INLS 560
Programming for Information Professionals

Dictionaries

UNC
SCHOOL OF INFORMATION
AND LIBRARY SCIENCE

Joan Boone
jpboone@email.unc.edu

Part 1: Overview, Functions, Iteration

Part 2: Example and Exercise

Dictionaries

- Dictionaries are objects that store a collection of data.
- Each element of a dictionary has 2 parts: a <u>key</u> and a <u>value</u>
- Creating a dictionary

```
state_capitals = {'Virginia' : 'Richmond',
                  'Alaska' : 'Juneau',
                  'Maine' : 'Augusta'}
```

- Get a <u>value</u> from the dictionary

```
state_capitals['Alaska']     →    'Juneau'

state_capitals['Unknown']    →    KeyError exception
```

- Finding a key in a dictionary using **in** operator

```
if 'Maine' in state_capitals:
    print(state_capitals['Maine'])    →    'Augusta'
```

Specify a <u>key</u> to find its <u>value</u>

Dictionary Functions

- Adding a new key-value pair to a dictionary

```
state_capitals['Ohio'] = 'Columbus'
```

  If the key already exists in the dictionary, its value will be updated

- Deleting a key-value pair from a dictionary

```
del state_capitals['Ohio']
```

- Getting the number of items in a dictionary (length)

```
number_of_capitals = len(state_capitals)
```

- Creating an empty dictionary: 2 ways

  a) `state_capitals = {}`

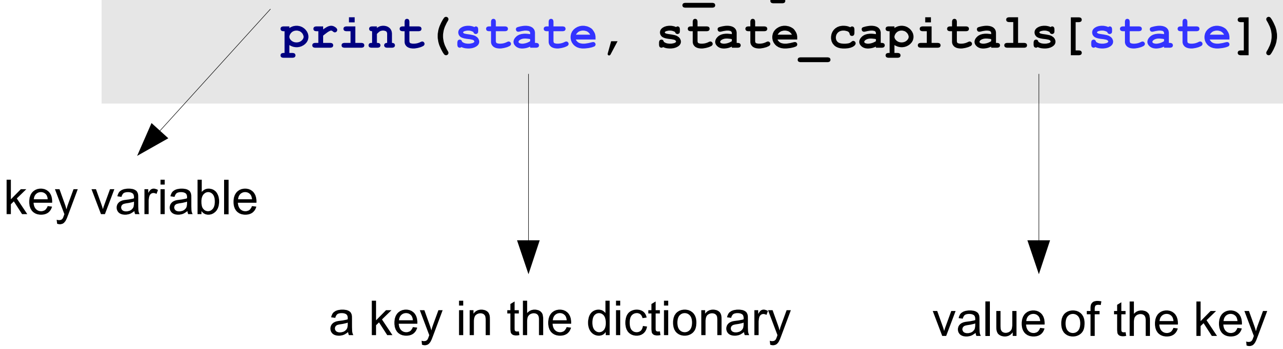
  b) `state_capitals = dict()`   (uses a built-in method)

Iterating over a Dictionary: for loop

- Each time the loop iterates, the next dictionary key is assigned to the key variable, state

```
state_capitals = {'Virginia' : 'Richmond',
                  'Alaska' : 'Juneau',
                  'Maine' : 'Augusta',
                  'Ohio' : 'Columbus'}

for state in state_capitals:
    print(state, state_capitals[state])
```



key variable

a key in the dictionary

value of the key

Output

```
The capital of Virginia is Richmond
The capital of Alaska is Juneau
The capital of Maine is Augusta
The capital of Ohio is Columbus
```

dictionary_loop.py

More Dictionary Functions

Method	Description
`clear`	Clears the contents of a dictionary
`get`	Gets the value associated with a specified key
`items`	Returns all keys and their values as a sequence of tuples
`keys`	Returns all 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
`popitem`	Returns a key-value pair from the dictionary and removes that key-value pair from the dictionary. Pairs are returned in LIFO (Last-In-First-Out) order. *Changed in version 3.7.*
`values`	Returns all values in the dictionary as a sequence of tuples

Source: Starting Out with Python by Tony Gaddis

Python documentation

Part 1: Overview, Functions, Iteration

Part 2: Example and Exercise

NCAA Basketball Champions

- Read a file containing NCAA Basketball Champions from 1939-2022 and create a dictionary, called `champions`
- `champions` contains keys that are the years, and each key's associated value is the name of the championship team
- Prompt the user for a year in the range 1939-2022 and display the name of the championship team

`champions`

Year	Team
1939	Oregon
1940	Indiana
1941	Wisconsin
...	...

Sample input and output:

```
Enter a year between 1939 and 2022 to find the NCAA Basketball
Champion: 2009
North Carolina won the NCAA Basketball Championship in 2009
```

NCAA_BB_Champions.py,   NCAA_BB_Champions.txt

Exercise: Update NCAA Basketball Champions program to include number of wins

- When reading the file containing NCAA Basketball Champions from 1939-2022, create a <u>second</u> dictionary, `champion_wins`
- `champion_wins` contains keys for each championship team, and each key's associated value is the number of wins
- For the team that won the championship in the user-specified year, also display the number of overall wins

`champion_wins`

Team	**Wins**
Duke	5
UCLA	11
Villanova	3
...	...

Sample input and output:

```
Enter a year between 1939 and 2022 to find the NCAA Basketball
Champion: 2018
Villanova won the NCAA Basketball Championship in 2018
Villanova won the NCAA Basketball Championship 3 times
```

Lists or Dictionaries, which to use ?

It depends, but some general guidelines...

- Use lists when
  - Data must be ordered, e.g., each list item represents a day of the week, or a month of the year
  - Data can be changed (mutable)
  - Data does not have to be unique
  - You need a simple collection of data that has no relationships with other data
- Use dictionaries when
  - Need a logical association between a unique key and a value
  - Need fast lookup of data