

6th Chapter

1st pdf

print() function

- 1) What does print() function do ? ---> Prints an object and moves to next line
- 2) How is the control moved to next line without '\n' ? ---> Since default end is '\n'
- 3) print('Hyd')
print('Sec')
print('Cyb')

Are the strings printed in same line (or) different lines ? ---> Different lines becoz default end is '\n'

- 4) Which object can print() function print (non-sequence (or) sequence) ? ---> Any python object
- 5) How many arguments can print() function take ? ---> 0, 1 (or) more than one
- 6) Where is print() function defined ? ---> In builtins module
- 7) Can print() function be used without import ? --->
Yes becoz it is automatically imported as it is a member of builtins module
- 8) What does print() function return ? ---> None

Program 2 (2nd pdf)

sep argument demo program (Home work)

```
a , b , c = 25 , 10.8 , 'Hyd'
print(a , b , c , sep = ',') # 25 , 10.8 , Hyd
print(a , b , c , sep = '\t') # 25 <tab> 10.8 <tab> Hyd
print(a , b , c , sep = '---') # 25 --- 10.8 --- Hyd
print(a , b , c , sep = '\n') # 25 <next line> 10.8 <next line> Hyd
print(a , b , c) # 25 <space> 10.8 <space> Hyd
#print(a , b , c , separator = ':') # Error becoz separator is an invalid arg to print()
'''
```

- 1) What is the default separator for print() function ? ---> Space
- 2) Can separator be modified ? ---> With sep argument of print() function
- 3) What does sep = 'delimiter' do ? ---> Modifies the separator to the specified delimiter
- 4) print(object , sep = ' , ')

Is sep argument required in the above print() function ? ---> No becoz single object is being printed

- 5) When is sep argument required in print() function ? ---> When more than one object is being printed
- 6) What is the separator between hours , minutes and seconds ? ---> ':' i.e. sep = ':'
What is the separator between date , month and year ? ---> '-' (or) '/' (or) ' '

'''

3rd pdf(3à)

end argument demo program

```
a , b , c = 10 , 20 , 30
```

```
print(a , end = '\t') # 10 <tab> <same line>
print(b , end = '\t') # 20 <tab> <same line>
print(c) # 30 <next line>
print('Bye') # Bye <next line>
#print(a , last = '\t') # Error becoz last is invalid argument for print() function
'''
```

- 1) What is the default end ? ---> '\n'
 - 2) Can end be modified ? ---> Yes with end argument of print() function
 - 3) What does end = 'delimiter' do ? ---> Modifies end to the specified delimiter
 - 4) How many lines of outputs is generated when the above program is executed ? ---> 2
- '''

4th pdf (3b)

```
# Find outputs (Home work)
a , b , c = 25 , 10.8 , 'Hyd'
print(a , b , c , end = '---') # 25 <space> 10.8 <space> Hyd --- <same line>
print(a , b , c , sep = ',') # 25 ,, 10.8 ,, Hyd <next line>
print(a , b , c , sep = ':::', end = '\t\t\t') # 25 ::: 10.8 ::: Hyd <tab><tab><tab> <same line>
print(a , b , c) # 25 <space> 10.8 <space> Hyd <next line>
```

```
'''
25 10.8 Hyd---25,,,10.8,,,Hyd
25:::10.8:::Hyd 25 10.8 Hyd
'''
```

- 1) How many lines of output is generated when the program is executed ? ---> 2
 - 2) What does sep argument do ? ---> Determines the delimiter to be printed 'between' the results
 - What does end argument do ? ---> Determines the delimiter to be printed at the 'end' of the line
- '''

5th pdf (4)

```
# Find outputs (Home work)
print('Hyd') # Hyd <next line>
print() # <next line>
print('Sec') # Sec <next line>
print() # <next line>
print('Cyb') # Cyb <next line>
'''
```

- 1) Hyd
- 2)
- 3) Sec
- 4)
- 5) Cyb

'''

'''

