Python Dictionaries and Frequency Tables: Takeaways

Syntax

- Check if a certain value exists in the dictionary as a key:

```
dictionary = {'key_1': 100, 'key_2': 200}
'key_1' in dictionary  # Outputs True
'key_5' in dictionary  # Outputs False
100 in dictionary  # Outputs False
```

- Use the in operator to check for dictionary membership:

```
content_ratings = {'4+': 4433, '9+': 987, '12+': 1155, '17+': 622}
print('12+' in content_ratings)
```

- Update dictionary values:

```
dictionary = {'key_1': 100, 'key_2': 200}
dictionary['key_1'] += 600  # This will change the value to 700
```

- Create a frequency table for the unique values in a column of a dataset:

```
frequency_table = {}
for row in a_data_set:
    a_data_point = row[5]
    if a_data_point in frequency_table:
        frequency_table[a_data_point] += 1
    else:
        frequency_table[a_data_point] = 1
```

- Loop over a dictionary:

```
content_ratings = {'4+': 4433, '9+': 987, '12+': 1155, '17+': 622}
for iteration_variable in content_ratings:
    print(iteration_variable)
```

- Compute the frequency for defined intervals:

```
data_sizes = {'0 - 10 MB': 0, '10 - 50 MB': 0, '50 - 100 MB': 0,
                    '100 - 500 MB': 0, '500 MB +': 0}

for row in apps_data[1:]:
    data_size = float(row[2])
    if data_size <= 10000000:
        data_sizes['0 - 10 MB'] += 1
    elif 10000000 < data_size <= 50000000:
        data_sizes['10 - 50 MB'] += 1
    elif 50000000 < data_size <= 100000000:
        data_sizes['50 - 100 MB'] += 1
    elif 100000000 < data_size <= 500000000:
        data_sizes['100 - 500 MB'] += 1
    elif data_size > 500000000:
        data_sizes['500 MB +'] += 1
print(data_sizes)
```

Concepts

- We can check if a certain value exists in the dictionary as a key using an `in` operator. An `in` expression always returns a Boolean value.
- We also call the number of times a unique value occurs the **frequency**. We call tables that map unique values to their frequencies **frequency tables**.
- When we iterate over a dictionary with a `for` loop, we loop over the dictionary keys by default.

Resources

- Dictionaries in Python