



Example - `age = int(input("enter your age: - "))`
`if (age > 18):`
`print("major")`

Question एक Program बनाए जिसमें एक number input करे और check करे की वह Number Positive, negative या शून्य है।

`n = int(input("enter a number :- "))`
`if (n > 0):`
`print("positive")`

`if (n < 0):`
`print("negative")`

`if (n == 0):`
`print("zero")`

• If Else Statement - ^{अ की} condition true होने पर इसका block execute हो जाता है।
 condition गलत होने पर else का statement block execute होता है।

syntax :- `if <condition> :`
`<statement block>`

`else :`
`<statement block>`

```
Example - age = int(input("enter your age:-"))  
if (age >= 18):  
    print("major")
```

```
else:  
    print("minor")
```

Question

```
a = int(input("enter first number:-"))  
b = int(input("enter second number:-"))  
if (a > b):  
    print("a is large")  
else:  
    print("b is large")
```

Question एक Program बनार जिसमें एक Number input करे और check करे की वह number odd है या even.

```
n = int(input("enter any number:-"))  
if (n % 2 == 0):  
    print("even")  
else:
```

```
    print("odd")
```

Question एक Program लिखे जिसमें एक year input करे और बताए की वो year leap है या नहीं।

```
n = int(input("enter year:-"))  
if (n % 4 == 0 or n % 400 == 0):  
    print("leap year")  
else: print("not a leap year")
```



• If elif statement - Python में `if` `elif` का use एक से अधिक conditions को store करने के लिए किया जाता है। यदि `if` statement की condition सही होती है तो उसके अन्दर का code Run होता है। Condition गलत होने पर `elif` की condition एक के बाद एक check की जाती है। यदि कोई भी `elif` की condition true होती है तो उसके अन्दर का code Run होता है। यदि सभी condition गलत होती है तो `else` के अन्दर लिखा code execute होता है।

syntax :-

```
if <condition 1>:
```

```
<statement block 1>
```

```
elif <condition 2>:
```

```
<statement block 2>
```

```
elif <condition 3>:
```

```
<statement block 3>
```

```
..
```

```
..
```

```
else:
```

```
<statement block n>
```

Example - `n = int(input("enter any number"))`

```
if (n > 0):
```

```
    print("positive")
```

```
elif (n < 0):
```

```
    print("negative")
```

```
else:
```

```
    print("zero")
```



Question एक Program लिखें जिसमें एक student के marks input करें और उसका grade बताएं अगर marks 90 से ज्यादा हों तो grade A अगर 80 से 90 के बीच हों तो grade B 70-80 के बीच हों तो grade C 60-70 हों तो grade D और अगर 60 से कम हों तो grade F है।

```
n = int(input("enter marks"))
```

```
if (n > 90):
```

```
    print("grade A")
```

```
elif (n > 80):
```

```
    print("grade B")
```

```
elif (n > 70):
```

```
    print("grade C")
```

```
elif (n > 60):
```

```
    print("grade D")
```

```
else:
```

```
    print("grade F")
```

Question एक Program बनाएं जिसमें एक cricket खिलाड़ी के द्वारा बनाए गए score को input करें। और check करें की वह half century, century, double century, triple century है।

```
n = int(input("enter runs scored by the cricket player"))
```

```
if (n >= 300):
```

```
    print("triple century")
```

```
elif (n >= 200):
```

```
    print("double century")
```

```
elif (n >= 100):
```

```
    print("century")
```

```
elif (n >= 50):
```

```
    print("half century")
```

* • Nested if statement - किसी if statement के अन्दर अन्य if statement nested if statement कहलाती है। हम nested if statement का use तब करते हैं जब दो या दो से अधिक condition को एक साथ check किया जाता है। Nested if statement को निम्न प्रकार लिखा जा सकता है -

syntax :-

```

if < condition > :
    if < condition > :
        < statement block >
    else :
        < statement block >
else :
    if < condition > :
        < statement block >
    else :
        < statement block >
    
```