1) How many lines of output is generated when the program is executed ? ---> 5

2) What does print(No-args) do ? ---> Prints nothing and moves to next line

'''

6th pdf(5)

Find outputs (Home work)

```
l = [10 , 20 , 30 , 40]
```

```
t = (10 , 20 , 30 , 40)
```

```
s = {10 , 20 , 30 , 40}
```

```
print(l , t , s) # [10 , 20 , 30 , 40] <space> (10 , 20 , 30 , 40) <space> {10 , 20 , 30 , 40} <next line>
```

7th pdf(6à)

Formats demo program

```
a = 25.68
```

```
print('%d' %a) # Converts object 'a' to string integer i.e. '25'
```

```
print('%s' %a) # Converts object 'a' to string i.e. '25.68'
```

```
print('%f' %a) # Converts object 'a' to string float i.e. '25.680000'
```

```
print('%g' %a) # Converts object 'a' to string float i.e. '25.68'
```

```
#print('%f' , %a) # Error due to comma
```

```
print('a : ' , a) # a : <space> 25.68
```

```
print('%f' , a) # %f <space> 25.68
```

```
#print('%f' a) # Error becoz % is missing for object 'a'
```

```
#print('a : ' %a) # Error becoz format is missing
```

'''

Formats

1) What does '%d' %object do ? ---> Converts object to string integer due to %d

2) What is '%d' called ? ---> String integer

3) What is the alternative to '%d' format ? ---> '%i' i.e. String integer

4) What does '%f' %object do ? ---> Converts object to string float with 6 digits after decimal point due to %f

5) What is '%f' called ? ---> String float

6) What is the alternative to '%f' format ? ---> '%g'

7) Which is a better format between '%f' and '%g' ? ---> '%g' becoz unnecessary zeroes are not printed

8) How many digits after decimal point for %g ? ---> Variable number of digits

9) What does 'g' stand for ? ---> general

10) What does '%s' %object do ? ---> Converts object to string due to %s

11) What is %s called ? ---> String

12) What is % in '%s' called ? ---> Format specifier

13) What are the two ways to convert object to string ? ---> str(object) and '%s' %object

'''

8th pdf (6b)

Find outputs (Home work)

```
a = 25
```

```
b = '%f' %a # Converts object 'a' to string float
```

```
print(b) # '25.000000'
```

```
print(type(b)) # <class 'str'>
```

```
x = 10.8
```

```
y = '%d' %x # Converts object 'x' to string integer
```

```
print(y) # '10'
```

```
print(type(y)) # <class 'str'>
```

```
m = [10 , 20 , 15 , 18]
```

```
n = '%s' %m # Converts list to string list
```

```
print(n) # '[10 , 20 , 15 , 18]'
```

```
print(type(n)) # <class 'str'>
```

9th pdf(6f)

#Find outputs (Home work)

```
a = 25
```

```
b = 10.9274
```

```
c = 'Hyd'
```

```
print('%d %f %s' %(a , b , c)) # 25 10.927400 Hyd
```

```
print('%i %g %s' %(a , b , c)) # 25 10.9274 Hyd
```

```
print('%s %s %s' %(a , b , c)) # 25 10.9274 Hyd
```

```
print('%d %g %s' , a , b , c) # %d %g %s <space> 25 <space> 10.9274 <space> Hyd
```

```
#print('%d %g %s' a , b , c) # Error becoz % is missing for objects a , b and c
```

```
#print('%d %g %s' , %(a , b , c)) # Error due to comma
```

```
#print('%d %g %s' %a%b%c) # Error due to multiple %'s
```

```
print('%d' %a , '%f' %b , '%s' %c) # 25 <space> 10.927400 <space> Hyd
```

PDF 7a

```
# F string demo program
x = 25
print(F'{x}') # Converts object 'x' to string i.e. '25'
print(F'x') # 'x' itself becoz 'x' is not in { }
print('{x}') # {x}
print('x') # x
print(x) # Value of object 'x' i.e. 25
print(F'x = {x}') # x = 25
print(F'{x = }') # x = 25
print(f'x =') # x =
print(F'x : {x}') # x : 25
print(F'{x:}') # 25 and : is ignored
'''
```

