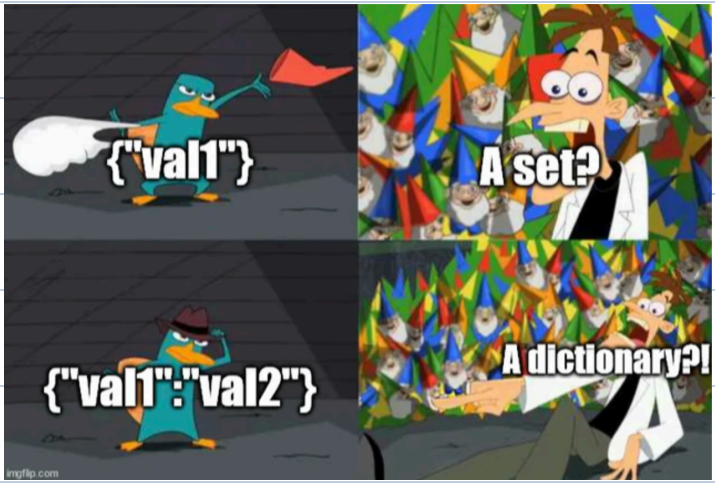
Agenda

1. Motivation behind Dictionaries
2. Creating a Dictionary
3. Properties of Dictionaries
4. Iterating over Dictionaries
5. Sorting a Dictionary



Real Life Examples

1) Phone book ⟶ Name & Ad. : Contact Number
2) Hotel Register ⟶ Room No. : Room & Guest Details
3) Oxford Dictionary → Word : Meaning

1) Phone book → Name & Ad. : Contact Number

2) Hotel Register → Room No. : Room & Guest Details

3) Oxford Dictionary → Word : Meaning

Unique | Useful information about Key

Key — Value

[ ] → List

( ) → Tuple

{ } → Set / Dictionary

{1, 2, 3}

{ 1: "A", 2: "B", 3: "C" }

Key : Value

Properties of Dictionary

1) Update
2) Add
3) Get
4) Remove

Iterating over dictionary

Code:

```
a = {'name': 'Thor Odinson',
     'age': 1500,
     'weapon': ['mjonir', 'stormbreaker'],
     'strongest': True}
```

```
for i in a:
    print(i)
```

Output

```
name
age
weapon
strongest
```

```
for i in a:
    print(a[i])
```

Output

```
Thor Odinson
1500
['mjonir', 'stormbreaker']
True
```

```
for i in a:
    print(f"{i} -> {a[i]}")
```

Output

```
name -> Thor Odinson
age -> 1500
weapon -> ['mjonir', 'stormbreaker']
strongest -> True
```

1) .keys()

2) .values()

3) .items()

How do we check if a key exists within a dictionary?

- We can simply use the `in` membership operator to achieve this.

Can we have object of any datatype as a key in a dictionary?

- While discussing sets, we talked about how we cannot have **sets**, **dictionaries** and **lists** as elements of sets because they are **mutable**.

- This is the exact same case with dictionaries.
    - **Values** - Can store any data structure or any data type.
    - **Keys** - Lists, Sets and Dictionaries are not allowed.

Sorting in Dictionaries

Code:

```
my_dict = {'banana': 3, 'apple': 2, 'orange': 5, 'grape': 1}
print(my_dict)
```

Output

```
{'banana': 3, 'apple': 2, 'orange': 5, 'grape': 1}
```

Sort the dictionary by keys -

Code:

typecast back to dictionary
sort list wrt K
list of tuples (K, V)

```
sorted_dict_keys = dict(sorted(my_dict.items()))
print("Sorted by keys:", sorted_dict_keys)
```

Output

```
Sorted by keys: {'apple': 2, 'banana': 3, 'grape': 1, 'orange': 5}
```

Sort the dictionary by values -

Code:

typecasting
list of tuple (K, V)
0 1

```
sorted_dict_values = dict(sorted(my_dict.items(), key=lambda item: item[1]))
print("Sorted by values:", sorted_dict_values)
```

Output

```
Sorted by values: {'grape': 1, 'apple': 2, 'banana': 3, 'orange': 5}
```

Question

- Take a string as input.
- Create a dictionary according to the following criteria :-
  - There will be one key-value pair for each unique character.
  - Key will be the character name.
  - Value will be the count of the character inside the string.

Example Input:

```
rrssssttt
```

Example Output:

```
{
    "r": 2,
    "s": 3,
    "t": 4
}
```