Python Programming
Dict 3

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Insertion order: NOW preserved (Python 3.7)

```python
dict = {}    # empty dict
dict[20] = 10
dict['mostafa'] = 10
dict[30] = 15
dict[(2, 7)] = 150
dict[30] = 10
# observe: values can be anything and can repeat

print(dict.keys())
# dict_keys([20, 'mostafa', 30, (2, 7)])
# Starting from python 3.7 specification : the keys order is preserved (insertion order)
# However, due to several reasons, it is still best practice to not depend on that
# Maybe after 10 years. For now, if order matter: use OrderedDict
# In practice: typically u don't care about insertion order but sorted keys themselves
```

Keys!

```python
dict = {'x': 11, 'b': 22, 'y': 30}

print(dict.items())      # dict_items([('x', 11), ('b', 22), ('y', 30)])

for key, value in dict.items():
    print(key, value)    # x 11 b 22 y 30

print(dict.keys())       # dict_keys(['x', 'b', 'y'])
print(list(dict.keys())) # ['x', 'b', 'y']

for key in dict.keys():
    print(key, dict[key])    # same, but slower (extra access)

for key in sorted(list(dict.keys())):
    print(key, dict[key])    # sorted keys: b x y

for key in sorted(dict):     # shortcut for ordered keys
    print(key, dict[key])    # sorted keys: b x y
```

List vs Dict

```python
lst = [10, 20, 30, 40]
print(lst)
# [10, 20, 30, 40]
# list: ordered sequence
# can be indexed or sliced


dict = {0:10, 3:40, 2:30, 1:20}
print(list(dict.values()))
# [10, 40, 30, 20]
# dict: ORDERED collection of key-value-pairs
# items INSERTION order is preserved (3.7)
```

Shallow copies

```python
class Employee:
    def __init__(self):
        self.id = 10
    def __repr__(self):
        return str(self.id)
emp = Employee()
lst = [5, 8, 9]
dict = {'ziad' : 25.5, 2 : lst, 'Hey' : emp}
print(dict) # {'ziad': 25.5, 2: [5, 8, 9], 'Hey': 10}

lst.pop()
emp.id += 100
print(dict) # {'ziad': 25.5, 2: [5, 8], 'Hey': 110}

lst = [5]
print(dict) # {'ziad': 25.5, 2: [5, 8], 'Hey': 110}
d2 = dict.copy()
print(d2['Hey'] is emp) # True - shallow copy
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."