6_dictionaries

Definition

Dictionary stores and connects pieces of information together.
In dictionaries, we categorize data into different groups like writing them into different dictionaries.
It is a collection of key-value pairs where you can access the value with the associated key.
For example, every individual can be a "dictionary" that contains all the important information about this person:

```python
marcos_gao={'race':'asian', 'age':'20'}
print(marcos_gao['race'])
#asian
```

A dictionary is included in "{}". Inside, keys and values are connected by ":" and different pairs are seperated by ",".
Value can be a word, a variable, a list and even a dictionary.

Dictionaries

Accessing values

```python
#name of the dictionary[key]
game={'goal_scored':'7','goal_lost':'1'}
new_goal=game['goal_scored']
print(f'you just scored your {new_goal}th goal in the game')
#you just scored your 7th goal in the game
```

Adding new pairs

```python
#to add new pairs in a dictionary, simply write down the new keys and connect them
with new values
game={'goal_scored':'7','goal_lost':'1'}
game['best player']='Marcos'
print(game)
#{'goal_scored': '7', 'goal_lost': '1', 'best player': 'Marcos'}
```

Modify values

```python
# to change the value, just repair the correct value with the key
game={'goal_scored':'7','goal_lost':'1','bset_player':'Marcos'}
game['best_player']='James'
print(game)
#{'goal_scored': '7', 'goal_lost': '1', 'bset_player': 'Marcos', 'best_player':
'James'}
```

Removing pairs

```language-python
marcos={'age':'20','wife':'sklfd'}
del marcos['wife']
print(marcos)
#{'age': '20'}
```

In many lines

```language-python
# add indentation lines to start another line
favorite_footballer={
         'marcos':'messi',
         'james':'gerrad',
         'bogart':'ronaldo'
}
```

Trouble shooting

When we use a key that haven't been assigned to a value, errors will be reported. However, we don't know where the problem occurs exactly.
By using "get()", we can assign a particular message that will be returned if the requested key doesn't exist.

```language-python
marcos={'race':'asian','weight':'66kg'}
height_marcos=marocs.get('height','Data undefined')
#if the key-value pair is in the dictionary, it will return the value, if not, it
will return the message assigned 'data undefined'
print(height_marcos)
#Data undefined
```

Looping through a dictionary

Through all pairs

```language-python
favorite_footballer={
        'marcos':'messi',
        'james':'gerrad',
        'bogart':'ronaldo'
}
# the first and second spot after "for" represents key and value respectively.
"items()" returns a list of key-value pairs.
for name,footballer in favorite_footballer.items():
        print(f"{name.title()}'s favorite footaller is {footballer.title()}'")

# Marcos's favorite footaller is Messi'
# James's favorite footaller is Gerrad'
# Bogart's favorite footaller is Ronaldo'
```

Through all keys/all values

```language-python
#we could access all keys or all values with keys.() or values.()
favorite_footballer={
         'marcos':'messi',
         'james':'gerrad',
         'bogart':'ronaldo'
}
for squad in favorite_footballer.keys():
        print(squad.title())
#Marcos
#James
#Bogart
```

Through all keys in order

```language-python
favorite_languages = {
 'jen': 'python',
 'sarah': 'c',
 'edward': 'ruby',
 'phil': 'python',
```

```
 }
 # sorted()
for student in sorted(favorite_languages.keys()):
        print(f'{student}, your registration is complete')
#edward, your registration is complete
#jen, your registration is complete
#phil, your registration is complete
#sarah, your registration is complete
```

Remove repetitions

```language-python
# there could be many value associated the same key, vice versa. we can remove the repetition with set()
favorite_languages = {
 'jen': 'python',
 'sarah': 'c',
 'edward': 'ruby',
 'phil': 'python',
 }
 for languages_mentioned in set(favorite_languages.values()):
         print(languages_mentioned)
#python
#ruby
#c
# we can see "python" only occurs once in the output
```

Sets

sets are dictionaries where there is no value associated with keys

```language-python
languages={'python','ruby','c'}
languages
#{'ruby', 'c', 'python'}
```

Nesting

Nesting helps us storing multiple dictionaries in a list, list of items as values, or dictionaries in dictionaries.

A list of dictionaries

```language-python
marcos={'nationality':'China','mbti':'entp','gender':'male'}
james={'nationality':'Hongkong(China)','mbti':'isfp','gender':'male'}
maggie={'nationality':'China','mbti':'infj','gender':'female'}
human_resources=[marcos,james,maggie]
print(human_resources)
#[{'nationality': 'China', 'mbti': 'entp', 'gender': 'male'}, {'nationality':
'Hongkong(China)', 'mbti': 'isfp', 'gender': 'male'}, {'nationality': 'China',
'mbti': 'infj', 'gender': 'female'}]
```

we could start an empty list

```language-python
aliens=[]
for alien_number in range(30):
        new_alien = {'color': 'green', 'points': 5, 'speed': 'slow'}
        aliens.append(new_alien)

# Show the first 3 aliens.
for alien in aliens[:3]:
        print(alien)
#{'color': 'green', 'points': 5, 'speed': 'slow'}
#{'color': 'green', 'points': 5, 'speed': 'slow'}
#{'color': 'green', 'points': 5, 'speed': 'slow'}
```

A dictionary of lists

```language-python
pizza_ordered={
        'size':'large',
        'toppings':['ham','pineapple']
}
#print the summary of the order

print(f"the pizza you ordered is {pizza_ordered['size']} with")
for topping in pizza_ordered['toppings']:
        print(topping)
#the pizza you ordered is large with
```

```
#ham
#pineapple
```

Dictionaries of dictionaries

```language-python
#having a dictionary of dictionaries can makes the code complicated. However, one can distinguish the key-value pair by seperating commas after'}''.
pets_collections={'cat':{'loyalty':'75','cuteness':'95','price':'50'},
'dog':{'loyalty':'95','cuteness':'50','price':'50'},'bird':
{'loyalty':'20','cuteness':'60','price':'50'}}
for pets, traits in pets_collections.items():
        print(pets)
        trait1=traits['loyalty']
        trait2=traits['cuteness']
        trait3=traits['price']
        print(f'loyalty:{trait1}\n cuteness:{trait2}\n price:{trait3}')

#cat
#loyalty:75
 #cuteness:95
 #price:50
#dog
#loyalty:95
 #cuteness:50
 #price:50
#bird
#loyalty:20
 #cuteness:60
 #price:50
```