Example -

```

a = int(input("enter number first"))
b = int(input("enter number second"))
c = int(input("enter number third"))
if (a > b):
    if (a > c):
        print("a is large")
    else:
        print("c is large")
else:
    if (b > c):
        print("b is large")
    else:
        print("c is large")
    
```



- **Complex condition** - अब तक के सभी `if` statement में हम केवल एक ही condition को check कर रहे थे, जबकि ऐसी स्थिति भी आ सकती है जब हमें किसी कार्य को करने के लिए दो या दो से अधिक conditions को एक साथ check करने की आवश्यकता है। **Complex condition** को दो तरीकों से लिखा जा सकता है।

- 1 **Logical operator and** के साथ - `if` statement के अंदर सभी condition true होने पर ही इसका block statement execute होता है। एक भी condition गलत होने पर else part execute होता है -

```
a = int(input("enter number first"))
b = int(input("enter number second"))
c = int(input("enter number third"))
```

```
if (a > b and a > c):
    print("a is large")
elif (b > a and b > c):
    print("b is large")
else:
    print("c is large")
```

- 2 **Logical operator or** के साथ - `if` statement में लिखी गई condition में से एक condition true होने पर भी `if` statement का code run होता है। सभी condition गलत होने पर ही else part का code run होता है।



```
a = int(input("enter number"))  
if (a == 1 or a == 0):  
    print("binary number")  
else:  
    print("not a binary number")
```

Question एक Program बनाये जिसमें कोई भी alphabet लिखने पर पता चले की वह vowel है या consonant

```
a = input("enter any alphabet")  
if (a == "a" or a == "e" or a == "i" or  
    a == "o" or a == "u"):  
    print("Vowel")  
else:  
    print("consonant")
```

(ii) Looping statement - इसे repetition statement के रूप में जाना जाता है। Looping statement एक ऐसी command है जो दी गई condition के आधार पर पूरे program या उसके एक हिस्से को दोहराती है। Python दो प्रकार के looping statement को support करता है।

- (1) While loop
- (2) For loop

1. While loop - इसे conditional loop भी कहते हैं।
 क्योंकि यह statement को तब तक दोहराती है जब तक की गई condition पूरा रहेगी।

syntax :
`while < condition > :
 body of loop`

Ex-
`a = 1
 while (a <= 5):
 print ("Python")
 a = a + 1`

Ques 1 Counting लिखने के लिए -
`a = 1
 while (a <= 10):
 print (a)
 a = a + 1`

Ques 2 के Table के लिए -
`a = 2
 while (a <= 20):
 print (a)
 a = a + 2`

OR

`a = 1
 while (a <= 10):
 print (a * 2)
 a = a + 1`



कौई भी table लिखने के लिए -

```
n = int(input("enter any number"))  
a = 1
```

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

एक Program लिखें जिसमें 1 से 100 के बीच
even संख्याएं Print कराये।

```
a = 2  
while (a ≤ 100):  
    print(a)  
    a = a + 2
```

Odd के लिए -

```
a = 1  
while (a ≤ 100):  
    print(a)  
    a = a + 2
```

प्रथम 10 संख्याओं को उल्टे क्रम में Print कराने के
लिए Program लिखें।

```
a = 10  
while (a >= 1):  
    print(a)  
    a = a - 1
```

Question पहले 15 Natural Number के अंकों का जोड़ Print करने का Program कराये।

a = 1

s = 0

while (a <= 15):

s = s + a

a = a + 1

print (s)

} direct जोड़ के लिए
बाहर करेंगे Print

Question एक Program बनाये जिसमें Circle की radius input करे उसके क्षेत्रफल को calculate करके Print करे। Program user की इच्छानुसार repeat होना चाहिए।

ch = "y"

while (ch == "y"):

r = int(input("enter radius of circle :- "))

area = 3.14 * r * r

print("area of circle :- ", area)

ch = input("do you want to repeat y/n :- ")

