



<https://apssdc.in>

APSSDC

Andhra Pradesh State Skill Development Corporation



APSSDC

Conditional Statements and Loops in Python

Day Objectives

- Conditional Statements
 - if
 - if-else
 - if-elif-else
- Loops in Python
 - For
 - While
- Strings
 - Declaring a String
 - Accessing the Elements from the String
 - String Methods

Conditional Statements

Controlling the flow of execution

Syntax

- if

```
if condition:  
    Block of Code
```

- if-else

```
if condition:  
    Block of Code  
else:  
    Block of Code
```

- if-elif-else

```
if condition:
    Block of Code
elif condition:
    Block of Code
else:
    Block of Code
```

- Biggest of 2 numbers

```
In [1]: ▶ 1 a = 5
          2 b = 15
          3
          4 if a > b:
          5     print(a, "is the biggest number")
          6 else:
          7     print(b, "is the biggest number")
```

15 is the biggest number

```
In [4]: ▶ 1 a = 5
          2 b = 5
          3
          4 if a > b:
          5     print(a, "is the biggest number")
          6 elif b > a:
          7     print(b, "is the biggest number")
          8 else:
          9     print(a, "Both numbers are equal")
```

5 Both numbers are equal

```
In [2]: ▶ 1 a = 5
          2 b = 15
          3
          4 if a > b:
          5     print(a, "is the biggest number")
          6 else:
          7     print(b, "is the biggest number")
```

```
File "<ipython-input-2-da1668ffd3b7>", line 7
    print(b, "is the biggest number")
      ^
```

IndentationError: expected an indented block

- Biggest of 3 numbers

```
In [5]: ▶ 1 a = 5
2 b = 15
3 c = 8
4
5 if a > b and a > c:
6     print(a, "is the biggest number")
7 elif b > c:
8     print(b, "is the biggest number")
9 else:
10    print(c, "is the biggest number")
```

15 is the biggest number

- check the input number is even or not

```
In [7]: ▶ 1 n = int(input("Enter a Number: "))
2
3 if n % 2 == 0:
4     print(n, "is the even number")
5 else:
6     print(n, "is not even number")
```

Enter a Number: 698567
698567 is not even number

Verify given year leap year or not

- It should be by 4 and rem as 0
- it shouldn't be divisible by 100
- divisible by 400

```
In [11]: ▶ 1 year = int(input("Enter a year: "))
2
3 if (year % 400 == 0) or (year % 4 == 0 and year % 100 != 0):
4     print(year, "is a leap year")
5 else:
6     print(year, "is not a leap year")
```

Enter a year: 1900
1900 is not a leap year

Calculator Application

- 2 input number
- Arithmetic operation

```
In [14]: ▶ 1 a = int(input("Enter a number: "))
2 b = int(input("Enter a number: "))
3 arthe = input("What Arthemetic I Need to Perform: ")
4
5
6 if arthe == '+':
7     print(a + b)
8 elif arthe == '-':
9     print(a - b)
10 elif arthe == '*':
11     print(a * b)
12 else:
13     print("Given arthemetic operation is invalid")
14
15 print(a+b)
16 print(a-b)
17 print(a*b)
```

```
Enter a number: 5
Enter a number: 5
What Arthemetic I Need to Perform: -
0
10
0
25
```

Loops in Python

execute the block of code for multiple times

- Initialization
- Condition
- inc/dec

Syntax

- For Loop -> How many time I need to Execute the loop

```
for IterativeVariable in GroupOfData:
    Block of Code
```

- While Loop -> Based on the condition

```
init
while condition:
    Block of Code
    inc/dec
```

GroupOfNumber

```
range(init, cond, inc/dec)
range(start, stop, step)
```

```
cond -> required arg
init -> 0
inc/dec -> +1
```

```
In [15]: ▶ 1 print(1)
           2 print(2)
           3 print(3)
           4 print(4)
```

```
1
2
3
4
```

