Process dict values

Let us understand how we can process the values in dicts.

We have already seen a function called as values which return all the values as a list like object.

order_item_subtotals = {

   1: 299.98,

   2: 199.99,

   3: 250.0,

   4: 129.99

}

order_item_subtotals

{1: 299.98, 2: 199.99, 3: 250.0, 4: 129.99}

type(order_item_subtotals)

dict

order_item_subtotals.values()

dict_values([299.98, 199.99, 250.0, 129.99])

If we would like to get total revenue by adding all the values, we can apply sum.

sum(order_item_subtotals.values())

min

min(order_item_subtotals.values())

max

max(order_item_subtotals.values())

sort

sorted(order_item_subtotals.values())

We can convert the values to list and perform all available list operations.

list(order_item_subtotals.values())

[299.98, 199.99, 250.0, 129.99]

The values in the dict need not be unique. Sometimes, we might want to get all the unique values.

For example, get all the unique sports that is part of this dict. In this dict, the key is name of the sports person and the value is the sport the person plays.

You can always use `set` to get unique elements from any list type object.

famous_players = {

   'Pete Sampras': 'Tennis',

   'Sachin Tendulkar': 'Cricket',

   'Brian Lara': 'Cricket',

   'Diego Maradona': 'Soccer',

   'Roger Federer': 'Tennis',

   'Ian Thorpe': 'Swimming',

   'Ronaldo': 'Soccer',

   'Usain Bolt': 'Running',

   'P. V. Sindhu': 'Badminton',

   'Shane Warne': 'Cricket',

   'David Beckham': 'Cricket',

   'Michael Phelps': 'Swimming'

}

famous_players.values()

dict_values(['Tennis', 'Cricket', 'Cricket', 'Soccer', 'Tennis', 'Swimming', 'Soccer', 'Running', 'Badminton', 'Cricket', 'Cricket', 'Swimming'])

```
len(famous_players.values())
```

12

```
set(famous_players.values())
```

{'Badminton', 'Cricket', 'Running', 'Soccer', 'Swimming', 'Tennis'}

```
len(set(famous_players.values()))
```

6