Important Object Methods

Consider we have an object like this:

```
let product = {
        name: "Iphone 14 Pro",
        company: "Apple",
        price: 125000,
        warranty: "1 Year",
        color: "Black"
}
```

Considering the fact it is an object, it will be having a lot of key value pairs. All the keys inside an object are always unique, we cannot have duplicate keys in an object.

Let's start talking about important object functions.

Object.keys()

If we want that, we need fetch all the unique keys of our object say product, then we can use the inbuilt `Object.keys` method.
This method takes one argument which is the object whose keys are expected to be fetched.

```
Object.keys(product);
```

It returns an array of strings, where each string represents a unique key.

```
['name', 'company', 'price', 'warranty', 'color']
```

Object.values()

If we want that, we need to fetch all the available values (values can be non unique) of our object, then we can use the inbuilt method `Object.values`.

```
Object.values(product);
```

It returns an array of strings, where each string represents a value corresponding to some key in the object.

```
['Iphone 14 Pro', 'Apple', 125000, '1 Year', 'Black']
```

Object.entries

If instead of just fetching the keys or values separately, we want to fetch the complete pair of (key, value), then we can use the `Object.entries` method. This method will take one argument which is the object , and returns an array where each index contains a 2 length array whose 0th index is the key and 1st index is the value.

```
Object.entries(product);
```

```
[
        ['name', 'Iphone 14 Pro'],
        ['company', 'Apple'],
        ['price', 125000],
        ['warranty', '1 Year'],
        ['color', 'Black']
]
```

How can we check the number of key value pairs in the object ?

We can use any of the above method. Let's say we use `Object.keys` , it will return us an array with all the unique keys, and the length of that array can be our answer.
Because all the keys are unique, and we have entry of each unique key in the response array, so length of the array will be denoting the number of key value pairs.

```
Object.keys(product).length; // No of key value pairs
```