Department of Computer Science and Engineering
PES University, Bangalore, India

Lecture Notes
Python for Computational Problem Solving
UE23CS151A

Lecture #12
Output function

By,
Prof. Sindhu R Pai,
Anchor, PCPS - 2023
Assistant Professor
Dept. of CSE, PESU

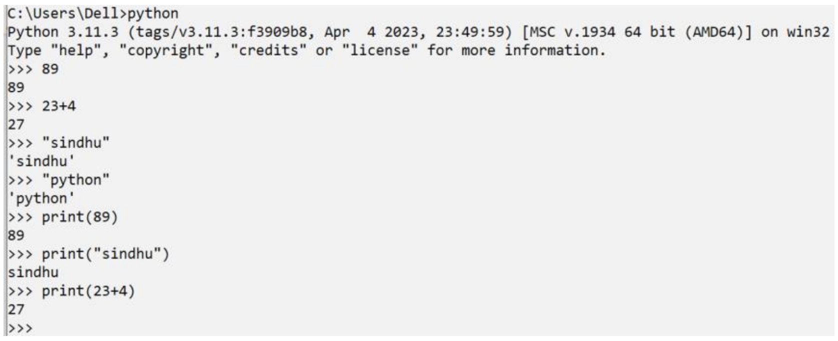
Verified by,
PCPS Team - 2023

Many Thanks to
Dr. Shylaja S S (Director, CCBD and CDSAML Research Centers, Former Chairperson, CSE, PES University)
Prof. Chitra G M (Asst. Prof, Dept. of CSE, PCPS Anchor – 2022)

Output Function – print:

Consider the below executions.

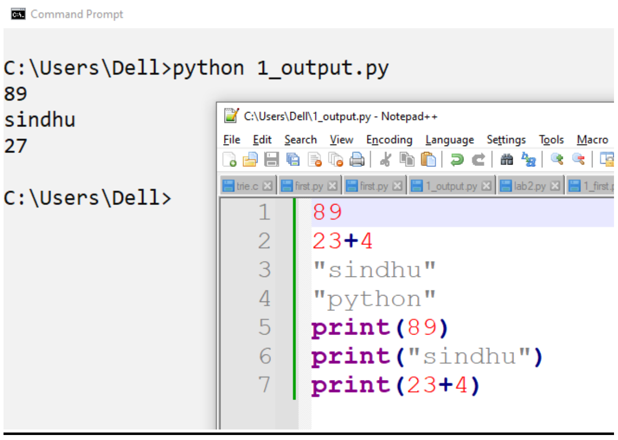
Interpreter mode



```
C:\Users\Dell>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 89
89
>>> 23+4
27
>>> "sindhu"
'sindhu'
>>> "python"
'python'
>>> print(89)
89
>>> print("sindhu")
sindhu
>>> print(23+4)
27
>>>
```

Observation: Runs **one statement at a time**. Here, **we need not tell the interpreter** what to be displayed because of the statement execution. When you press the enter key, **the result is displayed without any extra instruction** like print. It does the same job when you use the print function in interpreter mode.

Batch mode


Command Prompt
C:\Users\Dell\1_output.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro
trie.c first.py first.py 1_output.py lab2.py 1_first

```
C:\Users\Dell>python 1_output.py
89
sindhu
27

C:\Users\Dell>
```

```
89
23+4
"sindhu"
"python"
print(89)
print("sindhu")
print(23+4)
```

