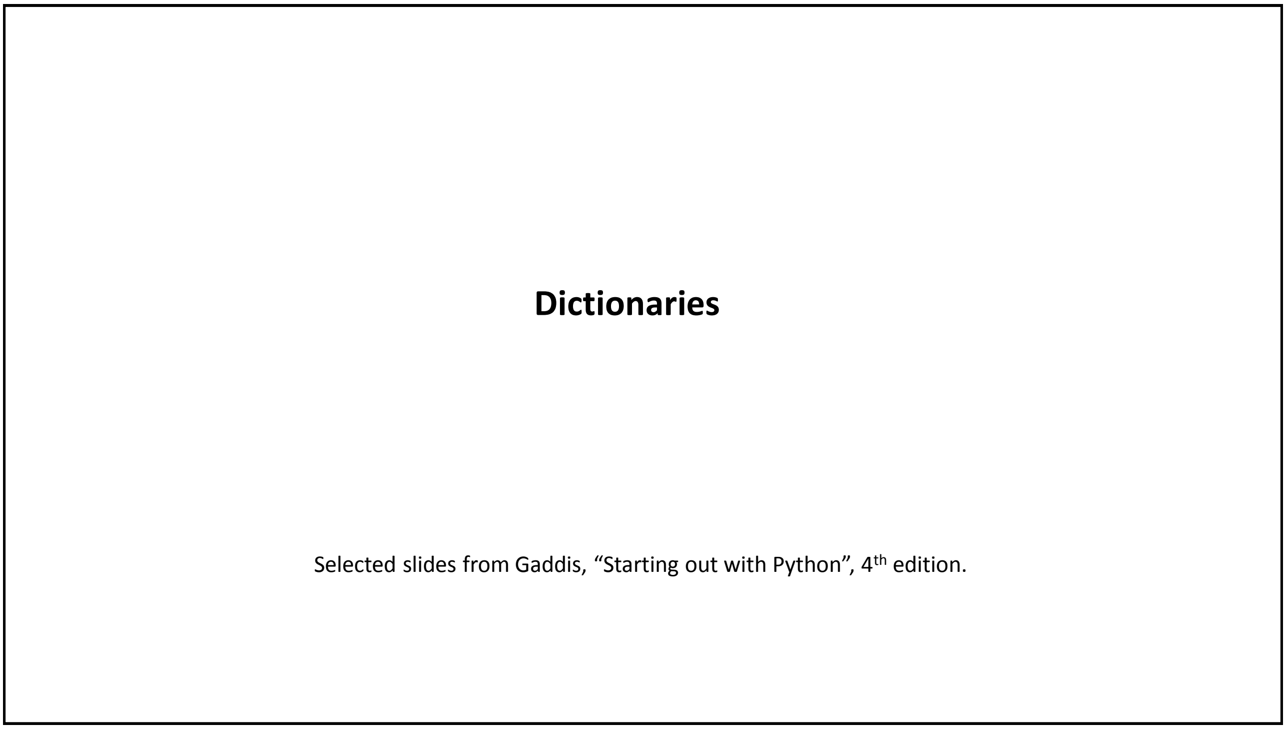
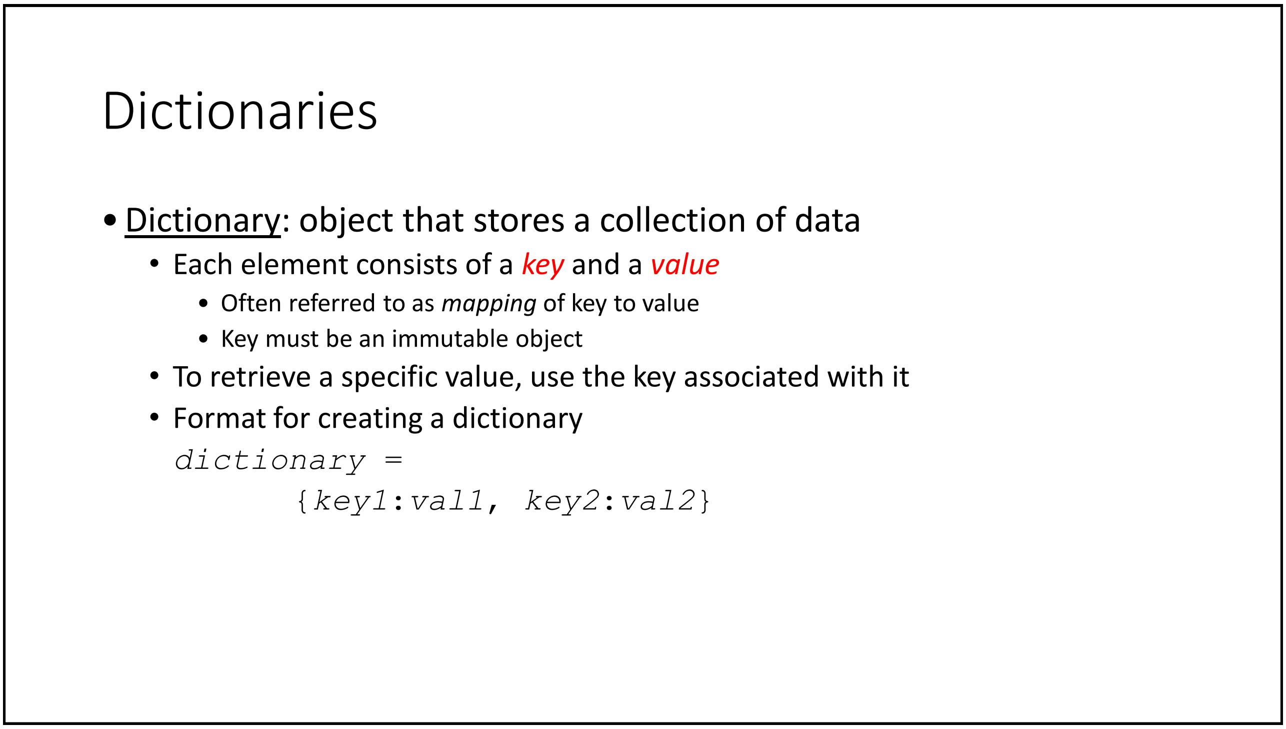


