

# CLASS 3

# WHILE LOOP



# Control Flow: While loops

```
while <condition>:  
    <expression>  
    <expression>  
    .....
```

```
n=0  
while n < 5:  
    print(n)  
    n = n + 1
```

```
0  
1  
2  
3  
4
```

- **Evaluates to a Boolean** (Checks whether the statement in the condition is true or false)
- if the Boolean is **True**, do all the steps inside the while **code block** (Executes code block)
- Check **again for Boolean to be true**
- **Repeat until <code is false>**

# Control Flow: while loop exercises

## 1. Print integers from 0 to 10 (both inclusive)

```
n = 0
while n <= 10:
    print(n)
    n = n+1
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10

# Control Flow: while loop exercises

## 2. Print even integers from 3 to 21.

```
n = 3
while n < 21:
    if n%2 == 0:
        print(n)
    n = n+1
```

4  
6  
8  
10  
12  
14  
16  
18  
20

What will this code do?



```
n = 3
while n < 21:
    if n%2 == 0:
        print(n)
    n = n+1
```

# Control Flow: while loop exercises

## 3. Print prime numbers between 900 to 1000

```
In [1]: n = 900
while (n<=1000):
    if n > 1:
        i = 2
        while i < n:
            if (n%i == 0):
                break
            i = i + 1
        else:
            print(n)
        n = n+1
```



Step 1:

Checking whether “n” is less than or equal to 1000 or not



Step 2:

If  $n > 1$ , initialize a new value “i”



Step 3:

3.1. For every value of “ $i < n$ ”, check whether ‘i’ can divide ‘n’ or not.

3.2. if ‘i’ can divide ‘n’ then ‘n’

has other factors, so n is not a prime number.

3.3. Perform for all ‘i’ less than n.

907

911

919

929

937

941

947

953

967

971

977

983

991

997



Step 4: If ‘break’ statement do not get executed, then it’s a prime number print it

# FOR LOOP

# Control Flow: for loop

```
for variable in range(some_num):  
    <expression>  
    <expression>  
    .....
```

**Example:**  
**for n in range(5):**  
 **print(n)**

**Output: 0**

**1**  
**2**  
**3**  
**4**

- for loop takes the smallest value
- Increments itself by 1
- It understands the Boolean condition without explicitly mentioning

**Example with while:**  
**n=0**  
**while n < 5:**  
 **print(n)**  
 **n = n + 1**



# Control flow: Specifying range

- `range(start, stop, step)`
- Default values are `start = 0` and `step = 1`
- **Example 1: `range(10)` is equivalent to `range(0, 10, 1)`**
- Represents values 0, 1, 2, 3, 4, 5, 6, 7, 9.
- Loop until value is `stop-1`
- **See the output for `range(5,15,2)`**

# Control flow: Exercises

## 1. Print the sum of numbers from 7 to 20 (both inclusive)

```
mysum = 0
for i in range(7,21):
    mysum = mysum + i
# or
# mysum += 1
print(mysum)
```

189



Remember the  
Assignment Operator  
We discussed

## 2. Print the sum of odd numbers from 7 to 20 (both inclusive)

