



SNS COLLEGE OF ENGINEERING
Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



DEPARTMENT OF CSE



19IT103 – COMPUTATIONAL THINKING AND PYTHON PROGRAMMING

❖ A readable, dynamic, pleasant, flexible, fast and powerful language



UNIT II DATA TYPES, EXPRESSIONS, STATEMENTS

- Python interpreter and interactive mode, debugging; values and types: int, float, boolean, string, and list; variables, **expressions, statements, tuple assignment**, precedence of operators, comments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.





Tuple Assignment

- Tuple is **sequence** data type.
- Initialise or create a tuple in various ways.
- The process of **assigning values** to a tuple is known as **packing**.
- The **unpacking** or tuple assignment is the process that **assigns** the values on the **right-hand side to the left-hand side variables**.





Tuple Packing (Creating Tuples)

- Tuple can contain all elements of the same data type as well as of mixed data types as well.

```
>>>tup = (22, 33, 5, 23)

>>>tup

(22, 33, 5, 23)
```





Tuple Packing (Creating Tuples)

- Tuple with mixed data type

```
>>>tup2 = ('hi', 11, 45.7)
```

```
>>>tup2
```

```
('hi', 11, 45.7)
```





Tuple Packing (Creating Tuples)

- Tuple with a tuple as an element

```
>>>tup3 = (55, (6, 'hi'), 67)

>>>tup3

(55, (6, 'hi'), 67)
```

```
>>> tup3 = (55, (6, 'hi'), 67)
>>> tup3[1][1]
'hi'
```





Tuple Packing (Creating Tuples)

- Tuple with a list as an element

```
>>>tup3 = (55, [6, 9], 67)
```

```
>>>tup3
```

```
(55, [6, 9], 67)
```

```
>>> tup3 = (55, [6, 9], 67)
```

```
>>> tup3[1][0]
```

```
6
```

```
>>> |
```





Tuple Packing (Creating Tuples)

- If there is only a single element in a tuple we should end it with a comma.
- Since writing, just the element inside the parenthesis will be considered as an integer.

```
>>> tup=(90)
>>> type(tup)
<class 'int'>
>>> tup=(90,)
>>> type(tup)
<class 'tuple'>
>>> |
```





Tuple Packing (Creating Tuples)

- If there is only a single element in a tuple we should end it with a comma.
- Since writing, just the element inside the parenthesis will be considered as an integer.

```
>>> tup=(90)
>>> type(tup)
<class 'int'>
>>> tup=(90,)
>>> type(tup)
<class 'tuple'>
>>> |
```





Tuple Packing (Creating Tuples)

- If you write any sequence separated by commas, python considers it as a tuple.

```
>>> seq = 22, 4, 56
```

```
>>> print(seq)
```

```
(22, 4, 56)
```

```
>>> type(seq)
```

```
<class 'tuple'>
```

```
>>> |
```





Tuple Assignment (Unpacking)

- Unpacking or tuple assignment is the process that assigns the values on the right-hand side to the left-hand side variables.

```
>>>(n1, n2) = (99, 7)
```

```
>>>print(n1)
```

```
99
```

```
>>>print(n2)
```

```
7
```



Tuple Assignment (Unpacking)

```
>>>tup1 = (8, 99, 90, 6.7)

>>>(roll no., english, maths, GPA) = tup1

>>>print(english)

99

>>>print(roll no.)

8

>>>print(GPA)

6.7

>>>print(maths)

90
```

Tuple Assignment (Unpacking)

```
>>> (num1, num2, num3, num4, num5) = (88, 9.8, 6.8, 1)
```

```
#this gives an error as the variables on the left are more than the  
number of elements in the tuple
```

```
ValueError: not enough values to unpack
```

```
(expected 5, got 4)
```