Display 1 to 100 numbers to the users

1, >101, 1

```
In [17]: ▶ 1 for i in range(1, 101, 1):
           2     print(i, end = '\t')
```

```
1      2      3      4      5      6      7      8      9      10
11     12     13     14     15     16     17     18     19     20
21     22     23     24     25     26     27     28     29     30
31     32     33     34     35     36     37     38     39     40
41     42     43     44     45     46     47     48     49     50
51     52     53     54     55     56     57     58     59     60
61     62     63     64     65     66     67     68     69     70
71     72     73     74     75     76     77     78     79     80
81     82     83     84     85     86     87     88     89     90
91     92     93     94     95     96     97     98     99     100
```

```
In [18]: ▶ 1 for i in range(101):
           2     print(i, end = '\t')
```

```
0      1      2      3      4      5      6      7      8      9
10     11     12     13     14     15     16     17     18     19
20     21     22     23     24     25     26     27     28     29
30     31     32     33     34     35     36     37     38     39
40     41     42     43     44     45     46     47     48     49
50     51     52     53     54     55     56     57     58     59
60     61     62     63     64     65     66     67     68     69
70     71     72     73     74     75     76     77     78     79
80     81     82     83     84     85     86     87     88     89
90     91     92     93     94     95     96     97     98     99
100
```

```
In [19]: ▶ 1 for i in range(1, 101, 2):  
          2     print(i, end = '\t')
```

```
1      3      5      7      9      11     13     15     17     19  
21     23     25     27     29     31     33     35     37     39  
41     43     45     47     49     51     53     55     57     59  
61     63     65     67     69     71     73     75     77     79  
81     83     85     87     89     91     93     95     97     99
```

Even numbers between 1 to 1000

1. Get the Leap Year between the given range
 - Starting range ending range is given as input
2. Get the prime numbers between the given range
3. Get n number of Fibonacci numbers

While Loop

```
In [22]: ▶ 1 i = 1
           2 while i < 101:
           3     print(i, end = '\t')
           4     i += 1
```

```
1      2      3      4      5      6      7      8      9      10
11     12     13     14     15     16     17     18     19     20
21     22     23     24     25     26     27     28     29     30
31     32     33     34     35     36     37     38     39     40
41     42     43     44     45     46     47     48     49     50
51     52     53     54     55     56     57     58     59     60
61     62     63     64     65     66     67     68     69     70
71     72     73     74     75     76     77     78     79     80
81     82     83     84     85     86     87     88     89     90
91     92     93     94     95     96     97     98     99     100
```

- Break: To terminate the execution of the loop
- Continue: Skip the current execution of the loop

print the numbers between 1 to 100 whenever I reached 50 stop travelling

```
In [23]: ▶ 1 for travel in range(1, 101, 1):
           2     if travel == 50:
           3         break
           4     print(travel, end = '\t')
```

```
1      2      3      4      5      6      7      8      9      10
11     12     13     14     15     16     17     18     19     20
21     22     23     24     25     26     27     28     29     30
31     32     33     34     35     36     37     38     39     40
41     42     43     44     45     46     47     48     49
```

```
In [25]: ▶ 1 for travel in range(1, 101, 1):
           2     if travel == 50:
           3         break
           4     elif travel % 2 == 0:
           5         continue
           6     print(travel, end = '\t')
```

```
1      3      5      7      9      11     13     15     17     19
21     23     25     27     29     31     33     35     37     39
41     43     45     47     49
```



```
In [26]: ▶ 1 while True:
           2     inp = input()
           3     if inp == 'cancel':
           4         print("Loop Terminated")
           5         break
           6     print(inp)
```

```
646
646
kbjh
kbjh
nb
nb
vj
vj
nv
nv
inp
inp
mbds
mbds
ksbcc
ksbcc
cancel
Loop Terminated
```

Strings

it is group of data enclosed between "", ", "''''''", ""'''' are called strings in python

Properties of Strings

- It is immutable data type
- It is iterable

```
In [28]: ▶ 1 s = ""
           2 s2 = str()
```

```
In [29]: ▶ 1 s = 'I am a single line String'
2 s2 = "I'm also a single line string"
3 s3 = '''I am
4 a
5 multi
6 line
7 string'''
8
9 s3 = """I'm
10 a
11 multi
12 line
13 string"""
```

```
In [31]: ▶ 1 print(type(s), type(s3))

<class 'str'> <class 'str'>
```

```
In [32]: ▶ 1 for char in s3:
2     print(char, end = '\t')
```

