CoGrammar

Managing Lists & Dictionaries

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Lecture Objectives

1. Distinguish between the functionality of a Lists and Dictionaries.

2. Expand on key operations relevant to Dictionaries.

3. Apply the above knowledge to improve data management in programs

Question:

What is a 2D list, and how does it differ from a 1D list?

Dictionaries

- ★ In Python, dictionaries function akin to the dictionaries we commonly used in English class, such as those from Oxford.
- ★ Python dictionaries are similar to a list, however each item has two parts, a key and a value.
- ★ To draw a parallel, consider an English dictionary where the key represents a word, and the associated value is its definition.

Dictionary Example

```
# Dictionary Example

my_dictionary = {
    "name": "Terry",
    "age": 24,
    "is_funny": False
}
```

Dictionaries are enclosed in curly brackets; key value pairs are separated by colon and each pair is separated by a comma.
On the left is the key, on the right is the value.

Dict. Functions

- The dict() function in Python is a versatile way to create dictionaries.
- Create dictionaries through assigning values to keys by passing in keys and values separated by an **=** sign.

```
# Creating a dictionary with direct key-value pairs
my_dict = dict(name="Kitty", age=25, city="Belarus")
print(my_dict)
# Output: {'name': 'Kitty', 'age': 25, 'city': 'Belarus'}
```

Question:

Why won’t the append() method work on the Dictionary data structure?

Dict Update

- To append or add elements to a dictionary in Python, you can use the update() method or simply use the square bracket notation.

```
my_dict = dict(name="Kitty", age=25, city="Belarus")
# Adding or updating a key-value pair
my_dict.update({'breed': 'Shorthair'})
print(my_dict)
# Output: {'name': 'Kitty', 'age': 25, 'city': 'Belarus', 'breed': 'Shorthair'}
```

```
my_dict = dict(name="Kitty", age=25, city="Belarus")
# Adding or updating a key-value pair
my_dict['breed'] = 'Shorthair'
print(my_dict)
# Output: {'name': 'Kitty', 'age': 25, 'city': 'Belarus', 'breed': 'Shorthair'}
```

Dictionary Cont.

- To access a value in a dictionary, we simply call the key and Python will return the value paired with said key.
- Similar to indexing, however we provide a key name instead of an index number.

```
my_dict = dict(name="Kitty", age=25, city="Belarus")
name = my_dict["name"]
# Output: 'Kitty'
```

Dictionary Operations

- **Key-Value Pairs**
  - items()
- **Fetching**
  - get()
- **Updating**
  - update()
- **Deleting**
  - pop() / popitem()

Challenge:

Which of the following code snippets will update the cat's age correctly ?

A. `cat_info['age'] += 1`

B. `cat_info.update({'age': 366})`

C. `cat_info['age'] = 366`

D. `cat_info['age'] = cat_info['age'] + 1`

Poll:

Assessment

Wrapping Up

Lists

2D lists in Python offer a powerful mechanism for organizing and manipulating data in a structured manner.

Dictionaries

Rows represent individual lists within the main list, while columns denote elements within each of these lists.

CoGrammar

Questions around Operations on Lists and Dictionaries

CoGrammar

Thank you for joining