

# Python while Loop

The while loop in Python is used to iterate over a block of code as long as the test expression (condition) is true.

## Syntax:

```
while test_expression:
```

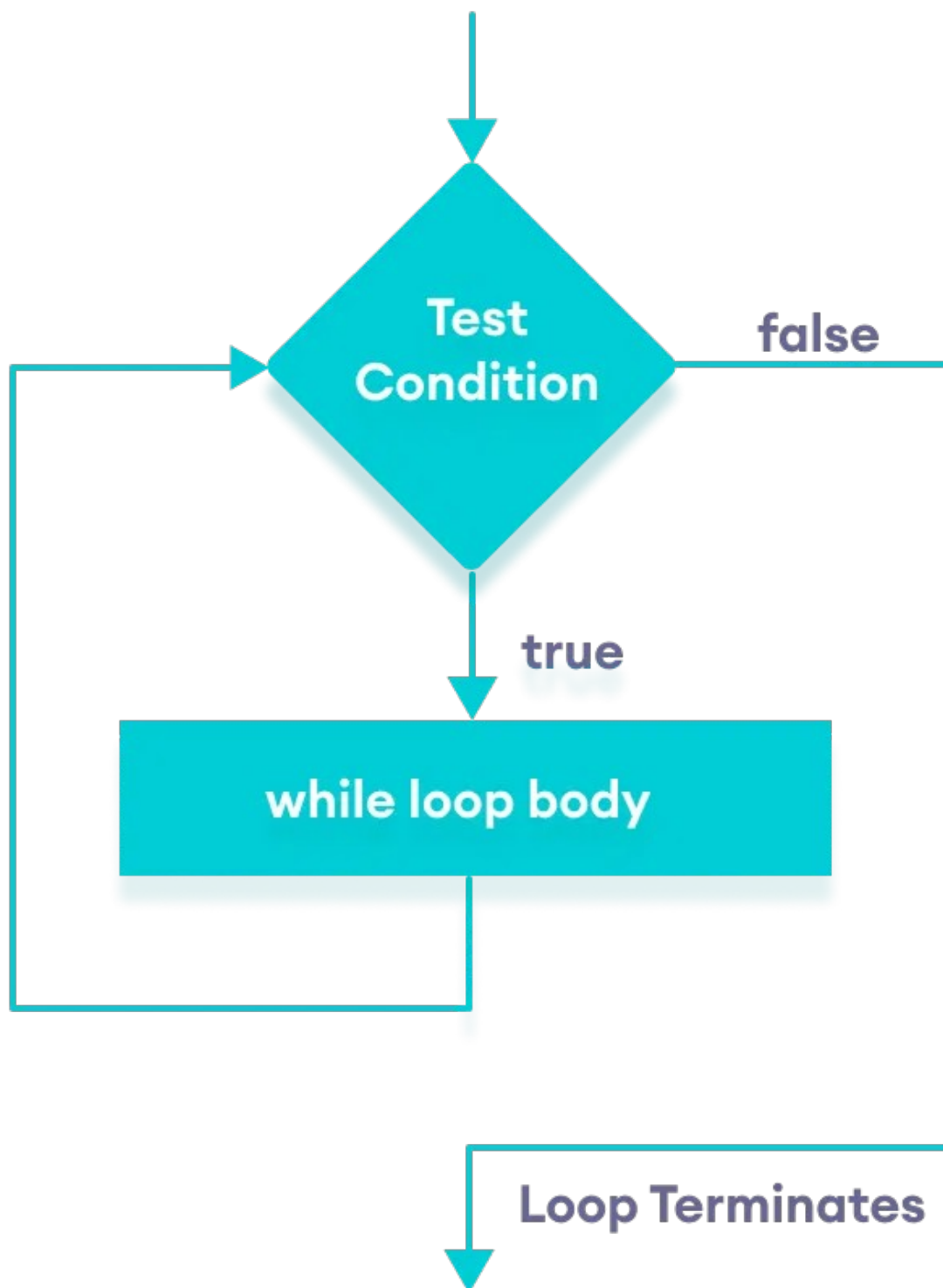
```
    Body of while
```

The body of the loop is entered only if the test\_expression evaluates to True.

After one iteration, the test expression is checked again.

This process continues until the test\_expression evaluates to False.

# Flow Chart



## Example

```
#Find product of all numbers present in a list
```

```
lst = [10, 20, 30, 40, 60]
```

```
product = 1
```

```
index = 0
```

```
while index < len(lst):  
    product *= lst[index]  
    index += 1
```

```
print("Product is: {}".format(product))
```

```
Product is: 14400000
```

## while Loop with else

Same as that of for loop, we can have an optional else block with while loop as well.

The else part is executed if the condition in the while loop evaluates to False. The while loop can be terminated with a break statement.

In such case, the else part is ignored. Hence, a while loop's else part runs if no break occurs and the condition is false.

```
numbers = [1, 2, 3,4,5]
```

```
#iterating over the list
```

```
index = 0
```

```
while index < len(numbers):  
    print(numbers[index])  
    index += 1
```

```
else:  
    print("no item left in the list")
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
no item left in the list
```

# Python Program to check given number is Prime number or not

```
num = int(input("Enter a number: "))           #convert string to int

isDivisible = False;

i=2;
while i < num:
    if num % i == 0:
        isDivisible = True;
        print("{} is divisible by {}".format(num,i) )
        i += 1;

if isDivisible:
    print("{} is NOT a Prime number".format(num))
else:
    print("{} is a Prime number".format(num))

Enter a number: 33
33 is divisible by 3
33 is divisible by 11
33 is NOT a Prime number
```