Question - कोई भी table लिखने पर -

$$12 * 1 = 12$$

$$12 * 2 = 24$$

इसी तरीके में

n = int(input("enter number :- "))

a = 1

while (a <= 10):

print(n, "*", a, "=", a * n)

a = a + 1



2. For loop - For loop एक counting loop है यह पहले से निर्धारित संख्या तक loop को दोहराता है यह एक बहुमुखी statement है। इसके कई रूप हो सकते हैं। इसे निम्न प्रकार लिखा जा सकता है।

Syntax-I

```
for <control variable> in range (<numbers>):  
    body of loop
```

Example :- for i in range (11):
 print (i)

```
for i in range (1, 11):  
    print (i*2)
```

Syntax II :-

```
for <control variable> in range (start, stop, step):  
    body of loop
```

Ex - for i in range (1, 11):
 print (i)

Ex - for t in range (2, 21, 2):
 print (t)

Ex - for t in range (5, 51, 5):
 print (t)

Question 50-100 के मध्य even and odd no print करे।
for e in range (50, 101, 2):
print (e)

for o in range (51, 101, 2):
print (o)

Question निम्न series को 15 terms तक print करने का
Program लिखे।

1, 4, 9, 16, 25 ----- 15 terms
for s in range (1, 16):
print (s*s)

Question एक Program बनाये जिसमे 1 से 30 तक की संख्याओं
का average print करे।

```
a = 1
s = 0
while (a <= 30):
    s = s + a
    a = a + 1
print ("average :-", s/30)
```

Question user से एक number input कराये और उसका factorial
Print करे।

```
n = int (input ("enter any number"))
a = 1
f = 1
while (a <= n):
    f = f * a
    a = a + 1
print ("fact :-", f)
```



Question User से एक संख्या input कराये और उस संख्या को reverse करने का Program लिखें।

```
n = int(input("enter number :-"))
```

```
g = 0
```

```
while (n != 0):
```

```
    m = n % 10
```

```
    g = g * 10 + m
```

```
    n = n // 10
```

```
print (g)
```

Question 30 से 70 के मध्य उन संख्याओं को Print करे जो 3 से विभाजित होती है।

```
I for i in range (30, 71, 3):
```

```
    print (i)
```

```
II a = 30
```

```
while (a <= 70):
```

```
    print (a)
```

```
    a = a + 3
```

• for loop में sequence data type का उपयोग -

for loop का use किसी भी sequence data type के element (जैसे - string, list) को access करने के लिए किया जा सकता है। इसमें <control variable>

पहले element से sequence को access करना शुरू करता है।

syntax -

```
for <control variable> in (<any sequence data>):  
    body of loop
```

Example - for a in "Python":
 print (a)

Question नीचे दी गई string में z की संख्या calculate करें और उसे print करें।

eizjhuzgzfztdzgjhgzdezfgzgh

a = "eizjhuzgzfztdzgjhgzdezfgzgh"

c = 0

for i in a:

if i == "z":

c = c + 1

print(c)

Question नीचे दी गई संख्याओं का square print करने का program बनाएँ।

a = [3, 5, 6, 4, 9]

for i in a:

print(i * i)

Question निम्न list में दी गई उम्र के आधार पर major or minor व्यक्तियों की संख्या गिने व print करने का program लिखें।

a = [3, 34, 18, 38, 22, 16, 2]

age = [3, 34, 18, 38, 22, 16, 2]

mj = 0

mn = 0

for a in age:

if (age >= 18):

mj = mj + 1

else:

mn = mn + 1

print("no of major", mj)

print("no of minor", mn)



Question एक किस गरु सं तक सभी संख्याओं का घन की गणना करे।

```
n = int(input("enter number :- "))
for i in range(1, n+1):
    print(i*i*i)
```