```
mysum = 0
for i in range(7,20,2):
    mysum += i
print(mysum)
```

91



Remember the  
Assignment Operator  
We discussed

# for vs while loop

## for

**Know number** of iterations

Can end early via break

**Uses a counter**

Can **rewrite a for loop**  
**using a while loop**

## while

**Unbounded number** of iterations

Can end early via break

Can use a counter but must  
**initialize before loop and**  
**increment it inside loop**

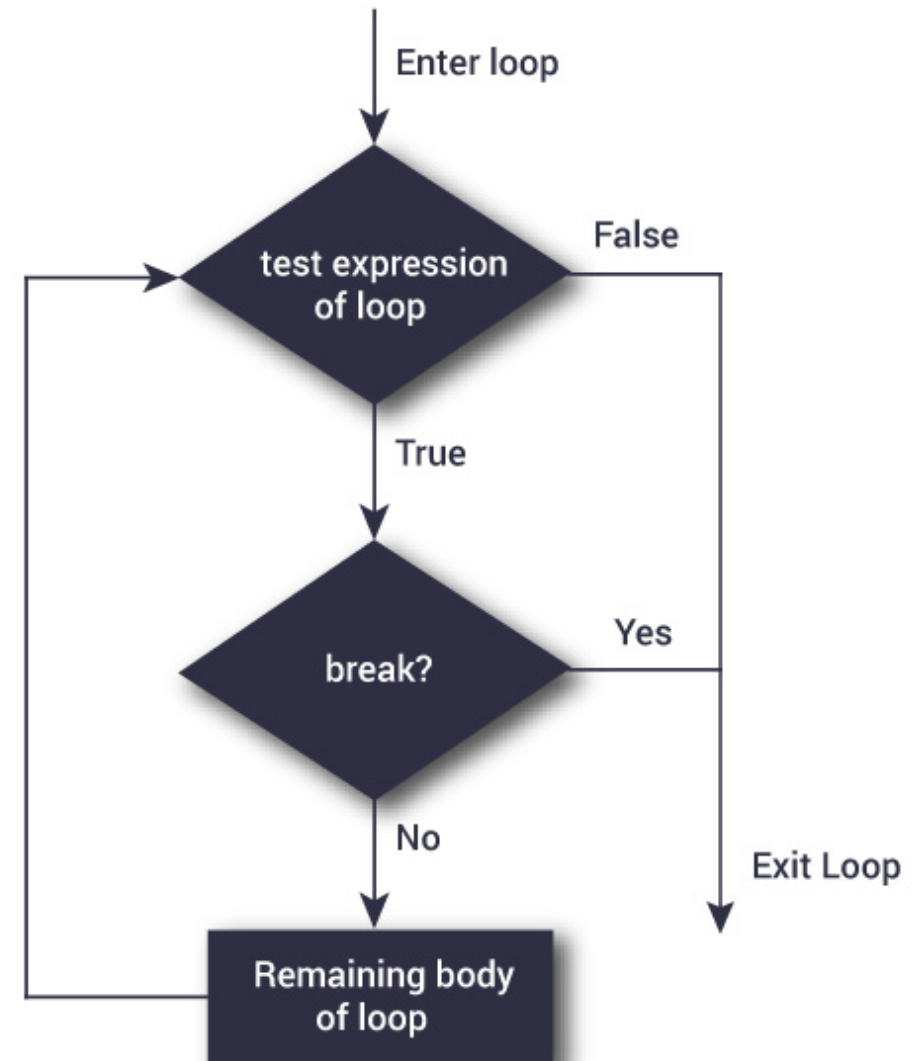
May not be able to rewrite a  
while loop using a for loop

# **“BREAK” and “CONTINUE” STATEMENTS**



# Control Flow: “Break” Statement

- **Immediately exists** whatever loop it is in.
- **Skips remaining expressions** in code block

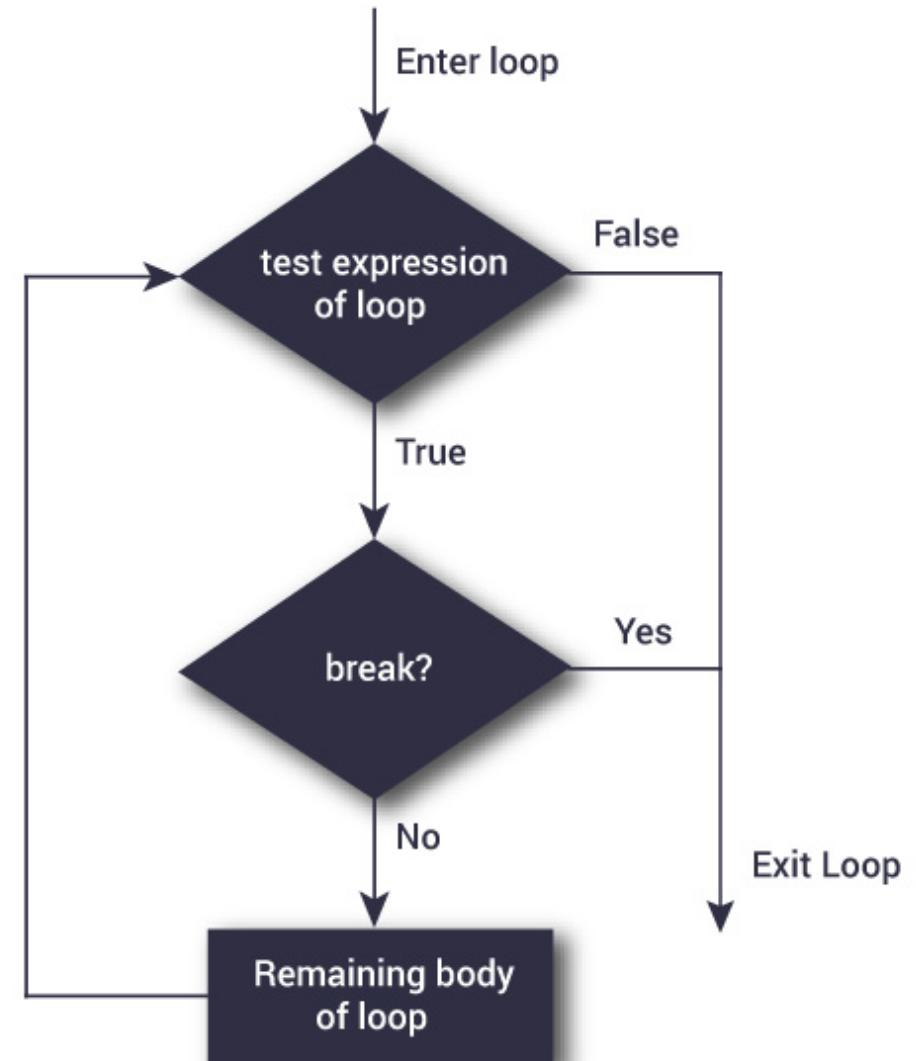


# Control Flow: “Break” Statement

- Try this