I ' m

 a

 m u l t i

 l i n e

 s t r i n g

```
In [33]: ▶ 1 s = """A leap year (also known as an intercalary year or bissextile year)
2
3 For example, in the Gregorian calendar, each leap year has 366 days instead
```

```
In [34]: ▶ 1 print(s)
```

A leap year (also known as an intercalary year or bissextile year) is a calendar year that contains an additional day (or, in the case of a lunisolar calendar, a month) added to keep the calendar year synchronized with the astronomical year or seasonal year.[1] Because astronomical events and seasons do not repeat in a whole number of days, calendars that have a constant number of days in each year will unavoidably drift over time with respect to the event that the year is supposed to track, such as seasons. By inserting (called intercalating in technical terminology) an additional day or month into some years, the drift between a civilization's dating system and the physical properties of the solar system can be corrected. A year that is not a leap year is a common year.

For example, in the Gregorian calendar, each leap year has 366 days instead of 365, by extending February to 29 days rather than the common 28. These extra days occur in each year which is an integer multiple of 4 (except for years evenly divisible by 100, which are not leap years unless evenly divisible by 400). The leap year of 366 days has 52 weeks and two days, hence the year following a leap year will start later by two days of the week.

In [35]: ▶

```
1 for char in s:  
2     print(char, end = '\t')
```

```
A r o i y b y c e o a d n o o d t t e y r t r y a r u o n a o i n d n h s e i a a d t r t t e p r a . t e  
      w n i e a a n d a f l a h o e o h o e s . s m t s t n u a d a t r n r v r i e h h a o a s i d  
l ( n t y s a l r t d y t a r ) c a n a o [ e i s o m y a v a o i m s e a r s c B n  
e a e s r e a i h a r , k a r i t o r n 1 c n r a b s r e n o e w i f e p t e k s y g i  
a l a r a e ) n t i t ( e a e l z h m a ] a a a s e e , s t f a i d t e e i d , e n  
p s s c r x d h n i o l c a d e e s e e i o l s l n p w r a c l a w c v t s a i ( t  
o a t i a a s o r c u a d p n y d c r B t d d e h c t n d h l b o i t e h t s n c e  
y a l o i s r t , a n l m e d n a a y e r e o a o a h c u a l v t n e s o u o s a r  
e k n a r l a a s i e o d t a c w s l s e c o v s t l f l a o m y y u y e h t t u c n e l c  
a n r e a y c n l i e s n n h r h i t e a a n e e n e t n b s e n r o y p t h s r l a
```

l t t y t o n e d e l d t e p o a c r y i e s y
l G c e y 6 s 6 e r 9 h h 2 e o a h i l 4 f e i 1 a a n l l .
a e e) i r t a r e i a e r f r a e e s a e
F e r a a e 6 t 5 n u e e 8 x c c i n t o v s 0 r p l y e
t c r o r i n z t m p o n c a p a a
o , e l c a e , d a d r . t c h c t i (r e i 0 e e T
i h m a n m s f a i h p t s t r n r
r g e h r d a i r a c r u h e p e n b , y s d b h
n n i n a o s , t a t n a y e h y b e o y c .
i o n a d b n y y t o T a r y g l x y l l n e s i y e
g i n l n o i g n s r e s e d t t e o
e n r d l h y y g s h m h e i e e c e y e w o a v
c o a t m t b c o d i t t . h a m
x i a e a s o t a m e d i a s r e a h t r e i 4 l
i a l d d h e h e i n s c i s e c a a r m
a t a r a s f e F o r n o s a n r o p r d b i s v s 0 e
n l o d a e t v ' y t a e o m o A t o
m h n , p i x e a n e y a m f t s i y c l e i 0 a
g i y i y w i s s h l s l r l i n p e 3 n 3 t b 2 t t s e w n u v h e u n b) p

```

y     e     a     r           o     f           3     6
6     d     a     y     s     h     a     s
5     2           w     e     e     k     s     a     a
n     d           t     w     o           d     a     y
s     ,           h     e     n     c     e     t
h     e           y     e     a     r           f     o
l     l     o     w     i     n     g     a     r
l     e     a     p           y     e     a     r
w     i     l     l           s     t     a     r     t
l     a     t     e     r           b     y     t
w     o           d     a     y     s     o     f
t     h     e     w     e     e     k     .

