Programming for Data Science (CSE3041)

Ramesh Ragala

VIT Chennai Campus

July 26, 2020

Introduction to Dictionary

Problem

Write a kids play program that prints the capital of a country given the name of the country.

Introduction to Dictionary

Introduction to Dictionary

- Pair of items
- Each pair has key and value
- Keys should be unique
- Key and value are separated by :
- Each pair is separated by comma(,)

Example:

dict = {'Alice' : 1234, 'Bob' : 1235}

Introduction to Dictionary

Properties of Dictionary

- Unordered mutable collections;
- Items are stored and fetched by key,
- Accessed by key, not offset position
- Unordered collections of arbitrary objects
- Variable-length, heterogeneous, and arbitrarily nestable

Introduction to Dictionary

Creating a Dictionary

- Creating an **EMPTY dictionary**
  dictname = {}
- Example:
  - Dict1 = {}
  - MyDict = {}
  - Books = {}

- Creating a **dictionary with items**
  dictname = {key1:val1, key2:val2, ...}
- Example:
  - MyDict = { 1 : 'Chocolate', 2 : 'Icecream'}
  - MyCourse = {'MS' : 'Python', 'IT' : 'C', 'CSE' : 'C++', 'MCA' : 'Java'}
  - MyCircle = {'Ramesh' : 9486028245, 'Mom': 9486301601}

Accessing Values

- Using keys within square brackets
- >>> print(MyDict[1])
  'Chocholate'
- >>> print (MyCourse['CSE'])
  'C++'

Accessing Values

- Using keys within square brackets
- >>> print(MyDict[1])
  'Chocholate'
- >>> print (MyCourse['CSE'])
  'C++'

Updating Elements

- Update by adding a new item (key-value) pair
- Modify an existing entry
  >>> MyDict[1] = 'Pizza'
  >>> MyCourse['MCA'] = 'UML'

Introduction to Dictionary

Accessing Values

- Using keys within square brackets
- `>>> print(MyDict[1])`
  `'Chocholate'`
- `>>> print (MyCourse['CSE'])`
  `'C++'`

Updating Elements

- Update by adding a new item (key-value) pair
- Modify an existing entry
  `>>> MyDict[1] = 'Pizza'`
  `>>> MyCourse['MCA'] = 'UML'`

Deleting Elements

- Remove an element in a dictionary using the key
  `>>> del MyCourse['IT']`
- Remove all the elements
  `>>> MyCourse.clear()`
- Delete the dictionary
  `>>> del MyCourse`

Introduction to Dictionary

Basic Operations

- >>> D = {'spam': 2, 'ham': 1, 'eggs': 3 }
- >>> len(D)   # Number of entries in dictionary → 3
- >>> 'ham' in D   # Key membership test
  True
- >>> list(D.keys())   # Create a new list of D's keys
  ['eggs','spam','ham']
- >>> list(D.values())→ [3, 2, 1]
- >>> list(D.items()) → [('eggs', 3), ('spam', 2), ('ham', 1)]
- >>> D.get('spam')   # A key that is there
  2
- >>> print(D.get('toast'))   # A key that is missing
  None

Introduction to Dictionary

Update Method

- >>> D → {'eggs': 3, 'spam': 2, 'ham': 1}
- >>> D2 = {'toast':4, 'muffin':5}
- >>> D.update(D2)
- >>> D

  {'eggs': 3, 'muffin': 5, 'toast': 4, 'spam': 2, 'ham': 1} # Unordered

Introduction to Dictionary

Update Method

- >>> D → {'eggs': 3, 'spam': 2, 'ham': 1}
- >>> D2 = {'toast':4, 'muffin':5}
- >>> D.update(D2)
- >>> D
  {'eggs': 3, 'muffin': 5, 'toast': 4, 'spam': 2, 'ham': 1} # Unordered

Pop Method

Delete and return value for a given key

- >>> D = {'eggs': 3, 'muffin': 5, 'toast': 4, 'spam': 2, 'ham': 1}
- >>> D.pop('muffin')
  5
- >>> D.pop('toast')
  4
- >>> D → output →{'eggs': 3, 'spam': 2, 'ham': 1}