```
x = range(1,5)
for val in x:
    if val == 3:
        break
    else:
        print(val)
```



# Control Flow: “Break” Statement

- In nested loops, “Break exits only the inner most loop ”

```
while <condition_1>:  
    while <condition_2>:  
        expression_a  
        break  
    <expression_b>  
    <expression_c>  
<expression_c>
```

```
n = 900  
while (n<=1000):  
    if n > 1:  
        i = 2  
        while i < n:  
            if (n%i == 0):  
                break  
            i = i + 1  
        else:  
            print(n)  
        n = n+1
```



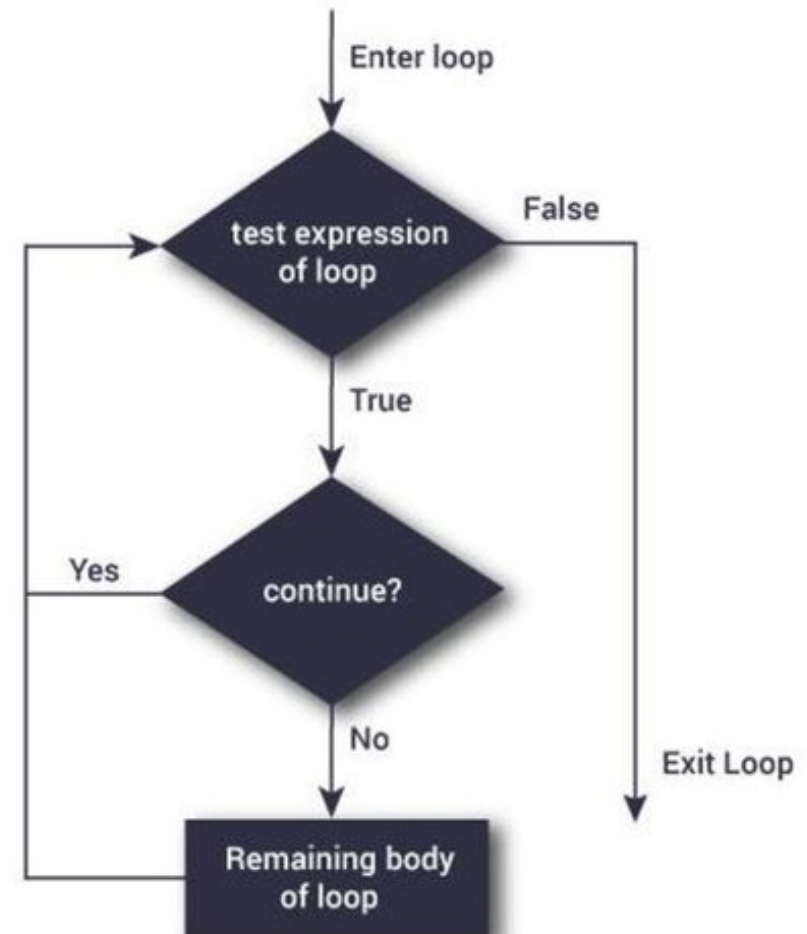
Step 4: If ‘*break*’ statement do not get executed, then it’s a prime number print it



# Control Flow: “Break” Statement