# Dictionaries

Selected slides from Gaddis, “Starting out with Python”, 4th edition.



## Dictionaries

- Dictionary: object that stores a collection of data
  - Each element consists of a *key* and a *value*
    - Often referred to as *mapping* of key to value
    - Key must be an immutable object
  - To retrieve a specific value, use the key associated with it
  - Format for creating a dictionary

```
dictionary =
       {key1:val1, key2:val2}
```

## Retrieving a Value from a Dictionary

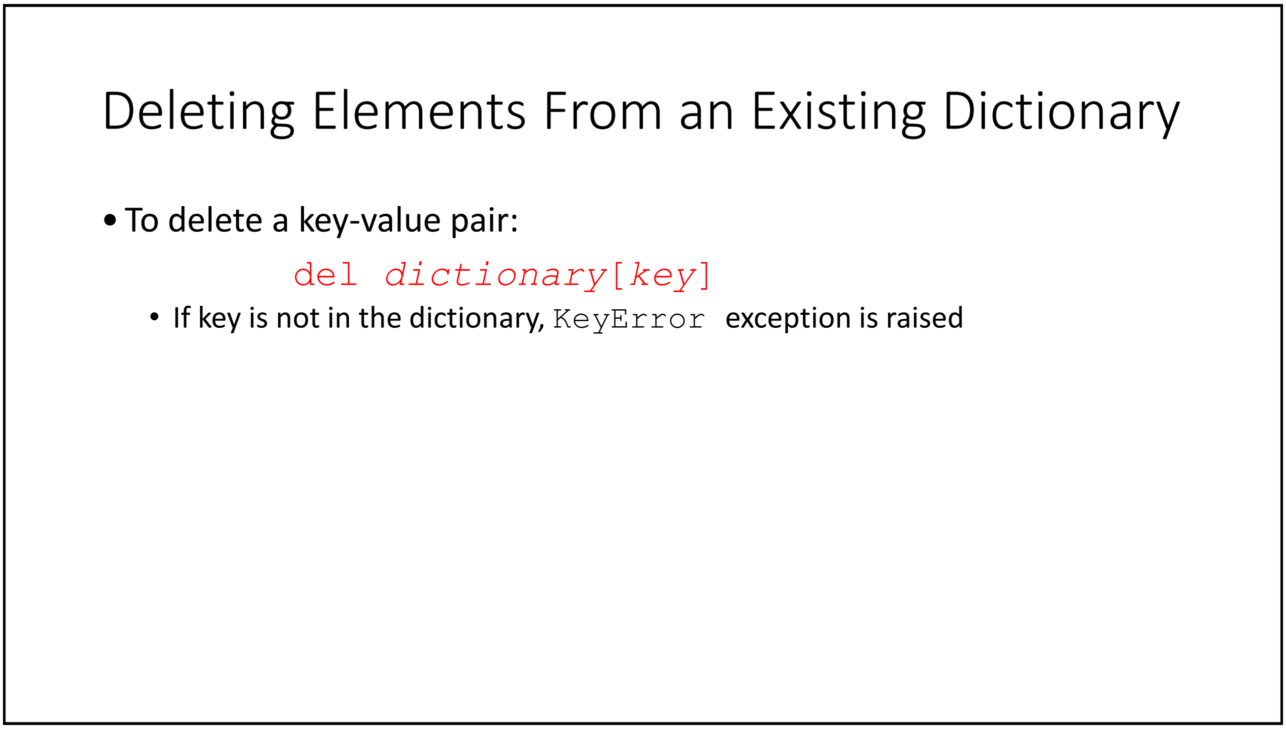
- Elements in dictionary are unsorted
- General format for retrieving value from dictionary:
  `dictionary[key]`
  - If `key` in the dictionary, associated value is returned, otherwise, `KeyError` exception is raised
- Test whether a key is in a dictionary using the `in` and `not in` operators
  - Helps prevent `KeyError` exceptions

## Adding Elements to an Existing Dictionary

- Dictionaries are mutable objects
- To add a new key-value pair:

  ```
  dictionary[key] = value
  ```