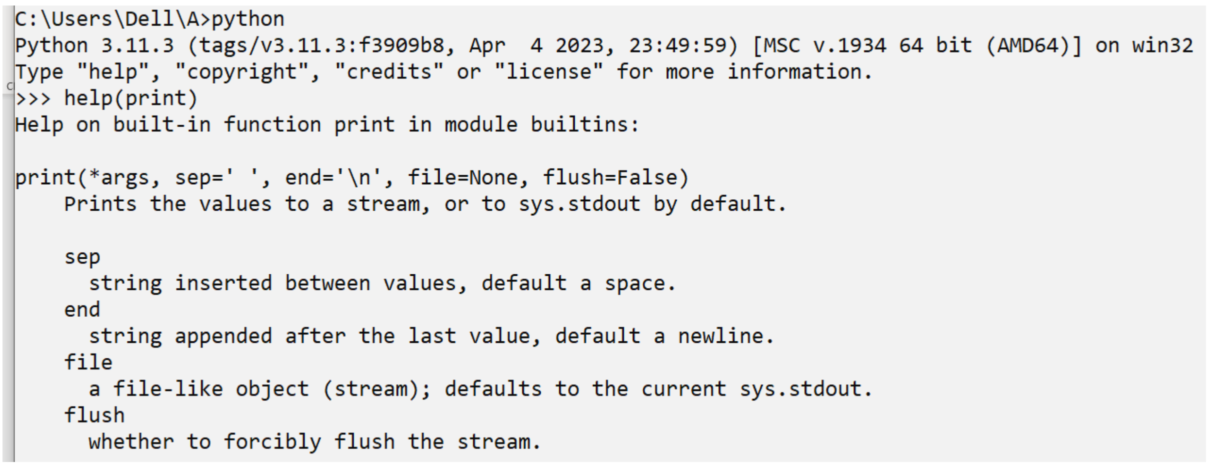
Observation: No output from the first four lines of code. Output of next three lines are there in the output window. Here, we **need to explicitly tell the interpreter** what should be displayed as a result of the program. Python has a specialized function that does just that, called the **print function.**

Note: In Batch mode, we must use the function print for displaying information to the output screen

Function: A group of commands that performs a specified task. It may take some input, do some processing, and returns a result(output). In programming, input given to a function are called arguments. For more details, refer to Lecture #51

Print in detail :

Print is a function that takes input and as part of the processing, displays the value of the argument to the output screen. We will be calling this print function using a **call operator ().** Let us get the help page of print to understand it better.



```
C:\Users\Dell\A>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> help(print)
Help on built-in function print in module builtins:

print(*args, sep=' ', end='\n', file=None, flush=False)
    Prints the values to a stream, or to sys.stdout by default.

    sep
      string inserted between values, default a space.
    end
      string appended after the last value, default a newline.
    file
      a file-like object (stream); defaults to the current sys.stdout.
    flush
      whether to forcibly flush the stream.
```

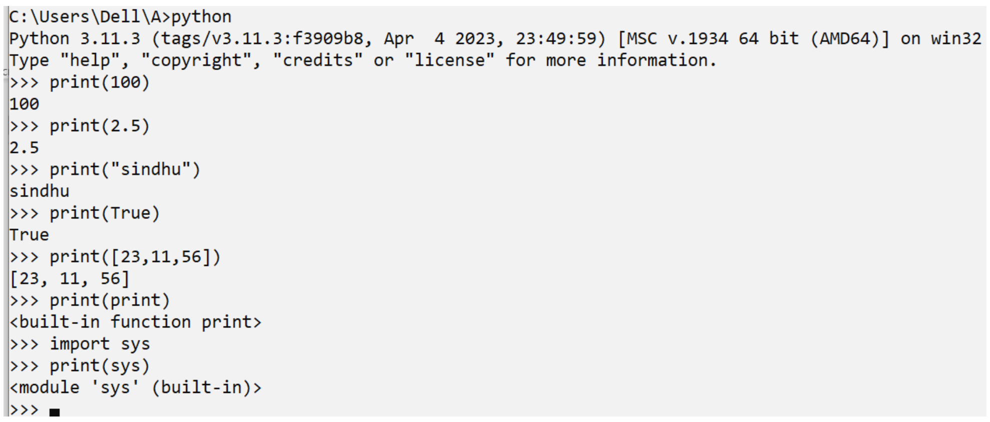
Characteristics of print function:

- Can display/print anything in the world.
- Can take any type of arguments.

