tuples, string etc.

Syntax

For val in sequence:
 # statements (n)

Here, val occurs each item of sequence on each

Iteration

The loop continues until we reach the last item in sequence.

Ex loop with string

```
name = "abc"
for i in name:
    print(i)
```

Output:

Note

Integer obj cannot be iterable
because of position

enumerate → displays characters
along with the position

for i in enumerate(name):

(0, a)
(1, b)
(2, c)

print(i)

5

→ fruit = ["apple", "mango"]

for i in fruit

print(i)

apple

mango

2) For in range

range (start value, stop value, step size)

name = abc

for i in range(0, 3, 1):

print(i, name[i]).

With len function

for i in range(0, len(name), 1):

print(i, name[i])

Factorial

```
n = int(input())
fact = 1
for i in range(n, 0, -1)
    fact = fact * i
print(fact)
```

want count total number of
values, num of user number of
add.

```
c = int(input())
n = int(input())
mc = 0
e = 0
o = 0
for i in range(c, n+1, 1):
    mc = mc + 1
    if n % 2 == 0:
        e = e + 1
    else:
```

 mc

 e = e + 1

```
print(mc)
print(e)
print(o)
```