Introduction to Dictionary

List Vs Dictionary

- >>> L = []
- >>> L[99] = 'spam'
- Traceback (most recent call last): File "< *stdin* >" , line 1, in ? IndexError: list assignment index out of range
- >>>D = {}
- >>> D[99] = 'spam'
- >>> D[99] → check output →'spam'
- >>> D → check output → {99: 'spam'}

Introduction to Dictionary

Nesting in Dictionaries

- >>> jobs = []
- >>> jobs.append('developer')
- >>> jobs.append('manager')
- >>>rec = {}
- >>> rec['name'] = 'Bob'
- >>> rec['age'] = 40.5
- >>> rec['job'] = jobs
- >>> rec
- {'name': 'Bob', 'age': 40.5, 'job': ['developer', 'manager']}
- >>> rec['name']
  'Bob'
- >>> rec['job']
  ['developer','manager']
- >>> rec['job'][1]
  'manager'

Introduction to Dictionary

Other Ways to Make Dictionaries

- D = {'name': 'Bob', 'age': 40}
- D = {}   # Assign by keys dynamically
- D['name'] = 'Bob'
- D['age'] = 40
- # Creating a dictionary by assignment
- dict(name='Bob', age=40)
- # Creating dictionary with tuples form
- dict([('name', 'Bob'), ('age', 40)])

Introduction to Dictionary

Comprehensions in Dictionaries

- >>> D = {k: v for (k, v) in zip(['a', 'b', 'c'], [1, 2, 3])}
- >>> D
  {'b': 2, 'c': 3, 'a': 1}
- >>> D = {x: x ** 2 for x in [1, 2, 3, 4]}
- # Or: range(1, 5)
- >>> D
  {1: 1, 2: 4, 3: 9, 4: 16}
- >>> D = {c: c * 4 for c in 'SPAM'}
- >>> D
  {'S': 'SSSS', 'P': 'PPPP', 'A': 'AAAA', 'M': 'MMMM'}
- >>> D = {c.lower(): c + '!' for c in ['SPAM', 'EGGS', 'HAM']}
- >>> D
  {'eggs': 'EGGS!', 'spam': 'SPAM!', 'ham': 'HAM!'}

Introduction to Dictionary

Initializing Dictionaries

- # Initialize dict from keys
- >>> D = dict.fromkeys(['a', 'b', 'c'], 0)
- >>> D
  {'b': 0, 'c': 0, 'a': 0}
- # Same, but with a comprehension
- >>> D = {k:0 for k in ['a', 'b', 'c']}
- >>> D
  {'b': 0, 'c': 0, 'a': 0}
- # Comprehension
- >>> D = {k: None for k in 'spam'}
- >>> D
  {'s': None, 'p': None, 'a': None, 'm': None}

Introduction to Dictionary

Dictionary methods

- $< dict >$.items()
  - displays the items in the dictionary (pair of keys and values)
- $< dict >$.keys()
  - display the keys in the dictionary
- $< dict >$.values()
  - displays the values in the dictionary
- $< dict >$.pop()
  - removes the last item from the dictionary
- $< dict2 > = < dict1 >$.copy()
  - copies the items from dict1 to dict2
- $< dict >$.clear()
  - removes all the items from the dictionary

Other methods

- str(dict) - produces printable string representation of a dictionary
- len(dict) - returns the number of items in the dictionary

Introduction to Dictionary

- Dictionaries can replace elif ladder/switch-case
- print ({1:'one',2:'two',3:'three',4:'four',5:'five'} [*choice*])
- if choice = 3 then the code prints three

Introduction to Dictionary

Exercise 1:

Write a program to maintain a telephone directory of the employees of an organization. If the employee has more than one number store all the numbers. Write a program to print the mobile numbers given full or part of the name of the employee. Eg: Given name of the employee as 'John' the program must print phone numbers of 'John Paul' and 'Michel John'.

Exercise 2:

Write a program to store the name of the players against each of a 20-20 cricket team. The program should print the name of the players given the team name.

thank you