  Values could be Numbers, Boolean, strings, collections, expressions, functions, etc.
- Can take any number of arguments.
- Has the capability to evaluate the expression.
- Formatting is possible to some extent using the keyword separators – sep & end

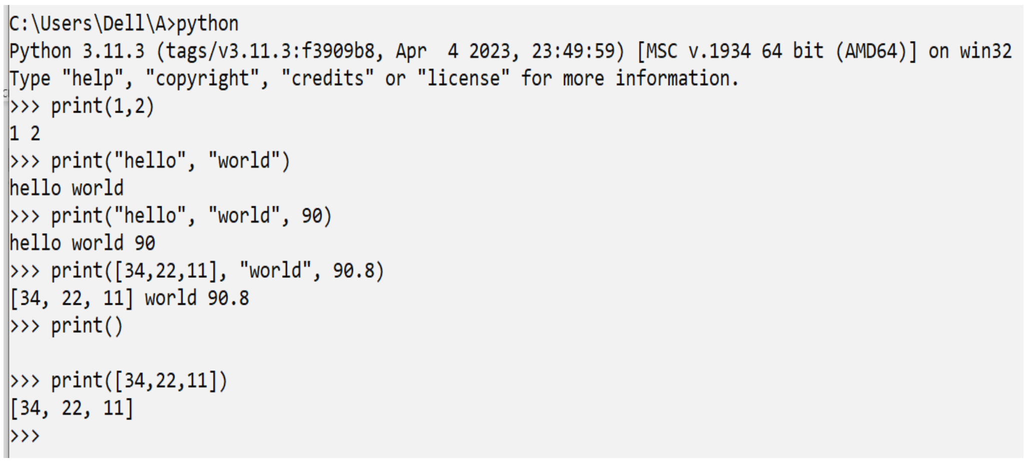
Few Examples:

Example 1: Can take any type of arguments



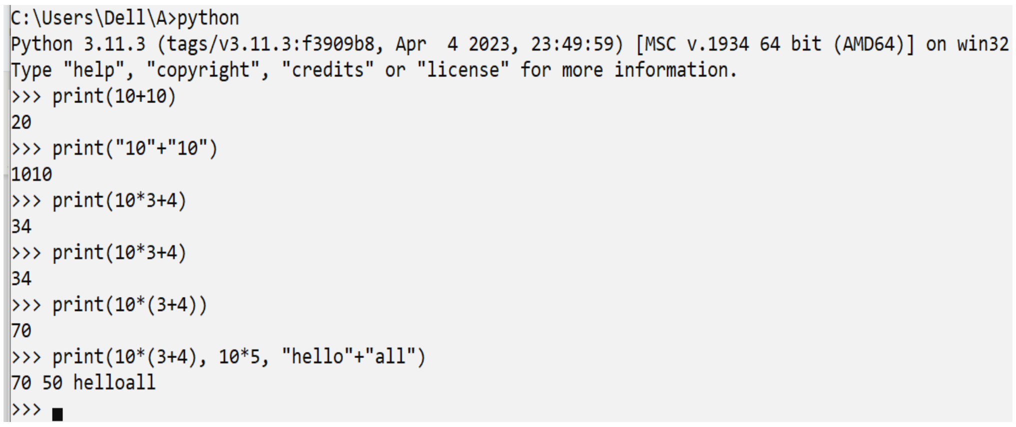
```
C:\Users\Dell\A>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print(100)
100
>>> print(2.5)
2.5
>>> print("sindhu")
sindhu
>>> print(True)
True
>>> print([23,11,56])
[23, 11, 56]
>>> print(print)
<built-in function print>
>>> import sys
>>> print(sys)
<module 'sys' (built-in)>
>>>
```

