

# Data Science & Artificial Intelligence



## Python For Data Science

### Functions

Lecture No.- 01

By- Satya sir



# Recap of Previous Lecture



- Tuples
  - Indexing
  - Slicing
  - Unpacking
  - Nested Tuples
- Sets
  - Sets methods (or) operations



# Topics to be Covered



- Set methods/operations
- Dictionaries
  - Nested Dictionaries
  - Access elements
  - Update/Delete/Add
- Functions in Python



## Topic : Sets and Dictionaries

### Example

$$S_1 = \{ 'p', 'y', 't', 'h', 'o', 'n' \}$$

$$S_2 = \{ 'j', 'a', 'v', 'd', 'c', 'p', 'p' \}$$

$$S_3 = \{ 'x', 'u', 'b', 'y', 'p', 'e', 'd', 'l' \}$$

$$S_4 = S_1 \cup S_3 \quad (\text{or}) \quad S_1.\text{union}(S_3) \neq S_4 = \{ 'p', 'y', 't', 'h', 'o', 'n', 'x', 'u', 'b', 'e', 'l' \}$$

$$S_5 = S_3 \cap S_2 \quad (\text{or}) \quad S_3.\text{symmetric\_difference}(S_2) \neq S_5 = \{ 'd', 'a', 'v', 'c', 'x', 'u', 'b', 'y', 'e', 'l' \}$$

$$S_6 = S_2 - S_1 \quad (\text{or}) \quad S_2.\text{difference}(S_1) \neq S_6 = \{ 'j', 'a', 'v', 'c' \}$$

$$S_7 = S_1 \cap S_2 \quad (\text{or}) \quad S_1.\text{intersection}(S_2) \neq S_7 = \{ 'p' \}$$

$$\text{Print}(\text{len}(S_4) + \text{len}(S_5) - \text{len}(S_6) + \text{len}(S_7))$$

$$\neq 11 + 10 - 4 + 1 = \underline{18}$$





## Topic : Sets and Dictionaries



### Example-2

$$S1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

$$S2 = \{2, 4, 6, 8, 10\}$$

$$S3 = \{0, 1, 3, 5, 7, 9\}$$

$$S4 = S1 - S3 \quad \# \quad S4 = \{2, 4, 6, 8\}$$

$$\text{Count} = \text{len}(S4) \quad \# \quad 4$$

for i in range(len(S3 & S1)):

for j in range(len(S2 & S4)):

$$\text{Count} -= i - j$$

$$\# \text{Count} = \text{Count} - (i - j) \\ = \text{Count} - i + j$$

Print(count)

o/p: \_\_\_\_\_

$$\text{Count} = 4$$

$$S3 \& S1 = \{1, 3, 5, 7\} \Rightarrow \text{len}() = 4$$

$$S2 \cap S4 = \{10\} \quad \text{len}() = 1$$

$$j = 0$$

$$i = 0 \quad \text{Count} = 4 - 0 + 0 = 4$$

$$i = 1 \quad \text{Count} = 4 - 1 + 0 = 3$$

$$i = 2 \quad \text{Count} = 3 - 2 + 0 = 1$$

$$i = 3 \quad \text{Count} = 1 - 3 + 0 = -2$$

$$\# \underline{\underline{\text{Count} = -2}}$$





## Topic : Sets and Dictionaries



### Example-3

$a = \{ 'E', 'x', 'A', 'M', 'I', 'N', 'A', 'T', 'T', 'O', 'N' \}$

$b = \{ 'R', 'E', 'P', 'E', 'T', 'I', 'T', 'T', 'O', 'N' \}$

$c = \{ 'S', 'O', 'L', 'U', 'T', 'I', 'O', 'N' \}$

$d = \{ 'P', 'R', 'E', 'P', 'A', 'R', 'A', 'T', 'I', 'O', 'N' \}$

$$i = \text{len}(a) - \text{len}(b) + \text{len}(d) \# 8 - 7 + 8 = 9$$

$$j = \text{len}(c) + \text{len}(b) - \text{len}(d) \# 7 + 7 - 8 = 6$$

$$\text{ans} = i + j \# \text{ans} = 15$$

$\{ 'E', 'I', 'N', 'T', 'O' \}$

for  $i$  in range( $\text{len}(a \cdot \text{intersection}(b))$ ):

for  $j$  in range( $\text{len}(c \cdot \text{difference}(c))$ ):

