

# Python Dictionaries Medical Insurance

October 16, 2024

## 1 Python Dictionaries: Medical Insurance Project

You have been asked to create a program that organizes and updates medical records efficiently.

In this project, you will use your new knowledge of Python dictionaries to create a database of medical records for patients.

Let's get started!

### 1.1 Storing Patient Names and Insurance Costs

1. We would like to keep a record of medical patients and their insurance costs.

First, create an empty dictionary called `medical_costs`.

```
[41]: medical_costs = {}
```

2. Let's populate our `medical_costs` dictionary by adding the following key-value pairs:

- Add "Marina" to `medical_costs` as a key with a value of 6607.0.
- Add "Vinay" to `medical_costs` as a key with a value of 3225.0.

```
[3]: medical_costs['Marina'] = 6607.0  
medical_costs['Vinay'] = 3225.0
```

3. Using one line of code, add the following three patients to the `medical_costs` dictionary:

- "Connie", with an insurance cost of 8886.0.
- "Isaac", with an insurance cost of 16444.0.
- "Valentina", with an insurance cost of 6420.0.

```
[43]: medical_costs.update({'Connie': 8886.0, 'Isaac': 16444.0, 'Valentina': 6420.0})
```

4. Print `medical_costs`. Make sure the dictionary is what you expected.

```
[45]: print(medical_costs)
```

```
{'Connie': 8886.0, 'Isaac': 16444.0, 'Valentina': 6420.0}
```

5. You notice that Vinay's insurance cost was incorrectly inputted. Update the value associated with Vinay to 3325.0.

Print the updated dictionary.

```
[47]: medical_costs['Vinay'] = 3325.0
      print(medical_costs)
```

```
{'Connie': 8886.0, 'Isaac': 16444.0, 'Valentina': 6420.0, 'Vinay': 3325.0}
```

6. Let's calculate the average medical cost of each patient. Create a variable called `total_cost` and set it equal to 0.

Next, iterate through the values in `medical_costs` and add each value to the `total_cost` variable.

```
[49]: total_cost = 0.0

      for cost in medical_costs.values():
          total_cost += cost
```

7. After the loop, create a variable called `average_cost` that stores the `total_cost` divided by the length of the `medical_costs` dictionary.

Print `average_cost` with the following message:

Average Insurance Cost: {average\_cost}

```
[51]: average_cost = total_cost / (len(medical_costs))
      print(f'Average Insurance Cost: {average_cost}')
```

Average Insurance Cost: 8768.75

## 1.2 List Comprehension to Dictionary

8. You have been asked to create a second dictionary that maps patient names to their ages.

First, create two lists called `names` and `ages` with the following data:

names	ages
Marina	27
Vinay	24
Connie	43
Isaac	35
Valentina	52

```
[53]: names = ['Marina', 'Vinay', 'Connie', 'Isaac', 'Valentina']

      ages = [27, 24, 43, 35, 52]
```

9. Next, create a variable called `zipped_ages` that is a zipped list of pairs between the `names` list and the `ages` list.

```
[55]: zipped_ages = zip(names, ages)
```

10. Create a dictionary called `names_to_ages` by using a list comprehension that iterates through `zipped_ages` and turns each pair into a key : value item.

Print `names_to_ages` to see the result.

```
[57]: names_to_ages = {key: value for key, value in zipped_ages}
      print(names_to_ages)
```

```
{'Marina': 27, 'Vinay': 24, 'Connie': 43, 'Isaac': 35, 'Valentina': 52}
```

11. Use `.get()` to get the value of Marina's age and store it in a variable called `marina_age`. Use `None` as a default value if the key doesn't exist.

Print `marina_age` with the following message:

Marina's age is {`marina_age`}

```
[59]: marina_age = names_to_ages.get("Marina", None)
      print("Marina's age is " + str(marina_age))
```

Marina's age is 27

### 1.3 Using a Dictionary to Create a Medical Database

12. Let's create a third dictionary to represent a database of medical records that contains information such as a patient's name, age, sex, gender, BMI, number of children, smoker status, and insurance cost.

First, create an empty dictionary called `medical_records`.

```
[61]: medical_records = {}
```

13. Next, add "Marina" to `medical_records` as a key with the value being a dictionary of medical data:

```
{"Age": 27, "Sex": "Female", "BMI": 31.1, "Children": 2, "Smoker": "Non-smoker", "Insurance Cost": 6607.0}
```

```
[63]: medical_records['Marina'] = {
      'Age': 27,
      'Sex': 'Female',
      'BMI': 31.1,
      'Smoker': 'Non-smoker',
      'Insurance Cost': 6607.0
      }
```

14. Do the same for the following individuals:

Name	Age	Sex	BMI	Children	Smoker	Insurance Cost
Vinay	24	Male	26.9	0	Non-smoker	3225.0
Connie	43	Female	25.3	3	Non-smoker	8886.0
Isaac	35	Male	20.6	4	Smoker	16444.0

