


 [sandeepsuryaprasad](#) / [python_tutorials](#) Private[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Ins](#) master ▾[python_tutorials](#) /
[3_booleans_packing_unpacking](#) /
[_unpacking.py](#) / [<> Jump to ▾](#)[Go to file](#)

...

**Sandeep Suryaprasad** testing

Latest commit a868eb3 on 29 Nov 2021

[History](#) 0 contributors

95 lines (76 sloc) | 2.15 KB

[Raw](#)[Blame](#)

```
1  """
2  1. A sequence is an object which can be indexed.
3  2. All Sequences are Iterables. But all iterables are not sequences.
4  """
5
6  point = (4, 5) # Unpacking Tuple
7  x, y = point
8  print(x)
9  print(y)
10
11 data = ['IBM', 50, 91.1, (2019, 7, 17)]
12 name, shares, price, date = data # Unpacking list of items
13 print(name)
14 print(shares)
15 print(price)
16 print(date)
17
18 y, m, d = date # Unpacking Date
19
20 print(y)
21 print(m)
22 print(d)
23
24 s = 'Hello'
25 a, b, c, d, e = s # Unpacking String
26 print(a)
```

```
27 print(b)
28 print(c)
29 print(d)
30 print(e)
31
32 # enumerate function
33 names = ['apple', 'google', 'gmail', 'yahoo', 'yahoo']
34
35 # Tuple unpacking
36 for item in enumerate(name):
37     index, name = item
38     print(index, name)
39
40 # Tuple unpacking (in for loop)
41 for index, name in enumerate(name):
42     print(index, name)
43
44 # zip function
45 a = [1, 2, 3, 4]
46 b = [5, 6, 7, 8]
47
48 # Tuple unpacking
49 for item in zip(a, b):
50     first_number, second_number = item
51     print(first_number, second_number)
52
53 # Tuple unpacking (unpacking in for loop)
54 for first_number, second_number in zip(a, b):
55     print(first_number, second_number)
56
57 # Unpacking a dictionary (method-1)
58 d = {'one': 1, 'two': 2, 'three': 3}
59 for item in d.items():
60     k, v = item
61     print(k, v)
62
63 # Unpacking a dictionary (method-2)
64 for k, v in d.items():
65     print(k, v)
66
67 # Normal unpacking
68 temperatures = {"Bangalore": (26, 32), "Chennai": (29, 35), "Delhi": (31, 36)}
69 for city, temperatures in temperatures.items():
70     print(city, temperatures)
71
```

```
72 # Deep Unpacking
73 for city, (_min, _max) in temperatures.items():
74     print(city, _min, _max)
75
76 # Unpacking Elements from iterables of Arbitrary length
77 # * is used to grab excess arguments/items
78 # * is called unpacking operator
79 least, *rest, maximum = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
80 print(least)    # Prints first item in the list
81 print(maximum)  # Prints last item in the list
82 print(rest)     # Prints all the item in between 1st and last item of the list
83 print(max(rest))
84 print(min(rest))
85
86 *trailing, current = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
87 print(trailing)
88 print(current)
89
90 a, b, *c = range(1, 10)
91
92 # Ignoring certain values while unpacking
93 # _ is called throwaway variable in python
94 data = ['IBM', 50, 91.1, (2019, 7, 17)]
95 name, *_ , (year, *_ ) = data
```