  - If key exists in the dictionary, the value associated with it will be changed

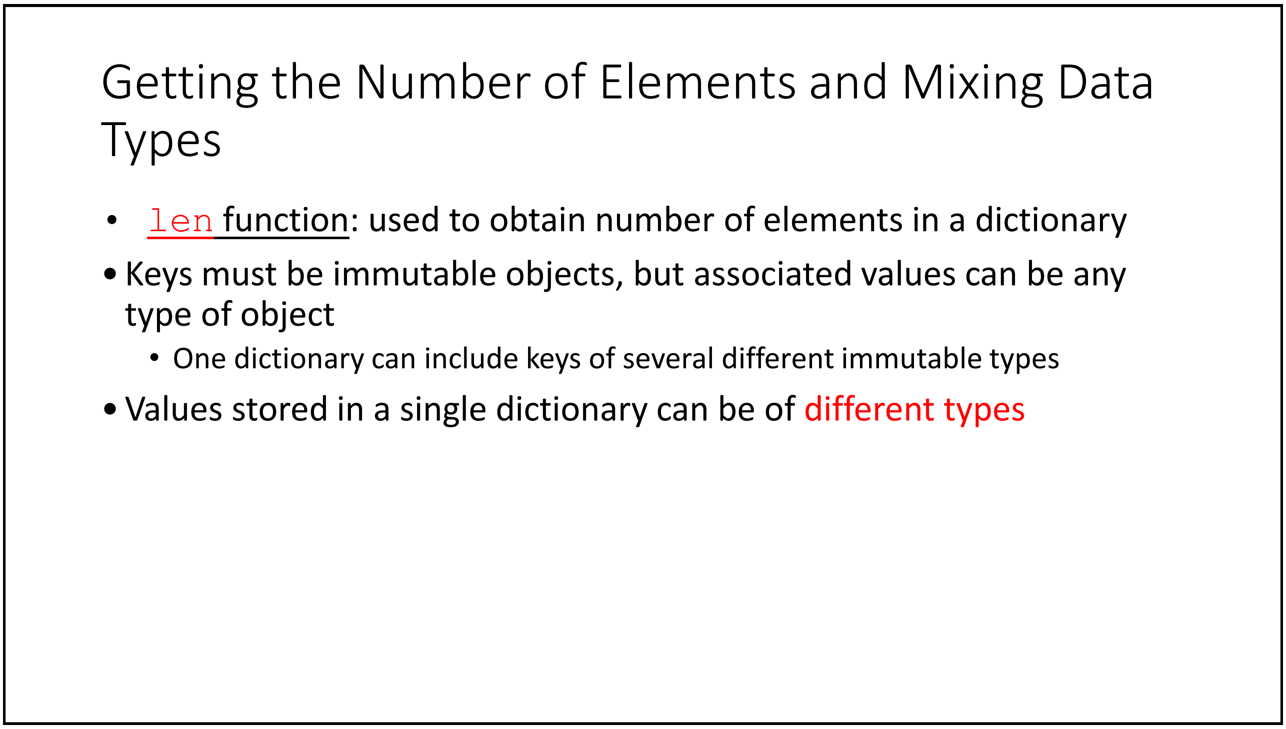


## Deleting Elements From an Existing Dictionary

- To delete a key-value pair:

  `del dictionary[key]`

  - If key is not in the dictionary, `KeyError` exception is raised



## Getting the Number of Elements and Mixing Data Types

- `len` function: used to obtain number of elements in a dictionary
- Keys must be immutable objects, but associated values can be any type of object
  - One dictionary can include keys of several different immutable types
