match statement

A **match** statement takes an expression and compares its value to successive patterns given as one or more case blocks.
Match statement is introduced in python 3.10 version

Syntax:

```
match(expression):
        case pattern:
                statement-1
        case pattern:
                statement-2
        case pattern:
                statement-3
        case _:
                statement-4
```

Example:

Write a program to input number from 1 to 10 and print roman format

```
num=int(input("Enter any number (1-10)"))
match(num):
   case 1:
      print("I")
   case 2:
      print("II")
   case 3:
      print("III")
   case 4:
      print("IV")
   case 5:
        print("V")
    case 6:
        print("VI")
    case 7:
        print("VII")
    case 8:
        print("VIII")
    case 9:
        print("IX")
    case 10:
        print("X")
    case _:
        print("number must be between 1-10")
```

Output:

```
Enter any number (1-10)12
number must be between 1-10
```

Example:

```
# Developing calculator

num1=int(input("Enter First Number "))
num2=int(input("Enter Second Number "))
operator=input("Enter Operator  (+,-,*,/)")
match(operator):
    case '+':
        print(f'sum of {num1} and {num2} is {num1+num2}')
    case '-':
        print(f'diff of {num1} and {num2} is {num1-num2}')
    case '*':
        print(f'prod of {num1} and {num2} is {num1*num2}')
    case '/':
        print(f'div of {num1} and {num2} is {num1/num2}')
    case _:
        print("invalid operator")
```

Output:

```
Enter First Number 10
Enter Second Number 5
Enter Operator  (+,-,*,/)*
prod of 10 and 5 is 50

Enter First Number 5
Enter Second Number 2
Enter Operator  (+,-,*,/)/
div of 5 and 2 is 2.5

Enter First Number 10
Enter Second Number 20
Enter Operator  (+,-,*,/)//
invalid operator
```

Example:

```
accountno=101
cname="naresh"
balance=50000
print("1.Deposit ")
print("2. Withdraw")
print("3. Balance ")
opt=int(input("Enter your option "))
match(opt):
    case 1:
        amt=float(input("Enter Amount "))
        balance=balance+amt
        print(f'Balance available {balance}')
    case 2:
        amt=float(input("Enter Amount "))
        if amt<balance:
            balance=balance-amt
            print(f'Balance available {balance}')
        else:
            print("Insuff balance")
    case 3:
        print(f'Balance available {balance}')
    case _:
            print("invalid option")
```

Output:

```
1.Deposit
2. Withdraw
3. Balance
Enter your option 1
Enter Amount 5000
Balance available 55000.0

1.Deposit
2. Withdraw
3. Balance
Enter your option 2
Enter Amount 10000
Balance available 40000.0
```

Looping Control Statements

Looping control statements are used to repeat one or more than one statement number of times or until given condition.

Python support 2 looping control statements

1. while
2. for

while loop

“while” keyword represents while loop.
“while” loop is used to repeat one or more than one statement until given condition is True.

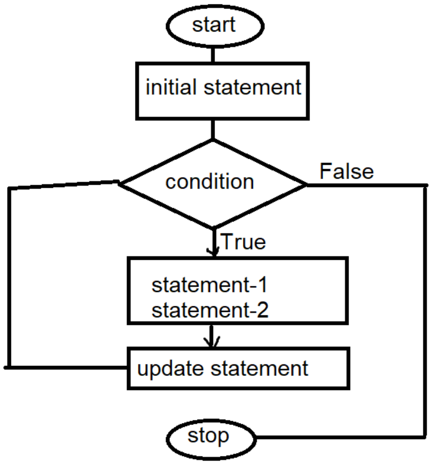
Syntax-1:	**Syntax-2:**

while <condition>:
   statement-1
   statement-2
statement-3
 	
PVM repeat statement-1,statement-2 until given condition is True.
	while <condition>:
	   statement-1
	   statement-2
	else:
	   statement-3
	   statement-4
	statement-5

While loop required 3 statements

1. initial statement
2. condition/Boolean expression
3. update statement

- Initial value of condition is called initial statement
- Condition is a Boolean expression which defines how many times while loop has to be repeated.
- Update statement, which updates condition.



Example:

# Write a program to print your name 5 times

```
c=0
while c<5:
    print("NIT")
    c=c+1
```

Output:

```
NIT
NIT
NIT
NIT
NIT
```

Example:

```
# Find Output
n=5
while n<=1:
    print(n)
    n=n+1
```

```
# Find Output
n=1
while n<=5:
   print(n);
   n=n+1
```

```
# Find Output

while False:
   print("Hello")
```

Output:

```
1
2
3
4
5
```