Question एक Program लिखे जिसमें उपयोगकर्ता चाहे वहां तक सं दर्ज कर सकता है और अंत में यह प्रोग्राम सभी दर्ज की गई सं का योग दिखाएगा।

```
ch = "y"
s = 0
while(ch == "y"):
    n = int(input("enter any number :- "))
    s = s+n
    ch = input("do you want to add any more number (y/n)")
print(s)
```

Question एक Program लिखे जो एक संख्या में अंको की कुल संख्या को गिनने के लिए एक while लूप का उपयोग करते हुए।

```
a = 678121345
c = 0
while(a != 0):
    a = a//10
    c = c+1
print(c)
```

* Jump statement - ये वे statement होते हैं जो looping statement के repetition की प्रक्रिया को बदलते हैं। Python में दो प्रकार के jump statement होते हैं।

- 1. Break statement
- 2. Continue statement

1. Break statement - Break statement loop repetition को समाप्त करता है और program के control को body के loop से बाहर लेकर आता है।

syntax 1 :-

```
while (<condition>):  
    statement 1  
    if condition:  
        break  
    statement 2  
    statement 3
```

syntax 2 :-

```
for <control variable> in range (<n>):  
    statement 1  
    if condition:  
        break  
    statement 2  
    statement 3
```



Question

दो संख्या को एक numerator के रूप में और दूसरी denominator के रूप में input करने का एक program लिखें। सं० को division के परिणाम की calculation करके print करें। Program को 10 संख्या के लिये इस प्रक्रिया को दोहराना चाहिए। user द्वारा denominator zero के रूप में input करने से program की repetition समाप्त होना चाहिए।

```
n = d = 1 = 0
```

```
a = 1
```

```
while (a <= 10):
```

```
n = int(input("enter numerator :- "))
```

```
d = int(input("enter denominator :- "))
```

```
if (d == 0):
```

```
break
```

```
m = n/d
```

```
print(m)
```

```
a = a + 1
```

```
print("program over")
```

Question

```
n = 1 = 1
```

```
a = 1
```

```
while (a <= 10):
```

```
n = int(input("enter any number :- "))
```

```
if (n == 0):
```

```
break
```

```
m = m * n
```

```
a = a + 1
```

```
print(m)
```

2. Continue statement - यह statement body के loop से किसी repetition को skip करने लिये किया जाता है।

Syntax 1 -

while (<condition>):

statement 1

if condition:

continue

statement 2

statement 3

Syntax 2 -

for <control variable> in range (<n>):

statement 1

if condition:

continue

statement 2

statement 3

Question 10 integer input करने के लिये program लिखें।

calculate करें की इसमें केवल positive integer का average print है। Negative integer को add नहीं किया जाना चाहिए। और zero input करने पर program समाप्त होना चाहिए।

S = 0

C = 0

for a in range (11):

n = int(input("enter number :- "))



```
if (n == 0):  
    break  
if (n < 0):  
    continue  
s = s + n  
c = c + 1  
average = s/c  
print("average")
```

```
m = 1  
for i in range(1, 11):  
    n = int(input("enter number :- "))  
    if (n == 0):  
        break  
    if (n < 0):  
        continue  
    m = m * n  
    print(m)
```

Q.11) एक loop बनाना जो 1-100 तक संख्या print करता है।
लेकिन जब loop 50 तक पहुँचता है तो वह program को break कर देता है।

```
for i in range(1, 101):  
    if (i == 50):  
        break  
    print(i)
```

Question एक loop बनार जो 1-20 तक संख्या print करता है।
लेकिन 15 को print नहीं करता।

```
for i in range (1, 21):  
    if (i == 15):  
        continue  
    print (i)
```

→ Loop else statement -

Python loop का else statement loop की समाप्ति के बाद
execute होता है। While loop में else statement तब execute
होता है जो condition false हो जाती है और for loop
में else statement तब execute होता है जब pre-defin
repetition complete हो जाता है जो break statement
loop को समाप्त करता है तब else part execute नहीं
होता। इसे निम्न प्रकार लिखा जा सकता है।