```

```

In [38]: ▶ 1 for char in s:
           2     if char == 'a' or char == 'e' or char == 'i' or char == 'o' or char == 'u':
           3         print(char, end = ' ')

```

```

e a e a a o o a a i e a a e a o i e i e e a i a a e a e a a o a i a a i i o
a a o i e a e o a u i o a a e a a o a e o e e e a e a e a o i e i e a o o i
a e a o e a o a e a e a u e a o o i a e e a e a o o o e e a i a o e u e o a
a e a a a e a o a u e o a i e a e a i u a o i a i o e i e i e e o e e e a e
e a i u o e o a u a e a o i e i a e i e a a i i e i a e i o o a a i i o a a
o o i o o e e a e i e e e a i i i a i o a i e a e i a o e i e o e o a e a e
o e e e a a i o a e a e a i a o o e a o e a e i e e o i a a e a e a e a e a
a a i e a o e e i e u a o a a e a e o o e e e a a o u i e a e a i i a i e e
u i e o e e o e a e e i i i e i a e o e a e a u e e e i i i e e e a e a o a
a e e a o a e e e e a o o i a e a e a i a a e o a o e e e

```

```

In [39]: ▶ 1 for char in s:
           2     if char in 'aeiouAEIOU':
           3         print(char, end = ' ')

```

```

A e a e a a o o a a i e a a e a o i e i e e a i a a e a e a a o a i a a i i
o a a o i e a e o a u i o a a e a a o a e o e e e a e a e a o i e i e a o o
i a e a o e a o a e a e a u e a o o i a e e a e a o o o e e a i a o e u e o
a a e a a a e a o a u e o a i e a e a i u a o i a i o e i e i e e o e e e a
e e a i u o e o a u a e a o i e i a e i e a a i i e i a e i o o a a i i o a
a o o i o o e e a e i e e e a i i i a i o a i e a e i a o e i e o e o a e a
e o e e A e a a i o a e a e a i a o o e a o e a e i e e o i a a e a e a e a
e a a a i e a o e e i e u a o a a e a e o o e e e a a o u i e a e a i i a i
e e u i e o e e o e a e e i i i e i a e o e a e a u e e e i i i e e e a e a
o a a e e a o a e e e e a o o i a e a e a i a a e o a o e e e

```

```
In [40]: ▶ 1 lo = 0
2 uo = 0
3 c = 0
4
5 for char in s:
6     if char in 'aeiou':
7         lo += 1
8     elif char in 'AEIOU':
9         uo += 1
10    else:
11        c += 1
12    print(uo, "Upper Case Ovels", lo , "Lower", c, "Consants")
```

2 Upper Case Ovels 371 Lower 857 Consants

Accessing the characters from the string

- Indexing
 - +ve/Forward
 - -ve/Reverse
- Slicing
 - +ve/Forward
 - -ve/Reverse

```
In [41]: ▶ 1 print(s)
```

A leap year (also known as an intercalary year or bissextile year) is a calendar year that contains an additional day (or, in the case of a lunisolar calendar, a month) added to keep the calendar year synchronized with the astronomical year or seasonal year.[1] Because astronomical events and seasons do not repeat in a whole number of days, calendars that have a constant number of days in each year will unavoidably drift over time with respect to the event that the year is supposed to track, such as seasons. By inserting (called intercalating in technical terminology) an additional day or month into some years, the drift between a civilization's dating system and the physical properties of the solar system can be corrected. A year that is not a leap year is a common year.

For example, in the Gregorian calendar, each leap year has 366 days instead of 365, by extending February to 29 days rather than the common 28. These extra days occur in each year which is an integer multiple of 4 (except for years evenly divisible by 100, which are not leap years unless evenly divisible by 400). The leap year of 366 days has 52 weeks and two days, hence the year following a leap year will start later by two days of the week.

```
In [42]: ▶ 1 print(s[0])
```

A

In [43]: ▶ 1 `print(s[4])`

a

In [44]: ▶ 1 `print(s[-1])`

.

In [45]: ▶ 1 `print(s[-5])`

w