$\{ 'P', 'R', 'E', 'A', 'T' \}$

$\text{ans} += i + j + \text{len}(c \cdot \text{Symmetric\_difference}(a))$

$\{ 'E', 'x', 'A', 'M', 'S', 'L', 'U' \} = 7$

Print(ans) **#225**

	j=0	1	2	3
i=0	$15 + 0 + 0 + 7 = 22$	$22 + 1 + 7 = 30$	$30 + 2 + 7 = 39$	$39 + 3 + 7 = 49$
i=1	$49 + 1 + 7 = 57$	$57 + 1 + 1 + 7 = 66$	$66 + 1 + 2 + 7 = 76$	$76 + 11 = 87$
i=2	$87 + 9 = 96$	$96 + 10 = 106$	$106 + 11 = 117$	$117 + 12 = 129$
i=3	$129 + 10 = 139$	$139 + 11 = 150$	$150 + 12 = 162$	$162 + 13 = 175$
i=4	$175 + 11 = 186$	$186 + 12 = 198$	$198 + 13 = 211$	$211 + 14 = 225$





## Topic : Sets and Dictionaries



$$S1 = \{\cancel{1}, \cancel{2}, \cancel{3}\} = \{\cancel{2}, \cancel{3}\} = \{4, 5, 7, 9\}$$

$$S2 = \{\cancel{2}, \cancel{3}, \cancel{4}, \cancel{5}\} = \{2, 3, 4, 5, 7, 9\}$$

$$S3 = \{\cancel{3}, \cancel{5}, \cancel{7}, \cancel{9}\} = \{5, 7, 9\}$$

$$S1. \text{intersection\_update}(S2) \quad \# S1 = \{2, 3\}$$

$$S2. \text{union\_update}(S3) \quad \# S2 = \{2, 3, 4, 5, 7, 9\}$$

$$S3. \text{difference\_update}(S1) \quad \# S3 = \{5, 7, 9\}$$

$$S1. \text{symmetric\_difference\_update}(S2) \quad \# S1 = \{4, 5, 7, 9\}$$



## Topic : Sets and Dictionaries



### Dictionaries

- Ordered, mutable collection of (key : value) Pairs.
  - In a dictionary, Each Element will have 2 Parts : 1) key 2) value associated with key.
  - Add, remove items and modification of values of keys is allowed, but keys cannot be modified.
  - Support duplicates for values only. [keys cannot be duplicated]
  - To create dictionaries :  $\text{object} = \{ \text{'key}_1' : \text{value}, \text{'key}_2' : \text{value}, \text{'key}_3' : \text{value} \dots \}$
  - To create empty dictionary :  $\text{object} = \{ \}$  (or)  $\text{object} = \text{dict}()$ 
    - Keys need to be of str type
    - Values can be of any type
- Ex:  $d = \{ \text{'abc'} : 10, \text{'xyz'} : 4.23, \text{'ijk'} : \text{True}, \text{'pqr'} : \text{'wow'} \}$



## Example (Add | modify | remove)

$d = \{ 'a': 3, 'b': 2, 'c': 4 \}$

$d['b'] = 6$  #  $d = \{ 'a': 3, 'b': 6, 'c': 4 \}$

$d['e'] = 8$  #  $d = \{ 'a': 3, 'b': 6, 'c': 4, 'e': 8 \}$

$\text{del } d['b']$  #  $d = \{ 'a': 3, 'c': 4, 'e': 8 \}$

$d.\text{pop}('c')$  #  $d = \{ 'a': 3, 'e': 8 \}$

$d['f'] = 10$  #  $d = \{ 'a': 3, 'e': 8, 'f': 10 \}$

$d.\text{popitem}()$  #  $d = \{ 'a': 3, 'e': 8 \}$

$\text{del } d$  # Entire dictionary gets deleted.

$d.\text{update}(\{ 'x': 10 \})$

#  $d = \{ 'x': 10 \}$

$e = d.\text{copy}()$

#  $e = \{ 'x': 10 \}$



## Topic : Sets and Dictionaries



Method	Description
<code>clear()</code>	Removes all the elements from the dictionary
<code>copy()</code>	Returns a copy of the dictionary
<code>fromkeys()</code>	Returns a dictionary with the specified keys and value
<code>get()</code>	Returns the value of the specified key
<code>items()</code>	Returns a list containing a tuple for each key value pair
<code>keys()</code>	Returns a list containing the dictionary's keys
<code>pop()</code>	Removes the element with the specified key
<code>popitem()</code>	Removes the last inserted key-value pair
<code>setdefault()</code>	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
<code>update()</code>	Updates the dictionary with the specified key-value pairs
<code>values()</code>	Returns a list of all the values in the dictionary