Syntax 1 :-

```
while <condition> :  
    statement 1  
    statement 2
```

```
else :  
    statement n
```

Syntax 2 :-

```
for <variable> in range <n> :  
    statement 1  
    statement 2
```

```
else :  
    statement n
```

```
Example: for i in range (5):  
    print (i)  
else:  
    print ("loop over")
```

Question एक circle की radius input करने के लिए program लिखे।
calculate कर उसके area को print करे। इस प्रक्रिया को तीन अलग अलग circle के लिये दोहराए। Radius को 0 enter करने पर program समाप्त होना चाहिए।
जब program असामान्य रूप से समाप्त हो जाता तो एक message print करे।

```
t = 0  
for i in range (3):  
    r = int (input ("enter radius :-"))  
    if (r == 0):  
        break  
    area = 3.14 * r * r  
    print (area)  
else:  
    print ("loop over")  
    t = 1  
    if (t == 0):  
        print ("loop terminated prematurely")
```

Question कुछ संख्या को बार बार input करके उनके योग को print करने के लिए program लिखे। user द्वारा कोई भी data enter ना करने पर program समाप्त हो जाता है।
जब user negative सं० input करता है program break हो जाता है।



```
ch = "y"
```

```
s = 0
```

```
while (ch == "y"):
```

```
n = int(input("enter any number :- "))
```

```
if (n < 0):
```

```
break
```

```
s = s + n
```

```
ch = ("enter any more data (Y/n) :- ")
```

```
else:
```

```
print("sum", s)
```

* **Nested loop** - जब एक loop के अन्दर दूसरा loop लिखा जाता है। nested loop कहलाता है। Python में दो looping statement दिए गए हैं for loop और while loop, इनमें से एक loop को external loop और दूसरे loop को internal loop के रूप में use किया जा सकता है।

Example - for i in range (4, 11):

```
for a in range (1, i+1):
```

```
print ("*", end = " ")
```

```
print ()
```

while loop से -

```
a = 1
```

```
while (a <= 10):
```

```
b = 1
```

```
while (b <= a):
```

```
print ("*", end = " ")
```

```
b = b + 1
```

```
a = a + 1
```

```
print ()
```



Question 25 - 100 के मध्य सभी Prime number print करने का Program लिखें।

```
for a in range (25, 101):  
    for b in range (2, a):  
        if (a % b == 0):  
            break
```

else:

print (a)

Question User से एक number input कराये और check करें कि वह Prime number है या नहीं।

```
a = int (input ("enter number"))
```

```
for b in range (2, a):
```

```
    if (a % b == 0):
```

```
        print ("not a prime number")
```

```
        break
```

else:

```
    print ("prime number")
```

Question दो variable a and b के data को आपस में बदलने के लिये एक Program लिखें।

```
a = 5
```

```
b = 7
```

```
c = a
```

```
a = b
```

```
b = c
```

```
print (a)
```

```
print (b)
```

OR

```
a = 5
```

```
b = 7
```

```
a, b = b, a
```

```
print (a)
```

```
print (b)
```

Question while statement का use करते हुए amount और rate of exchange दोनो में input करें और amount को Rupee में convert कर print करें | print का दोहराव user की पसंद के आधार पर है |

```
ch = "y"
while (ch == "y") :
    r = float(input("rate of exchange: -"))
    a = float(input("enter amount"))
    converted = r * a
    print(converted)
    ch = input("do you want to repeat")
```

Question

```
n = int(input("enter number"))
```

```
a = 0
```

```
b = 1
```

```
for i in range(0, n):
```

```
    print(a, end = " ")
```

```
    c = a + b
```

```
    a = b
```

```
    b = c
```

[Fibonacci Series]

पहली दो संख्या को जोड़कर तीसरी संख्या आती है।