- 1) What does F string do ? ---> Converts any python object to string
- 2) What is the syntax of F string ? ---> F'{object}'
- 3) What does F'{object}' do ? ---> Converts object to string
- 4) What does F stands for ---> Format string
- 5) Are F and f same ? ---> Yes
- 6) How to obtain object name and value of object with F string ? ---> F'{object=}'
- 7) What are the three ways to convert object to string ? ---> str(object) , '%s' %object , F'{object}'
- 8) What is the difference between eval() function and F string ? ---
eval() function converts string to any python object but
F '{object}' converts object to string
'''

Pdf 7b

```
# Find outputs (Home work)
x = 25
y = F'{x}' # Converts object 'x' to '25'
print(y) # '25'
print(type(y)) # <class 'str'>
x = 10.8
y = F'{x}' # Converts object 'x' to '10.8'
print(y) # '10.8'
print(type(y)) # <class 'str'>
x = [10,20,30,40]
y = F'{x}' # Converts list to string list
print(y) # '[10,20,30,40]'
print(type(y)) # <class 'str'>
```

PDF 7c

```
#Find outputs (Home work)
a , b , c = 25 , 10.8 , 'Hyd'
print(F'{a} \t {b} \t {c}') # 25 <tab> 10.8 <tab> Hyd
print(F'a = {a} \t b = {b} \t c = {c}') # a = 25 <tab> b = 10.8 <tab> c = Hyd
print(F'{a=} \t {b=} \t {c=}') # a = 25 <tab> b = 10.8 <tab> c = 'Hyd'
print(F'{a:} \t {b:} \t {c:}') # 25 <tab> 10.8 <tab> Hyd (colons are ignored)
print('a = {a} \t b = {b} \t c = {c}') # a = {a} <tab> b = {b} <tab> c = {c}
print(F'a = a \t b = b \t c = c') # a = a <tab> b = b <tab> c = c
#print(F'{x=} \t {y=} \t {z=}') # Error becoz there are no objects x , y and z
'''
```

How to obtain quotes for string output ? ---> print(F'{str-object=}')
'''

PDF 7d

```
# Find outputs (Home work)
x = 25
print(F'{x}') # 25
print(F'{{x}}') # {x}
print(F'{{{x}}}') # {25}
print(F'{{{x}}}') # {{x}}
print(F'{{{x}}}') # {{25}}
print(F'{{{x}}}') # {{{x}}}
print(F'{{{x}}}') # {{{25}}}
print(F'{{{x}}}') # {{{{x}}}}
print(F'{{{x}}}') # {{{{25}}}}
'''
```

- 1) What is printed when 'x' is in even number of braces ? ---> 'x' itself
 - 2) What is printed when 'x' is in odd number of braces ? ---> Value of 'x' in the form of string
 - 3) How many braces are printed in the output ? ---> Number of braces // 2
- '''

PDF 8

```
# R string demo program (Home work)
a = 'Hyd is \n green \t city'
print(a) # Hyd is <next line> green <tab> city
b = R'Hyd is \n green \t city'
print(b) # Hyd is \n green \t city
c = 'Hyd is \\n green \\t city'
print(c) # Hyd is \n green \t city
```

'''

1) What are the two ways to print `\n` ? ---> `print(R'\n')`

and

`print('\n')`

2) What does `print('\n')` do ? ---> Moves to next line twice but does not print `\n`

3) What are the two ways to print `\t` ? ---> `print(R'\t')`

and

`print('\t')`

4) What does `print('\t')` do ? ---> Generates a tab but does not print `\t`

5) What does R string do ? ---> Does not treat `\n` as new line character

and

`\t` is not treated as tab character

6) What does F stand for in F string ? ---> Format string

What does R stand for in R string ? ---> Raw string

7) Are 'R' and 'r' same ? ---> Yes

'''

