Python Dictionary - Q&A

1. What is a Python Dictionary?

A dictionary in Python is a collection of key-value pairs, where each key is unique and is used to access its corresponding value.

Example:

```
my_dict = {'name': 'Prem', 'age': 24}
```

2. How can you access a value from a dictionary?

You can access a value by using its key inside square brackets [] or using the .get() method.

Example:

```
my_dict['name']  # Output: 'Prem'

my_dict.get('age')  # Output: 24
```

3. What will happen if you access a non-existent key? Give an example.

If you access a non-existent key using [], it will raise a KeyError.

If you use .get(), it will return None (or a default value if provided).

Example:

```
my_dict = {'name': 'Prem'}

my_dict['age']  # KeyError

my_dict.get('age')  # Output: None
```

4. How do you add or update a key-value pair in a dictionary?

You can add or update a key-value pair by assigning a value to a key.

Example:

```
my_dict['age'] = 24  # Add

my_dict['name'] = 'Sai'  # Update
```

5. How do you remove a key-value pair from a dictionary?

You can use del or pop() to remove a key-value pair.

Example:

```
del my_dict['name']

my_dict.pop('age')
```

6. How do you get all the keys, values, or items in a dictionary?

- Keys: my_dict.keys()

- Values: my_dict.values()

- Items: my_dict.items()

Example:

```
my_dict = {'name': 'Prem', 'age': 24}

my_dict.keys()  # dict_keys(['name', 'age'])

my_dict.values()  # dict_values(['Prem', 24])

my_dict.items()  # dict_items([('name', 'Prem'), ('age', 24)])
```