Introduction to Dictionaries

Python Dictionaries

Traditional dictionaries contain key-value pairs of words and definitions for them

In key-value pairs, a unique key is used as the index for a value

A Python dictionary is an ordered collection of key-value pairs

Dictionary syntax

Denoted by a pair of curly braces, can be empty

If not empty, delimit key-value pairs with colon, separate pairs with commas

Keys must be unique, duplicate keys will overwrite previous key

Keys can be of any immutable data type, including integers, floats, strings, Boolean

Most commonly strings or integers

Syntax:

{key: value, key: value, ...}

Empty dictionary example:

user_settings = {}

Dictionary syntax

Syntax:

{key: value, key: value, ...}

Examples:

user_settings = {“lang”: “en-us”, “platform”: “Windows 10”}

member_levels = {1: “Gold”, 2: “Silver”, 3: “Bronze”}

Duplicate keys

Keys must be unique, duplicate keys will overwrite previous key

Example:
member_levels = {1: “Gold”, 2: “Silver”, 3: “Bronze”, 3: “Copper”}

Dictionary order and Indexing

A Python dictionary is an ordered collection of key-value pairs

Dictionaries have insertion order since Python 3.6 (2016)

Dictionaries are indexed by their unique keys, unlike lists and strings which are automatically indexed by numbers

Access values using keys in bracket notation

Dictionary order and Indexing

Example:

```
user_settings = {“lang”: “en-us”, “platform”: “Windows 10”}

print(user_settings[“lang”])
```

Result: The string “en-us” is printed

More about Dictionaries

Dictionary values can be of any data type, including composite data types

Dictionary values can be duplicates, as long as keys are unique

Dictionaries are mutable – you can change the values

More about Dictionaries

Example:

orders = {5: “bbq”, 2: “tacos”, 9: “pizza”}

orders[9] = “tacos”

Result - The orders variable now contains:
{5: “bbq”, 2: “tacos”, 9: “tacos”}

Dictionaries recap

An ordered collection of key-value pairs

Denoted with curly braces, each key and value is delimited by a colon, key-value pairs are comma-separated

Keys can be of any immutable data type, most often strings and integers

Keys must be unique, duplicates will be overwritten

Keys are used as indexes to access values, using bracket notation

Values can be any data type, can be duplicates, and are mutable