PDF 8a

eval() function demo program

`print(eval('25'))` # Converts '25' to 25

`print(eval('10.8'))` # Converts '10.8' to 10.8

`print(eval('False'))` # Converts 'False' to False

`print(eval('3+4j'))` # Converts '3+4j' to 3+4j

`#print(eval('Hyd'))` # Error becoz 'Hyd' is converted to object Hyd which does not exist

`print(eval(" 'Hyd' "))` # Converts " 'Hyd' " to 'Hyd'

`print(eval('3 + 4 * 5'))` # Converts '3 + 4 * 5' to 3 + 4 * 5 = 23

`print(eval('[10 , 20 , 15 , 18]'))` # Converts '[10 , 20 , 15 , 18]' to [10 , 20 , 15 , 18]

`print(eval('{10,20,15,18,20,12,18}'))` # Converts '{10,20,15,18,20,12,18}' to {10,20,15,18,12}

`print(eval('(10 , 20 , 30)'))` # Converts '(10,20,30)' to (10,20,30)

`print(eval('{10 : 'Hyd' , 10 : 'Sec'}'))` # Converts "{10 : 'Hyd' , 10 : 'Sec'}" to {10 : 'Sec'}

`#print(eval(4 + 5))` # Error becoz 4 + 5 is not a string

'''

eval() function

1) What does eval() function do ? ---> Converts string to appropriate object

2) What does int(x) do ? ---> Converts object 'x' to integer

What does float(x) do ? ---> Converts object 'x' to float

What does complex(x) do ? ---> Converts object 'x' to complex number

What does bool(x) do ? ---> Converts object x' to boolean

3) What is the advantage of eval() function ? --->

It can be used as an alternative to int() , float() , complex() , bool() and so on

'''

```
# Gift
# Find outputs (Home work)
print(eval(" 'hyd' ")) # 'hyd'
hyd = 'Sec'
print(eval('hyd')) # Converts 'hyd' to object hyd i.e. 'Sec'
sec = '25'
print(eval('sec')) # Converts 'sec' to object sec i.e. '25'
print(eval(sec)) # Converts object sec (i.e. '25') to 25
cyb = 10.8
print(eval('cyb')) # Converts 'cyb' to object cyb i.e. 10.8
#print(eval(cyb)) # Error becoz cyb is not a string nor a str object
'''
```

1) hyd = 'Sec'

```
print(eval('hyd'))
```

What does eval('hyd') do ? ---> Converts 'hyd' to object hyd

How is the above print() reduced to ? ---> print(hyd)

What does print(hyd) do ? ---> Prints value of object hyd i.e. 'Sec'

2) sec = '25'

```
print(eval('sec'))
```

What does eval('sec') do ? ---> Converts 'sec' to object sec

How is the above print() reduced to ? ---> print(sec)

What does print(sec) do ? ---> Prints value of object sec i.e. '25'

3) sec = '25'

```
print(eval(sec))
```

What does eval(sec) do ? ---> Converts sec object to integer 25

How is the above print() reduced to ? ---> print(25)

What does print(25) do ? ---> Prints 25

4) cyb = 10.8

```
print(eval('cyb'))
```

What does eval('cyb') do ? ---> Converts 'cyb' to object cyb

How is the above print() reduced to ? ---> print(cyb)

What does print(cyb) do ? ---> Prints value of object cyb i.e. 10.8

5) cyb = 10.8

What is the issue with eval(cyb) ? ---> cyb is neither a string nor a str object

'''

```
# Gift
```

```
# Find output (Home work)
```

```
print(eval('print("Hyd")')) # Hyd <next line> None
```


'''

1) print(eval('print("Hyd")'))

What does eval() function do ? ---> Converts 'print("Hyd")' to statement print("Hyd")

2) print(eval('print("Hyd")'))

How is the above statement reduced to ? ---> print(print("Hyd"))

3) print(print("Hyd"))

What does inner print() function do ? ---> Prints Hyd and returns None

4) print(print("Hyd"))

How is the above statement further reduced to ? ---> print(None)