## Iterate over Dictionary elements

$d = \{ 'a': 1, 'b': 2, 'c': 5, 'e': 3 \}$

`Print(d.keys())`  $\neq \{ 'a', 'b', 'c', 'e' \}$

`Print(d.values())`  $\neq \{ 1, 2, 5, 3 \}$

`Print(d)` (or) `Print(d.items())`  $\neq \{ 'a': 1, 'b': 2, 'c': 5, 'e': 3 \}$

for  $i$  in `d.keys()`:

for  $j$  in `d.values()`: (OR)

`Print(i, j)`

for  $i, j$  in `d.items()`:

`Print(i, j)`

$\Rightarrow$

O/P:	<code>'a': 1</code>
	<code>'b': 2</code>
	<code>'c': 5</code>
	<code>'e': 3</code>



## Topic : Functions - 1



Function : A single statement (or) multiple statements, that Performs specific Task (or) Subtask.

→ 2 Types : 1) Pre-defined / System-defined / Library / Built-in functions : Ex: `len()`, `type()`, `oct()`, `hex()`, `bin()` -- `id()` --

2) User-defined functions : The functions created/developed by User

Syntax : `def function_Name(argument(s)):`  
`Statement(s)`

Ex: `def display( ):`  
`Print('Hello')`

- For functions to get Executed, they need to be invoked / Called.

