quiz-dictionary

September 16, 2024

0.1 QUIZ - Dictionary

Q 1:

Define 4 functions named **car_1, car_2, car_3, car_4**.

These functions will create dictionaries as below (name of the dictionary will be car):

Functions will create the dictionary by different ways and return the dictionary.

Hints: * { } * dict() * update()

[2]:

```python
# Q 1:

# Function 1
# ---- your solution here ----
def car_1():
    car = {
        'brand': 'Ford',
        'model': 'Mustang',
        'year': 1964,
        'color': 'Red',
        'price': 30000,
        'km': 89000,
        'motor': 1.6
    }
    return car
# call the function you defined
print(car_1())


# Function 2
# ---- your solution here ----
def car_2():
    car = dict(
        brand='Ford',
        model='Mustang',
        year=1964,
        color='Red',
        price=30000,
        km=89000,
        motor=1.6
    )
    return car
# call the function you defined
print(car_2())


# Function 3
# ---- your solution here ----
def car_3():
    car = {}
    car.update({'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color':␣
 ↪'Red'})
    car.update({'price': 30000, 'km': 89000, 'motor': 1.6})
    return car
# call the function you defined


print(car_3())
# Function 4
# ---- your solution here ----
def car_4():
    car = dict([
        ('brand', 'Ford'),
        ('model', 'Mustang'),
        ('year', 1964),
        ('color', 'Red'),
        ('price', 30000),
        ('km', 89000),
        ('motor', 1.6)
    ])
    return car
# call the function you defined
print(car_4())
```

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'Red', 'price':
30000, 'km': 89000, 'motor': 1.6}
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'Red', 'price':
30000, 'km': 89000, 'motor': 1.6}
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'Red', 'price':
30000, 'km': 89000, 'motor': 1.6}
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'Red', 'price':
30000, 'km': 89000, 'motor': 1.6}
```

Q 2:

Define a function named **create_a_new_car**.

It will call one of the functions defined in Q3 and will get the car dictionary.

Then it will copy the items of this car dictionary into another dictionary via a loop.

It will first copy all the elements in car dictionary, then create new keys via appending ”_2” at the end of existing key, and create a new element.

Values will be the same.

It will return the new dictionary.

Hint: * copy() * update() * items()

[3]:

```python
# Q 2:

# ---- your solution here ----
def create_a_new_car():
    car = car_1()
    new_car = {}

    for key, value in car.items():
        new_car[key] = value
        new_car[key + "_2"] = value

    return new_car


# call the function you defined
create_a_new_car()
```

[3]:

```
{'brand': 'Ford',
 'brand_2': 'Ford',
 'model': 'Mustang',
 'model_2': 'Mustang',
 'year': 1964,
 'year_2': 1964,
 'color': 'Red',
 'color_2': 'Red',
 'price': 30000,
 'price_2': 30000,
 'km': 89000,
 'km_2': 89000,
 'motor': 1.6,
 'motor_2': 1.6}
```

Q 3:

Define a function named **concat_dicts**.

It will concatenate the dictionaries below and return the resulting dict.

The function will take these dictionaries as parameters.

Hints * use only one for loop * search for looping on multiple dictionaries. Here is an example for statement: **for x in (d1, d2, ...):** * update()

[5]:

```
# Q 3:

# ---- your solution here ----
def concat_dicts(*dicts):
    result = {}
    for d in dicts:
        result.update(d)
    return result

# call the function you defined
d1={4:120, 7:60}
d2={'A': 'AAA', 'B':'BBB'}
d3={True: 'Correct', False: 'Incorrect'}

d = concat_dicts(d1, d2, d3)
print(d)
```

```
{4: 120, 7: 60, 'A': 'AAA', 'B': 'BBB', True: 'Correct', False: 'Incorrect'}
```

Q 4:

Define a function named **delete_odds**.

It will take a dictionary as parameter.

It will delete the items with odd indices from the dictionary and return a new dictionary with remaining items.

Hints: * Do not change original dictionary (parameter) * `items()` for the loop * `enumerate()` for the index

[6]:

```
# Q 4:

# ---- your solution here ----
def delete_odds(dictionary):
    new_dict = {key: value for index, (key, value) in enumerate(dictionary.
 ↪items()) if index % 2 == 0}
    return new_dict
```

```
# call the function you defined
dictionary = {'a': 'A', 'b': 'B', 'c': 'C', 'd': 'D', 'e': 'E', 'f': 'F'}
evens = delete_odds(dictionary)
print(evens)
```

```
{'a': 'A', 'c': 'C', 'e': 'E'}
```

Q 5:

Define a function named **convert_lists_into_dict**.

It will take two lists as parameters.

The function will use the first list elements as Keys and second list elements as Values and it will create a dictionary.

Then it will return this dictionary.

Hints: * enumerate()

[7]:

```
# Q 5:

# ---- your solution here ----
def convert_lists_into_dict(l_1, l_2):
    return {key: value for key, value in zip(l_1, l_2)}


# call the function you defined
l_1 = ['name', 'lastname', 'age', 'gender']
l_2 = ['John', 'Doe', 100, 'Male']
employee = convert_lists_into_dict(l_1, l_2)
print(employee)
```

```
{'name': 'John', 'lastname': 'Doe', 'age': 100, 'gender': 'Male'}
```

Q 6:

Let’s consider a function with keys being both numbers and letters.

Example: {‘a’: ‘A’, ‘b’: ‘B’, 2: 200, ‘d’: ‘D’, 5: 300, ‘f’: ‘F’, 1: 50}

Define a function named **alphabetical**.

It will delete the elements with keys being number.

And it will return the final dictionary which has only alphabetical keys.

Hints: * Mutate the original dictionary that is the parameter * use two loops * `keys()` for loops * `pop()` for delete * to check if alphabetical -> `isalpha()` * keep in mind `isalpha()` is a string (str) function

[8]:

```python
# Q 6:

# ---- your solution here ----
def alphabetical(dictionary):
    keys_to_delete = [key for key in dictionary.keys() if not isinstance(key,␣
 ↪str) or not key.isalpha()]

    for key in keys_to_delete:
        dictionary.pop(key)

    return dictionary

# call the function you defined
dictionary = {'a': 'A', 'b': 'B', 2: 200, 'd': 'D', 5: 300, 'f': 'F', 1: 50}
print("dictionary before calling alphabetical:", dictionary)
alphabetical(dictionary)
print("dictionary after calling alphabetical:", dictionary)
```

```
dictionary before calling alphabetical: {'a': 'A', 'b': 'B', 2: 200, 'd': 'D',
5: 300, 'f': 'F', 1: 50}
dictionary after calling alphabetical: {'a': 'A', 'b': 'B', 'd': 'D', 'f': 'F'}
```

[ ]:

```

```