5) What does print(None) do ? ---> Prints None and returns None which is ignored

'''

PDF 9.1

Find outputs (Home work)

print(bool('False')) # True becoz 'False' is non-empty string

print(eval('False')) # Converts 'False' to False

print(bool('')) # False due to empty string

print(eval(' "" ')) # ""

#print(eval('')) # Error becoz there is nothing to print

print(eval(' " " ')) # Space i.e. " "

#print(eval(' ')) # Error becoz there is nothing to print

'''

1) What is the result of eval(' "" ') ? ---> Empty string i.e. ""

2) What is the result of eval(' " " ') ? ---> Space i.e. " "

3) What is the result of eval('') ? ---> Error becoz result is nothing

4) What is the result of eval(' ') ? ---> Error becoz result is space which is not in quotes

'''

PDF 9.2

Find outputs (Home work)

a = 'Hyd is green city'

print(a) # Hyd is green city

b = 'Hyd is "green" city'

print(b) # Hyd is "green" city

c = 'Hyd is \'green\' city'

print(c) # Hyd is 'green' city

#print('Hyd is ' green ' city') # Error due to single quote in single quote

'''

'''

1) What are the three ways to print single quote ? --->

a) print(" ' ") ---> Single quote in double quotes

'''

b) print(""" ' ') ---> Single quote in triple quotes

'''

c) print(' \' ') ---> \ is mandatory becoz single quote is used in single quote

2) What are the three ways to print double quote ? --->

a) print(' " ') ---> Double quotes in single quotes

'''

b) print(""" " """) ---> Double quote in triple quote

'''

c) print(" \" ") ---> \ is mandatory becoz double quote is used in double quotes

'''

PDF 10a

What is the advantage of eval(input()) ?

x = eval(input('Enter any input : '))

print(type(x))

print(x)

'''

Input input() reads eval(input()) returns

25 '25' eval('25') is 25

10.8 '10.8' eval('10.8') is 10.8

3+4j '3+4j' eval('3+4j') is 3+4j

False 'False' eval('False') is False

None 'None' eval('None') is None

Hyd 'Hyd' eval('Hyd') is object Hyd

'Hyd' " 'Hyd' " eval(" 'Hyd' ") is 'Hyd'

[25 , 10.8 , 'Hyd'] "[25 , 10.8 , 'Hyd']" eval("[25 , 10.8 , 'Hyd']") is [25 , 10.8 , 'Hyd']

1) What is the advantage of eval(input()) ? ---> It can read any type of input

What is the dis-advantage of eval(input()) ? ---> String input has to be in quotes

2) What does int(input()) do ? ---> Reads only integer input

What does float(input()) do ? ---> Reads either float input (or) integer input

What does complex(input()) do ? ---> Reads only complex input

3) Finally eval(input()) can be used as an alternative to int(input()) , float(input()) and so on

4) Which function is ideal to read employee number ? ---> int(input()) becoz emp number is usually integer

Which function is ideal to read employee name ? ---> input() becoz emp name is a string

Which function is ideal to read salary ? ---> float(input()) becoz salary is float

Which function is ideal to read gender ? ---> input() becoz gender is male (or) female which is a string

Which function is ideal to read married ? ---> eval(input()) but not bool(input())

'''

PDF 10b

Which is a better approach to read string input ?

```
a = input('Enter any string : ') # Hyd
```

```
print(len(a)) # 3
```

```
print(a) # Hyd
```

```
b = eval(input('Enter any string : ')) # 'Hyd'
```

```
print(len(b)) # 3
```

```
print(b) # Hyd
```

'''

1) What is the issue with eval(input()) for string input ? ---> Input string has to be in quotes

2) What is the advantage of input() function for string input ? ---> Input string should not be in quotes

3) Input input() reads eval(input()) returns

Hyd 'Hyd' eval('Hyd') converts 'Hyd' to object Hyd

'Hyd' " 'Hyd' " eval(" 'Hyd' ") is 'Hyd'

'''