Example 2: Can take any type of arguments of any type



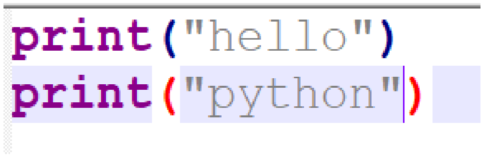
```
C:\Users\Dell\A>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print(1,2)
1 2
>>> print("hello", "world")
hello world
>>> print("hello", "world", 90)
hello world 90
>>> print([34,22,11], "world", 90.8)
[34, 22, 11] world 90.8
>>> print()

>>> print([34,22,11])
[34, 22, 11]
>>>
```

Example 3: Can be used to evaluate the expression

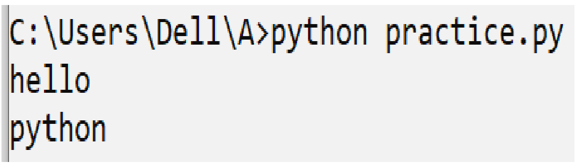


```
C:\Users\Dell\A>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print(10+10)
20
>>> print("10"+"10")
1010
>>> print(10*3+4)
34
>>> print(10*3+4)
34
>>> print(10*(3+4))
70
>>> print(10*(3+4), 10*5, "hello"+"all")
70 50 helloall
>>>
```

Example 4: Batch Mode executions



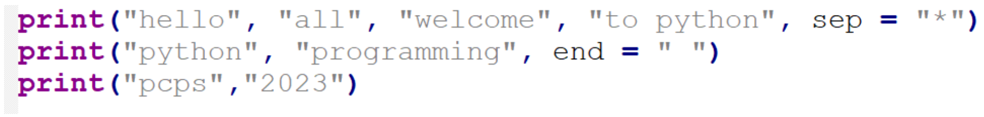
```
print("hello")
print("python")
```



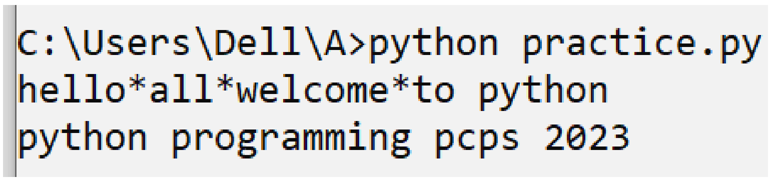
```
C:\Users\Dell\A>python practice.py
hello
python
```

In all above examples, we see that if there is more than one argument to print, they are separated by a space and between two print calls, there is a new line in the output. This is because of the keyword parameters sep and end respectively. **The default value for sep is space (' ') and the default value for end is '\n' (newline).**

Example 5:



```
print("hello", "all", "welcome", "to python", sep = "*")
print("python", "programming", end = " ")
print("pcps","2023")
```



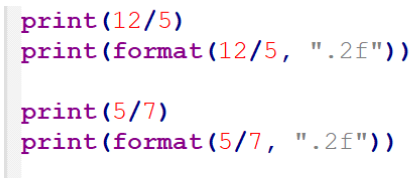
```
C:\Users\Dell\A>python practice.py
hello*all*welcome*to python
python programming pcps 2023
```

Few points to note in Example 5:

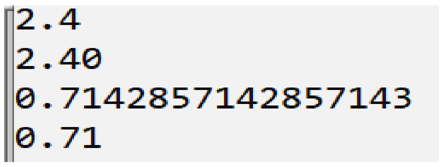
The sep and end field behavior is only applicable for the print function in which it is specified. The subsequent print function calls will use its default value as per the help page of print.

Now think about printing only 2 digits after the decimal point in each expression like 5/7. Possible using the **format function** in python. The built-in format function can be **used to produce a numeric value containing a specific number of decimal places.** More about this will be discussed in Lecture #17.

Example 6:



```
print(12/5)
print(format(12/5, ".2f"))

print(5/7)
print(format(5/7, ".2f"))
```



```
2.4
2.40
0.7142857142857143
0.71
```

- END -