```
while (test Expression)
{
    // codes
    if (condition for continue)
    {
        continue;
    }
    // codes
}
```

```
for (init, condition, update)
{
    // codes
    if (condition for continue)
    {
        continue;
    }
    // codes
}
```



# Practice Exercises: HW

1. write a program that generates the second, third, and fourth powers of a list of whole numbers from 1 to n where n is input by the user.
2. Write a python program to print the square of all numbers from 0 to 10
3. Write a python program to find the sum of all even numbers from 0 to 10
4. Write a python program to read three numbers (a,b,c) and check how many numbers between 'a' and 'b' are divisible by 'c'
5. Write a python program to get the following output  
1-----9  
2-----8 Similar to i ----- n-i
6. Write a python program to print the factorial of a given number
7. Write a python program to print the first 10 numbers Fibonacci series

# LOOPS for STRINGS



# String: for loop

- **Syntax for range** is a way to iterate over numbers.
- But in python, a for loop variable can iterate over any set of values like characters, strings also not just numbers.



# String: for loop

- General way of working with strings in any other language
- **Example:** Check whether vowels “i” or “u” are in a given word ?

```
s = 'abcdefghi'
for index in range(len(s)):
    if (s[index] == 'i') or (s[index] == 'u'):
        print('there is an i or u')
```

there is an i or u

# String: for loop

- Python makes it easier

```
s = 'abcdefghi'
for char in s:
    if char == 'i' or char == 'u':
        print('There is an i or u')
```



Python Style



General Style

```
s = 'abcdefghi'
for index in range(len(s)):
    if (s[index] == 'i') or (s[index] == 'u'):
        print('there is an i or u')
```

there is an i or u

# String: for loop: Exercise

1. Print the alphabet in the string. Take any string.
  - Example: My name is vidya

```
my_string = " My name is vidya "  
  
for alphabet in my_string:  
    print(alphabet)
```

M  
y  
  
n  
a  
m  
e  
  
i  
s  
  
v  
i  
d  
y  
a

# String: for loop: Exercise

2. Print vowels in a string. Take any string.

- Example: Education

```
my_string = " Education "  
  
vowels = ''  
for alphabet in my_string:  
    if alphabet in 'aeiou':  
        vowels += ' ' + alphabet  
print(vowels)
```

u a i o



# String: for loop: Exercise

3. Print words in a sentence.

Example: “Is Python simpler than R ?”

```
sentence = "Is Python simpler than R ?"  
for word in sentence.split():  
    print(word)
```

Is  
Python  
simpler  
than  
R  
?

# Practice Exercises: HW

1. Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.

Sample String : 'w3resource'

Expected Result : 'w3ce'

Sample String : 'w3'

Expected Result : 'w3w3'

Sample String : ' w'

Expected Result : Empty String

2. Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.

Sample String : 'abc', 'xyz'

Expected Result : 'xyc abz'

3. Write a Python function that takes 3 words and returns the length of the longest one.

4. Write a Python program to change a given string to a new string where the first and last chars have been exchanged

5. Write a Python program to remove the characters which have odd index values of a given string.