Tutorial 12

Date: March 7, 2021

Email: mohdakram.ansari@ucalgary.ca

Agenda

1. Dictionaries
2. Lists

1. Dictionaries

Creating Python Dictionary

Creating a dictionary is as simple as placing items inside curly braces {} separated by commas.

```
# empty dictionary
my_dict = {}

# dictionary with integer keys
my_dict = {1: 'apple', 2: 'ball'}

# dictionary with mixed keys
my_dict = {'name': 'John', 1: [2, 4, 3]}

# using dict()
my_dict = dict({1:'apple', 2:'ball'})

# from sequence having each item as a pair
my_dict = dict([(1,'apple'), (2,'ball')])
```

Accessing Elements from Dictionary

Keys can be used either inside square brackets [] or with the get() method.

```
# get vs [] for retrieving elements
my_dict = {'name': 'Jack', 'age': 26}

# Output: Jack
print(my_dict['name'])
```

```
# Output: 26
print(my_dict.get('age'))

# Trying to access keys which doesn't exist throws error
# Output None
print(my_dict.get('address'))

# KeyError
print(my_dict['address'])
```

Changing and Adding Dictionary elements

Dictionaries are mutable. We can add new items or change the value of existing items using an assignment operator.

```
# Changing and adding Dictionary Elements
my_dict = {'name': 'Jack', 'age': 26}

# update value
my_dict['age'] = 27

#Output: {'age': 27, 'name': 'Jack'}
print(my_dict)

# add item
my_dict['address'] = 'Downtown'

# Output: {'address': 'Downtown', 'age': 27, 'name': 'Jack'}
print(my_dict)
```

Checking if dictionary has a key

Use the `in` and `not in` operator to check if dictionary has a key.

```
numbers = {1: "one", 2: "two", 3: "three", 4: "four"}

print("one" in numbers)
print("one" not in numbers)

print(3 in numbers)
print(3 not in numbers)

print(5 in numbers)
print(5 not in numbers)
```

Exercise

Make a program to count the number of each letter in a word entered by the user.
Print the frequency as a dictionary

2. Lists

A list in python is a sequence of zero or more elements of any data type (including lists itself).

```
# a list of programming languages
['Python', 'C++', 'JavaScript']
```

Create Python Lists

A list is created by placing elements inside square brackets [], separated by commas.

```
# list of integers
my_list = [1, 2, 3]

# empty list
my_list = []

# list with mixed data types
my_list = [1, "Hello", 3.4]

# nested list
my_list = ["mouse", [8, 4, 6], ['a']]
```

Access List Elements

We can use the index operator [] to access an item in a list. In Python, indices start at 0.

```
my_list = ['p', 'r', 'o', 'b', 'e']

# first item
print(my_list[0])  # p

# third item
print(my_list[2])  # o

# fifth item
print(my_list[4])  # e

# Nested List
nested_list = [["queen", "king", "joker"], [2, 0, 1, 5]]

# Nested indexing
print(nested_list[0][1])
```

```
print(nested_list[1][3])

# Error! Only integer can be used for indexing
print(my_list[4.0])
```

Add/Change List Elements

We can use the assignment operator = to change an item.

```
# Correcting mistake values in a list
odd = [2, 4, 6, 8]

# change the 1st item
odd[0] = 1
# odd: [1, 4, 6, 8]

# Add an item
odd.append([999, 55])
# odd: [1, 4, 6, 8, [999, 55]]
```

Iterating Through a List

```
fruits = ['apple','banana','mango']
for fruit in fruits:
    print(fruit)

length = len(fruits)
for i in range(len(fruits)):
  print(fruits[i])
```