- Values stored in a single dictionary can be of different types

## Creating an Empty Dictionary

- To create an empty dictionary:
  - Use `{}` → `phonebook = {}`
  - Use built-in function `dict()` →

    `phonebook = dict()`
  - Elements can be added to the dictionary as program executes

    `phonebook[‘Tom’] = ‘555-555-8888’`

## Using for Loop to Iterate Over a Dictionary

- Use a for loop to iterate over a dictionary
- General format: for key in dictionary:

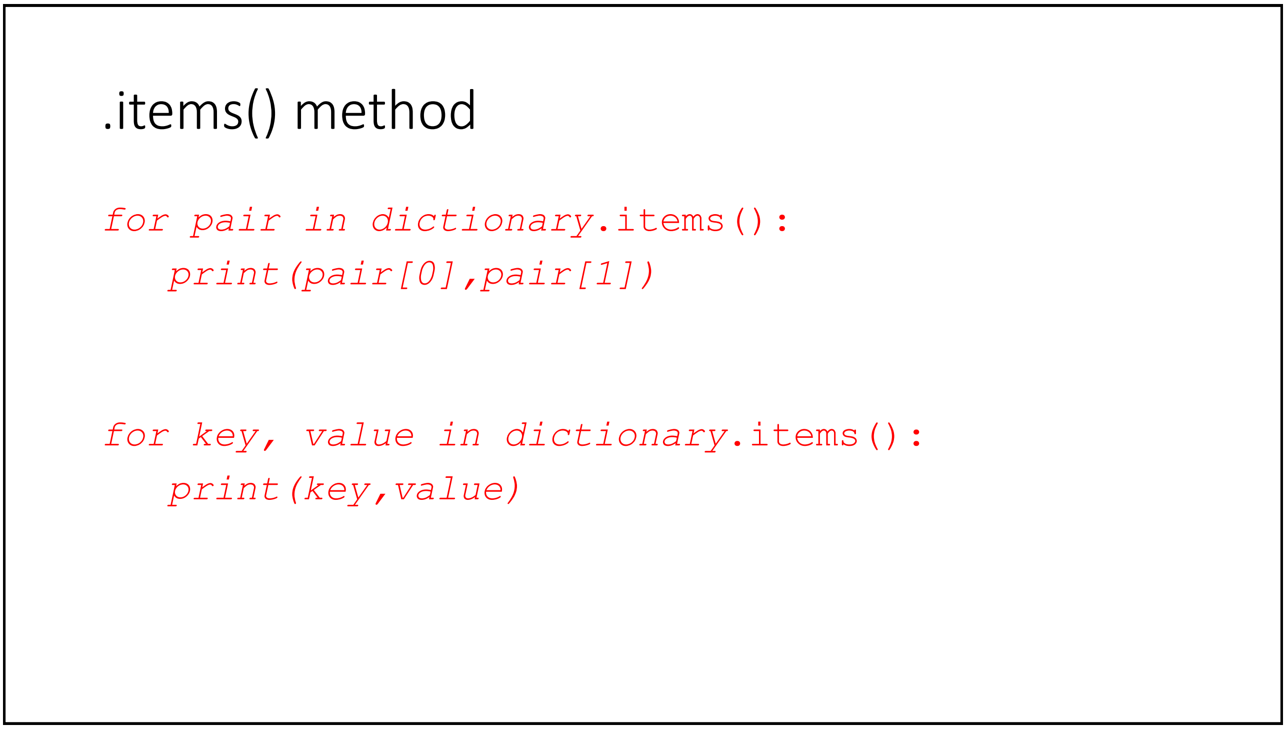
```
for key in phonebook:
  print(key)
```

