

Output:

we use function print for displaying.

As print is a function, we should call or invoke using parentheses following print.

Characteristics:

a) can take argument of any type

```
>>> print(100)
100
>>> print(2.5)
2.5
>>> print(True)
True
>>> print([11, 22])
[11, 22]
>>> print("enjoy")
enjoy
```

b) can take any number of arguments

```
print(1, 2, 3, 4)
1 2 3 4
```

c) each argument is evaluated as an expression

```
>>> print(2 + 2, 3 * 4)
4 12
```

d) In the display, we observe a space between the output fields.

By default, output field separator is a space.

It can be changed by specifying `sep = <val>` in print.

e) After each print, we get a newline. This is called the output record separator.

This can be changed by specifying `end = <val>` in print.

Please run and check the output of the following program. Some of this code is tricky!

```
# file : 1_output.py
# output
#     output field separator : appears between fields in the output
#         default : space
#         use sep to change this
#     output record separator : appears at the end of each print
#         default : newline
#         use end to change this
"""

print("one", "two", "three")
print("four", "five")

print("one", "two", "three", sep = "-----", end = "\n*****\n")
print("four", "five", sep = "^^^^^^^^")
"""

#end = "" # NO
print("to", end = "")
print("get", end = "")
print("her")
"""

sep = "stupid"
print("rama", "krishna")
print("apple", "banana", sep = "fool") # does not create a variable called sep
print(sep) # stupid and not fool
"""

sep = " stupid "
# value of variable sep is substituted and nothing special
print("rama", "krishna", "parama ", "hamsa ", " not ", sep) # sep refers to the
variable in the program
```

```
print("apple", "banana", "carrot", sep = sep) # field separator takes the value  
of the variable sep
```