Name	Age	Sex	BMI	Children	Smoker	Insurance Cost
Valentina	52	Female	18.7	1	Non-smoker	6420.0

```
[65]: medical_records["Vinay"] = {"Age": 24, "Sex": "Male", "BMI": 26.9, "Children": 0, "Smoker": "Non-smoker", "Insurance Cost": 3225.0}
medical_records["Connie"] = {"Age": 43, "Sex": "Female", "BMI": 25.3, "Children": 3, "Smoker": "Non-smoker", "Insurance Cost": 8886.0}
medical_records["Isaac"] = {"Age": 35, "Sex": "Male", "BMI": 20.6, "Children": 4, "Smoker": "Smoker", "Insurance Cost": 16444.0}
medical_records["Valentina"] = {"Age": 52, "Sex": "Female", "BMI": 18.7, "Children": 1, "Smoker": "Non-smoker", "Insurance Cost": 6420.0}
```

15. Print `medical_records` to see the result.

```
[67]: from pprint import pprint
pprint(medical_records)
```

```
{'Connie': {'Age': 43,
            'BMI': 25.3,
            'Children': 3,
            'Insurance Cost': 8886.0,
            'Sex': 'Female',
            'Smoker': 'Non-smoker'},
 'Isaac': {'Age': 35,
           'BMI': 20.6,
           'Children': 4,
           'Insurance Cost': 16444.0,
           'Sex': 'Male',
           'Smoker': 'Smoker'},
 'Marina': {'Age': 27,
            'BMI': 31.1,
            'Insurance Cost': 6607.0,
            'Sex': 'Female',
            'Smoker': 'Non-smoker'},
 'Valentina': {'Age': 52,
               'BMI': 18.7,
               'Children': 1,
               'Insurance Cost': 6420.0,
               'Sex': 'Female',
               'Smoker': 'Non-smoker'},
 'Vinay': {'Age': 24,
           'BMI': 26.9,
           'Children': 0,
           'Insurance Cost': 3225.0,
           'Sex': 'Male',
           'Smoker': 'Non-smoker'}}
```

16. The `medical_records` dictionary acts like a database of medical records. Let's access a specific piece of data in `medical_records`.

Print out Connie's insurance cost with the following message:

Connie's insurance cost is X dollars.

```
[69]: print("Connie's insurance cost is " + str(medical_records["Connie"]["Insurance_↵Cost"]) + " dollars.")
```

Connie's insurance cost is 8886.0 dollars.

17. Vinay has moved to a new country, and we no longer want to include him in our medical records.

Remove Vinay from `medical_records`.

```
[71]: medical_records.pop('Vinay')
pprint(medical_records)
```

```
{'Connie': {'Age': 43,
            'BMI': 25.3,
            'Children': 3,
            'Insurance Cost': 8886.0,
            'Sex': 'Female',
            'Smoker': 'Non-smoker'},
 'Isaac': {'Age': 35,
           'BMI': 20.6,
           'Children': 4,
           'Insurance Cost': 16444.0,
           'Sex': 'Male',
           'Smoker': 'Smoker'},
 'Marina': {'Age': 27,
            'BMI': 31.1,
            'Insurance Cost': 6607.0,
            'Sex': 'Female',
            'Smoker': 'Non-smoker'},
 'Valentina': {'Age': 52,
               'BMI': 18.7,
               'Children': 1,
               'Insurance Cost': 6420.0,
               'Sex': 'Female',
               'Smoker': 'Non-smoker'}}
```

18. Let's take a closer look at each patient's medical record.

Use a `for` loop to iterate through the items of `medical_records`. For each key-value pair, print out a string that looks like the following:

{Name} is a {Age} year old {Sex} {Smoker} with a BMI of {BMI} and insurance cost of {Insur

```
[73]: for name, record in medical_records.items():
      print(name + " is a " + str(record["Age"]) + \
            " year old " + record["Sex"] + " " + record["Smoker"] \
            + " with a BMI of " + str(record["BMI"]) + \
            " and insurance cost of " + str(record["Insurance Cost"]))
```

Marina is a 27 year old Female Non-smoker with a BMI of 31.1 and insurance cost of 6607.0

Connie is a 43 year old Female Non-smoker with a BMI of 25.3 and insurance cost of 8886.0

Isaac is a 35 year old Male Smoker with a BMI of 20.6 and insurance cost of 16444.0

Valentina is a 52 year old Female Non-smoker with a BMI of 18.7 and insurance cost of 6420.0

## 1.4 Extra

19. Congratulations! In this project, you used Python dictionaries to store and update medical data for individuals.

If you'd like extra practice with dictionaries, here are some suggestions to go further with this project:

- Create a function called `update_medical_records()` that takes in the name of an individual as well as their medical data, and then updates the `medical_records` dictionary accordingly.
- Create a new dictionary of your choice - feel free to get creative!

Happy coding!

[ ]:

[ ]: