

Advanced data types and functions

by : R.K.Pradhan

tuples

- Simply put , tuples are immutable lists
- Immutable => You can't change the size of the tuple or the elements within it
- Represented with '()'
- Eg : `(1, 2, 3)` , `('a', 'bb', 9, 23)`
- Why do I need tuples ? (it's fast)
- Tuple is also an iterator(?)
- Indexing is same as in lists

tuples

- Can be nested, just like lists
- Changing values :

```
a = (1, 2, 3)
```

```
a[1]=5 # wrong
```

```
b = tuple((5 if i==2 else i for i in a))
```

- Adding items:

```
(1, 2, 3) + (4, 5)
```

- Deleting items:

```
a = a[:2] # slicing
```

dictionaries

- Objects with key-value pairs
- Contains a list of items where each item is a pair, i.e (key,value)
- Represented as '{}'
- Eg : `D = { 'a':5, 'b':6, 'c':7 }`
- Accessing items : `D['a']` # will print 5
- **No variables**

dictionaries

- `D = { 'a':2, 'b':3, 'c':5 }`
- `D.keys()` # will print `['a', 'b', 'c']`
- `D.values()` # will print `[2,3,5]`
- `D.items()` # will print `[('a',2), ('b',3), ('c',5)]`
- `dict([(1,2), (3,5)])` # will return `{1:2, 3:5}`
- Why do I need a dictionary ?
- **Practice problem** : Reverse a dictionary .

sets

- A set is an iterator where no same item is present more than once
- $A = \{1, 2, 3, 5, 9, 6\}$
- Why use sets ?
- Union : $A \mid B$
- Intersection : $A \& B$
- Complement : $A - B$

sets

- Symmetric difference : $A \oplus B$
- Subset checking : $A \subset B$
- Superset checking : $A \supset B$
- Converting list/tuples/strings into sets,
`set(list/tuple/string)`
- **Practice problem** : remove duplicate characters of a string

Some special built in functions

- **zip**

Format : `zip(iterator1, iterator2, iterator3, ...)`

- **map**

Format : `map(function_name, iterator)`

- **filter**

Format : `filter(function_name, iterator)`

Assignments - 1/1

- Input a string and create a dictionary where the words are keys and the count of those words are values.
- Using map function square each element of a 3x3 square matrix(3 rows and 3 columns).
- Using zip function and list comprehension duplicate each character in a string once. For example : for input 'abc' output should be 'aabbcc'