

Step0 - Dictionary

May 8, 2018

0.0.1 Dictionary

Each key is separated from its value by a colon (:), the items are separated by commas, and the whole thing is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this: {}. Keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples.

0.0.2 Creating a dictionary

```
In [1]: # Creating a dictionary
```

```
dict = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}

print "Sample dictionary: ", dict
```

```
Sample dictionary: {'Age': 6, 'Name': 'Jivin', 'Class': 'First'}
```

0.0.3 Accessing items from dictionary

```
In [3]: # Accessing items from dictionary
```

```
print "Value of key Name, from sample dictionary:", dict['Name']
```

```
Value of key Name, from sample dictionary: Jivin
```

0.0.4 Deleting a dictionary

```
In [11]: # Deleting a dictionary
```

```
dict = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
print "Sample dictionary: ", dict
del dict['Name'] # Delete specific item
print "Sample dictionary post deletion of item Name:", dict

dict = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
dict.clear() # Clear all the contents of dictionary
```

```

print "dict post dict.clear():", dict

dict = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
del dict # Delete the dictionary

```

Sample dictionary: {'Age': 6, 'Name': 'Jivin', 'Class': 'First'}

Sample dictionary post deletion of item Name: {'Age': 6, 'Class': 'First'}

dict post dict.clear(): {}

0.0.5 Updating a dictionary

In [12]: *# Updating a dictionary*

```

dict = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
print "Sample dictionary: ", dict
dict['Age'] = 6.5

print "Dictionary post age value update: ", dict

```

Sample dictionary: {'Age': 6, 'Name': 'Jivin', 'Class': 'First'}

Dictionary post age value update: {'Age': 6.5, 'Name': 'Jivin', 'Class': 'First'}

0.0.6 Basic operations on dictionary

In [31]: *# Basic operations*

```

dict = {'Name': 'Jivin', 'Age': 6, 'Class': 'First'}
print "Length of dict: ", len(dict)

dict1 = {'Name': 'Jivin', 'Age': 6};
dict2 = {'Name': 'Pratham', 'Age': 7};
dict3 = {'Name': 'Pranuth', 'Age': 7};
dict4 = {'Name': 'Jivin', 'Age': 6};
print "Return Value: dict1 vs dict2", cmp (dict1, dict2)
print "Return Value: dict2 vs dict3", cmp (dict2, dict3)
print "Return Value: dict1 vs dict4", cmp (dict1, dict4)

# String representation of dictionary
dict = {'Name': 'Jivin', 'Age': 6}
print "Equivalent String: ", str (dict)

# Copy the dict
dict1 = dict.copy()
print dict1

# Create new dictionary with keys from tuple and values to set value
seq = ('name', 'age', 'sex')

```

```

dict = dict.fromkeys(seq)
print "New Dictionary: ", str(dict)

dict = dict.fromkeys(seq, 10)
print "New Dictionary: ", str(dict)

# Retrieve value for a given key
dict = {'Name': 'Jivin', 'Age': 6};
print "Value for Age: ", dict.get('Age')
# Since the key Education does not exist, the second argument will be returned
print "Value for Education: ", dict.get('Education', "First Grade")

# Check if key in dictionary
print "Age exists? ", dict.has_key('Age')
print "Sex exists? ", dict.has_key('Sex')

# Return items of dictionary
print "dict items: ", dict.items()

# Return items of keys
print "dict keys: ", dict.keys()

# return values of dict
print "Value of dict: ", dict.values()

# if key does not exists, then the arguments will be added to dict and returned
print "Value for Age : ", dict.setdefault('Age', None)
print "Value for Sex: ", dict.setdefault('Sex', None)

# Concatenate dicts
dict = {'Name': 'Jivin', 'Age': 6}
dict2 = {'Sex': 'male' }

dict.update(dict2)
print "dict.update(dict2) = ", dict

```

Length of dict: 3

Return Value: dict1 vs dict2 -1

Return Value: dict2 vs dict3 1

Return Value: dict1 vs dict4 0

Equivalent String: {'Age': 6, 'Name': 'Jivin'}

{'Age': 6, 'Name': 'Jivin'}

New Dictionary: {'age': None, 'name': None, 'sex': None}

New Dictionary: {'age': 10, 'name': 10, 'sex': 10}

Value for Age: 6

Value for Education: First Grade

Age exists? True



```
Sex exists?  False
dict items:  [('Age', 6), ('Name', 'Jivin')]
dict keys:   ['Age', 'Name']
Value for Age :  6
Value for Sex:  None
dict.update(dict2) =  {'Age': 6, 'Name': 'Jivin', 'Sex': 'male'}
Value of dict:  [6, 'Jivin', 'male']
```

Reference :mastering machine learning with python in six steps