Dictionary Methods

Ali Moustafa
@ali-moustafa2000

clear()

The "clear()" method removes all items from the dictionary.

```python
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}
my_dict.clear()

print(my_dict)
# Output: {}
```

copy()

The "copy()" method returns a shallow copy of the dictionary.

NB: changes to the copy of the dictionary do not affect the original dictionary.

```python
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}
new_dict = my_dict.copy()

new_dict['age'] = 30

print(my_dict)
# Output: {'name': 'John', 'age': 25, 'city': 'New York'}

print(new_dict)
# Output: {'name': 'John', 'age': 30, 'city': 'New York'}
```

@ali-moustafa2000

get()

The "get()" method returns the value of the specified key. If the key is not present in the dictionary, it will return "None" or a default value specified as the second argument of the method.

```python
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}

age = my_dict.get('age')
print(age)  # Output: 25

# trying to get a value for a key that does not exist
occupation = my_dict.get('occupation')
print(occupation)  # Output: None

# setting a default value if the key does not exist
occupation = my_dict.get('occupation', 'unemployed')
print(occupation)  # Output: 'unemployed'
```

items()

The "items()" method returns a list of key-value pairs in the dictionary.

```python
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}
items = my_dict.items()

print(items)
# Output: dict_items([('name', 'John'), ('age', 25), ('city', 'New York')])
```

@ali-moustafa2000

keys()

The "keys()" method returns a list of keys in the dictionary.

```python
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}
keys = my_dict.keys()

print(keys)
# Output: dict_keys(['name', 'age', 'city'])
```

values()

The "values()" method Returns a list of values in the dictionary.

```python
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}

values = my_dict.values()

print(values)
# Output: dict_values(['John', 25, 'New York'])
```

@ali-moustafa2000

pop()

The "pop()" method removes and returns the value of the specified key. If the key does not exist, it raises a KeyError.
To avoid this, you can pass a default value to be returned if the key is not found.

```python
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}
age = my_dict.pop('age')

print(age)  # Output: 25
print(my_dict)  # Output: {'name': 'John', 'city': 'New York'}

country = my_dict.pop('country')
print(country)  # Output: KeyError: 'country'

country = my_dict.pop('country', 'USA')
print(country)  # Output: USA
```

@ali-moustafa2000

popitem()

The "popitem()" method removes and returns the last inserted key-value pair in the dictionary. If the dictionary is empty, it raises a KeyError.

```python
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}

# Before popitem()
print(my_dict)
# Output: {'name': 'John', 'age': 25, 'city': 'New York'}

# Removing and returning the last inserted key-value pair
popped_item = my_dict.popitem()
print(popped_item)
# Output: ('city', 'New York')

# After popitem()
print(my_dict)
# Output: {'name': 'John', 'age': 25}
```

@ali-moustafa2000

setdefault()

The "setdefault()" method returns the value of the specified key.
If the key does not exist, it inserts the key with the specified default value.

```python
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}

country = my_dict.setdefault('country', 'USA')

print(country)
# Output: USA

print(my_dict)
# Output: {'name': 'John', 'age': 25, 'city': 'New York', 'country': 'USA'}
```

@ali-moustafa2000

update()

The "update()" method updates the dictionary with the specified key-value pairs.

```python
my_dict = {'name': 'John', 'age': 25, 'city': 'New York'}

my_dict.update({'age': 30, 'country': 'USA'})

print(my_dict)
# Output: {'name': 'John', 'age': 30, 'city': 'New York', 'country': 'USA'}
```