## Some Dictionary Methods

- `clear` method: deletes all the elements in a dictionary, leaving it empty
  - Format: `dictionary.clear()`
- `get` method: gets a value associated with specified key from the dictionary
  - Format: `dictionary.get(key, default)`
    - `default` is returned if `key` is not found
  - Alternative to `[]` operator
    - Cannot raise `KeyError` exception

## Some Dictionary Methods (cont'd.)

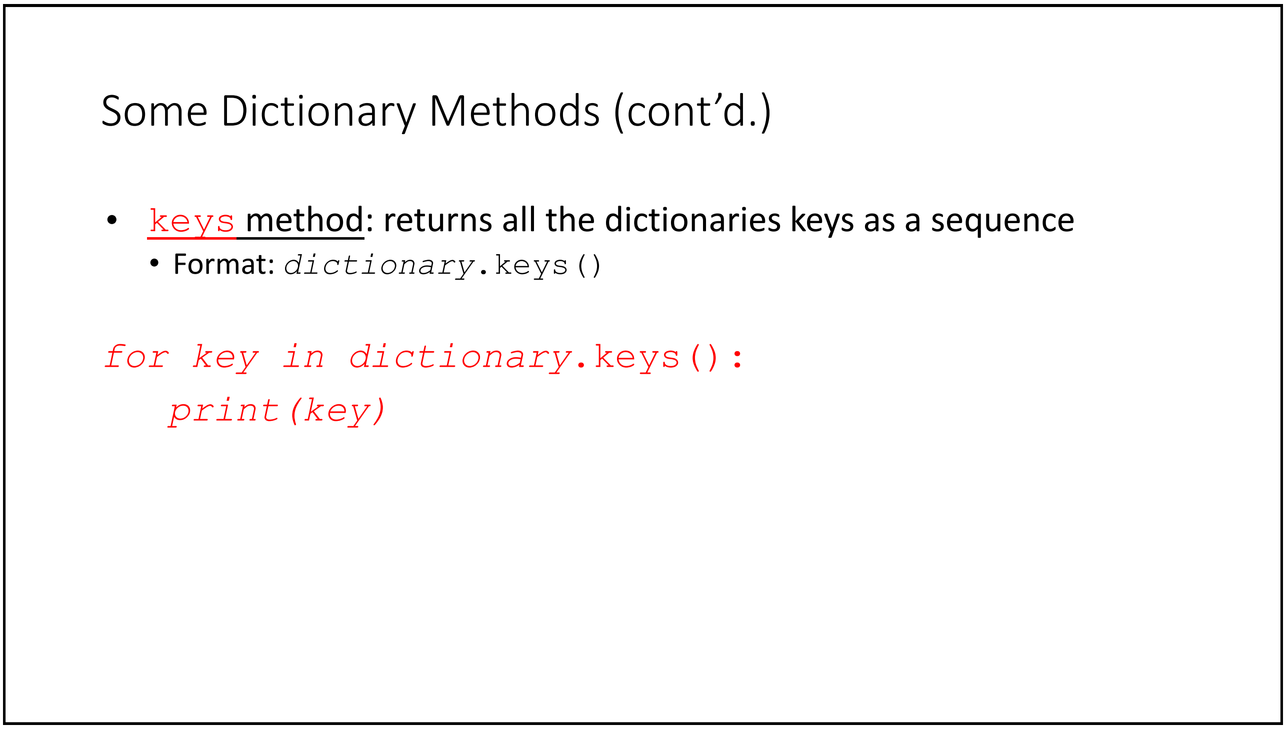
- `items` method: returns all the dictionaries keys and associated values
  - Format: `dictionary.items()`
  - Returned as a ***dictionary view***
    - Use a `for` loop to iterate over the tuples in the sequence
      - Can use a variable which receives a tuple, or can use two variables which receive key and value



