

WHAT ARE PYTHON DICTIONARIES?

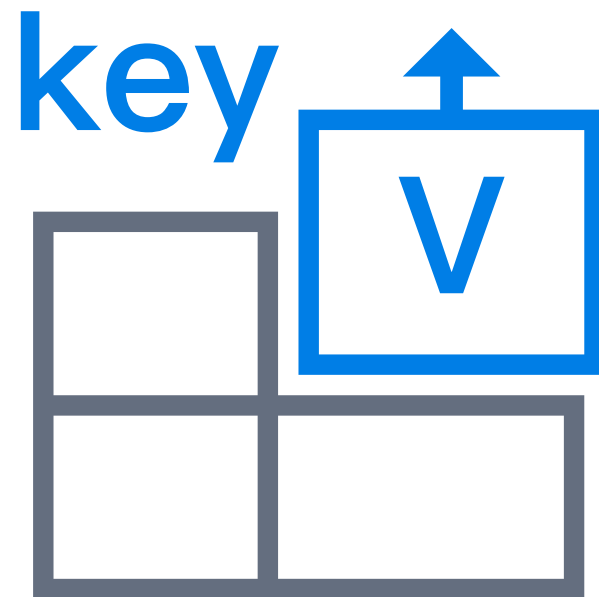


DICTIONARIES IN PYTHON ARE COLLECTIONS OF KEY-VALUE PAIRS. THINK OF THEM AS A REAL-WORLD DICTIONARY WHERE EACH WORD (KEY) HAS A DEFINITION (VALUE).



HOW TO CREATE A DICTIONARY

DICTIONARIES ARE CREATED WITH CURLY BRACES {} AND CONSIST OF KEY-VALUE PAIRS SEPARATED BY A COLON : .



```
my_dict = {  
    'name': 'Alice',  
    'age': 30,  
    'city': 'New York'  
}
```



ACCESS VALUES

RETRIEVE VALUES
USING THEIR KEYS. USE
MY_DICT[KEY] TO
ACCESS A VALUE.



```
name = my_dict['name'] # Alice
```



ADD OR UPDATE ITEMS

ADD NEW KEY-VALUE
PAIRS OR UPDATE
EXISTING ONES BY
ASSIGNING A VALUE TO
A KEY.



```
my_dict['email'] = 'alice@example.com' # Adds a new key-value pair  
my_dict['age'] = 31 # Updates the existing value
```



REMOVE ITEMS

REMOVE ITEMS USING
THE DEL KEYWORD OR
THE POP() METHOD.



```
del my_dict['city']    # Removes 'city'  
email = my_dict.pop('email') # Removes 'email' and returns its value
```



DICTIONARY METHODS

DICTIONARIES COME WITH USEFUL METHODS LIKE KEYS(), VALUES(), AND ITEMS().



```
keys = my_dict.keys() # dict_keys(['name', 'age'])
values = my_dict.values() # dict_values(['Alice', 31])
items = my_dict.items() # dict_items([('name', 'Alice'), ('age', 31)])
```



NESTED DICTIONARIES

DICTIONARIES CAN
CONTAIN OTHER
DICTIONARIES,
CREATING NESTED
STRUCTURES.



```
nested_dict = {  
  'person': {  
    'name': 'Bob',  
    'age': 25  
  },  
  'city': 'Los Angeles'  
}
```

LOOP THROUGH DICTIONARIES

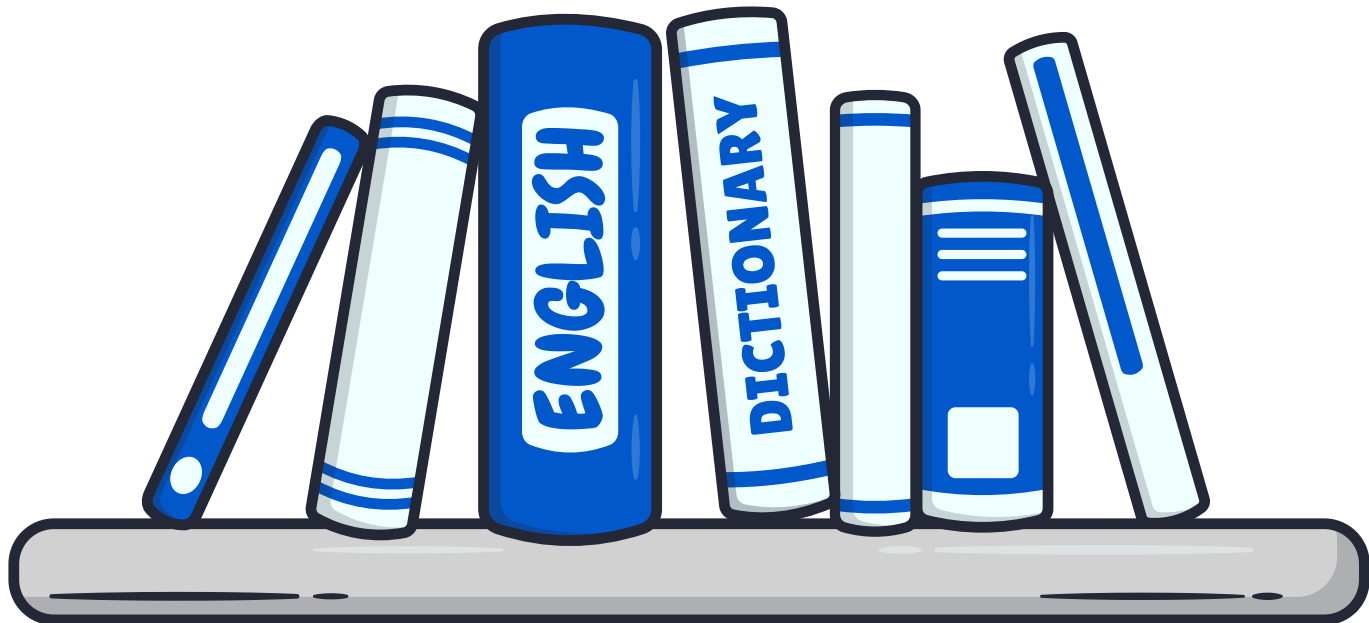
USE LOOPS TO ITERATE
THROUGH KEYS,
VALUES, OR KEY-VALUE
PAIRS.



```
for key, value in my_dict.items():  
    print(f"{key}: {value}")
```



WRAPPING UP



DICTIONARIES ARE A POWERFUL WAY TO ORGANIZE AND ACCESS DATA. PRACTICE USING THEM TO ENHANCE YOUR PYTHON SKILLS!



Since 2006

FUTURE

VISION

Computer institute

Follow Us

 +91 87991 41678

 futurevisioncomputers.com

 PAL, CITYLIGHT, VESU