

Lecture 05: Data Structures: Dictionaries and Tuples

February 27, 2023



1 Python Data Structures: Dictionaries & Tuples

Ahmed Moustafa

2 Dictionaries

- A dictionary is an unordered collection of key-value pairs.
- Each key is unique and maps to a value.
- The syntax for creating a dictionary is:

```
[2]: my_dict = {"key1": "value1", "key2": "value2", "key3": "value3"}  
my_dict
```

```
[2]: {'key1': 'value1', 'key2': 'value2', 'key3': 'value3'}
```

You can access values in a dictionary using their keys:

```
[3]: print(my_dict["key1"])
```

```
value1
```

3 Example

```
[4]: states = {}                                # Create empty dict  
states['ca'] = 'California'                     # 1. Set key/value pairs into dict  
states['ok'] = 'Oklahoma'  
states['nj'] = 'New Jersey'  
states['tx'] = 'Texas'  
states
```

```
[4]: {'ca': 'California', 'ok': 'Oklahoma', 'nj': 'New Jersey', 'tx': 'Texas'}
```

```
[5]: type(states)
```

```
[5]: dict
```

```
<IPython.core.display.Image object>
```

Source: [Nick Parlante's Python guide](#)

4 Dictionary commonly used functions

- `keys()`: Returns a list of all the keys in the dictionary
- `values()`: Returns a list of all the values in the dictionary
- `items()`: Returns a list of all the key-value pairs in the dictionary

```
[7]: phonebook = {"Alice": "555-1234", "Bob": "555-5678", "Charlie": "555-9012"}  
phonebook
```

```
[7]: {'Alice': '555-1234', 'Bob': '555-5678', 'Charlie': '555-9012'}
```

```
[8]: print(phonebook["Alice"])
```

```
555-1234
```

```
[9]: print(phonebook.keys())
```

```
dict_keys(['Alice', 'Bob', 'Charlie'])
```

```
[10]: print(phonebook.values())
```

```
dict_values(['555-1234', '555-5678', '555-9012'])
```

```
[11]: print(phonebook.items())
```

```
dict_items([('Alice', '555-1234'), ('Bob', '555-5678'), ('Charlie',  
'555-9012')])
```

```
[12]: phonebook["John"]
```

```
-----  
KeyError
```

```
Traceback (most recent call last)
```

```
Cell In[12], line 1
```

```
----> 1 phonebook["John"]
```

```
KeyError: 'John'
```

5 Tuples

- A tuple is an ordered collection of values.
- Tuples are immutable, meaning you can't modify their values once they're created.
- The syntax for creating a tuple is:

```
[ ]: my_tuple = ("value1", "value2", "value3")
my_tuple
```

You can access values in a tuple using their index:

```
[ ]: print(my_tuple[0])
```

6 Tuples commonly used functions:

- `count()`: Returns the number of times a value appears in the tuple
- `index()`: Returns the index of the first occurrence of a value in the tuple

```
[ ]: days_of_week = ("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
    ↪ "Saturday", "Sunday")
days_of_week
```

```
[ ]: print(days_of_week[0])
```

```
[ ]: print(days_of_week.count("Monday"))
```

```
[ ]: print(days_of_week.index("Wednesday"))
```

7 Differences between Dictionaries and Tuples

- Dictionaries and tuples are used for different purposes.
- Dictionaries are used to *map* **unique** keys to values, while tuples are used to *store* collections of values.
- Dictionaries are **mutable**, while tuples are **immutable**.
- Dictionaries are **unordered**, while tuples are **ordered**.

8 More Example

Storing information about a person, such as their name, address, and phone number:

```
[ ]: person = {"name": "Jane Doe", "address": "123 Main St", "phone": "555-1234"}
person
```

Storing a dictionary of translations between different languages:

```
[ ]: translations = {"hello": {"spanish": "hola", "french": "bonjour"}}
translations
```

Representing a point in two-dimensional space:

```
[ ]: point = (3, 5)
      point
```

Storing the RGB color values for a particular color:

```
[ ]: color = (255, 128, 0)
      color
```

9 Exercise: Factorial

Factorial is a mathematical function that is represented by an exclamation mark (!). It is used to find the product of all positive integers from 1 to a given number. For example, the factorial of 5 (represented as 5!) is calculated as $5 \times 4 \times 3 \times 2 \times 1$, which equals 120. In other words, factorial is the multiplication of all positive integers up to a given number.

Write a program that calculates the factorial of 5.

```
[ ]: # Solution: Factorial
```