`display()`

Syntax: `function name(argument(s))`

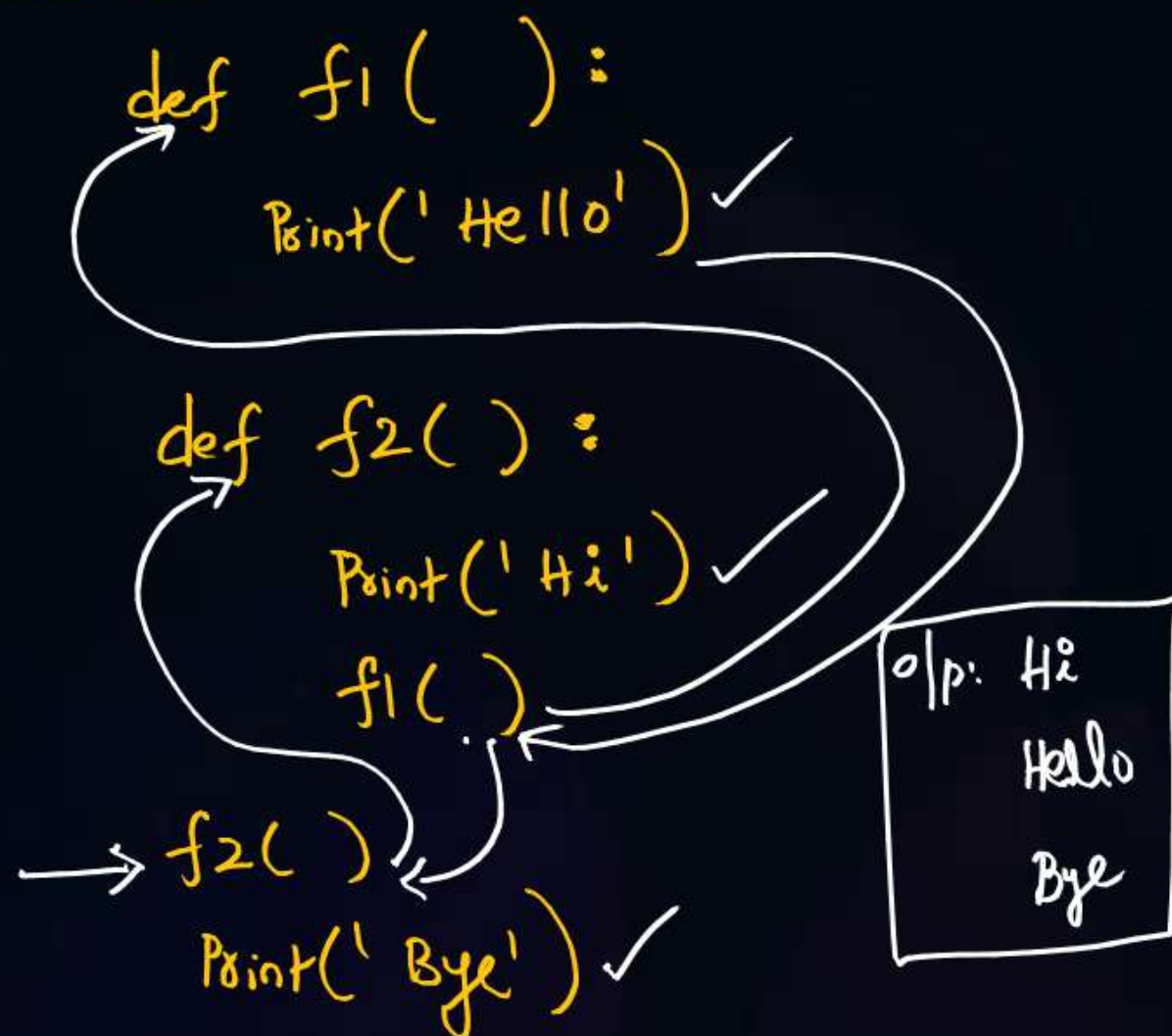




## Topic : Functions - 1

- A Function can call another function(s)

Example :



## Recursion

- A function, which calls itself is said to be recursive function.

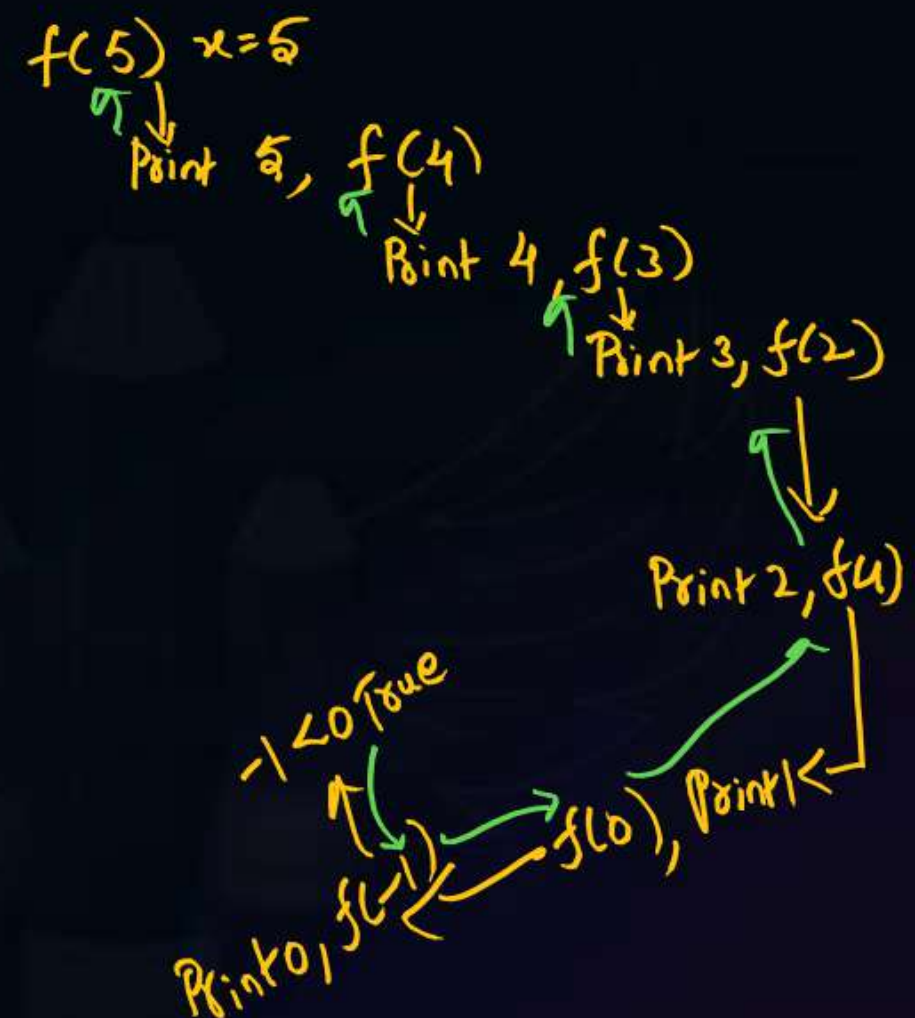
- The Process of calling itself is Recursion.

Ex:

```
def f(x):
    if x < 0:
        return
    Print(x)
    f(x-1) ✓
```

f(5)

o/p: 5 4 3 2 1 0





## 2 mins Summary



- Sets
- Dictionaries
- functions\*

To be contd . . .







**THANK - YOU**