## .items() method

```
for pair in dictionary.items():
   print(pair[0],pair[1])


for key, value in dictionary.items():
   print(key,value)
```



## Some Dictionary Methods (cont'd.)

- keys method: returns all the dictionaries keys as a sequence
  - Format: *dictionary*.keys()

```
for key in dictionary.keys():
   print(key)
```

## Some Dictionary Methods (cont'd.)

- `values` method: returns all the dictionaries values as a sequence
  - Format: `dictionary.values()`
  - Use a `for` loop to iterate over the values

```
for value in dictionary.values():
 print(value)
```

## Some Dictionary Methods (cont'd.)

- `pop` method: returns value associated with specified key and removes that key-value pair from the dictionary
  - Format:

```
value = dictionary.pop(key, 'Error MSG')
```

  - `Error MSG` is returned if `key` is not found

# Some Dictionary Methods (cont’d.)

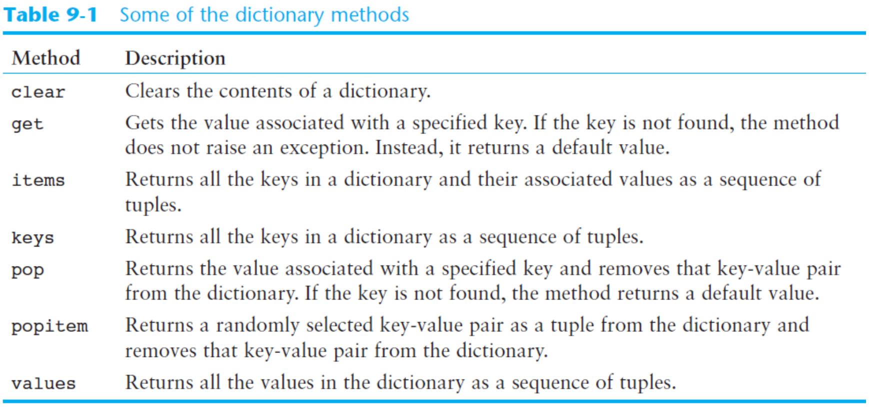


**Table 9-1** Some of the dictionary methods

| Method | Description |
|---|---|
| `clear` | Clears the contents of a dictionary. |
| `get` | Gets the value associated with a specified key. If the key is not found, the method does not raise an exception. Instead, it returns a default value. |
| `items` | Returns all the keys in a dictionary and their associated values as a sequence of tuples. |
| `keys` | Returns all the keys in a dictionary as a sequence of tuples. |
| `pop` | Returns the value associated with a specified key and removes that key-value pair from the dictionary. If the key is not found, the method returns a default value. |
| `popitem` | Returns a randomly selected key-value pair as a tuple from the dictionary and removes that key-value pair from the dictionary. |
| `values` | Returns all the values in the dictionary as a sequence of tuples. |

# Summary

- We covered:
  - Dictionaries, including:
    - Creating dictionaries
    - Inserting, retrieving, adding, and deleting key-value pairs
    - `for` loops and `in` and `not in` operators
    - Dictionary methods