

# OBJECTS AND IT'S INTERNAL REPRESENTATION IN JAVASCRIPT

---

## Objects:

In any language, Objects are considered as a fundamental data type that can represent complex data structures. We can use objects to create custom/user-defined data types that contain one or more pre-defined data types.

## Objects in JavaScript:

- Objects in JavaScript are a collection of key-value pairs
- A key is always a string (also known as property)
- Whereas, a value can be of any,
  - Data type
  - Array
  - Function or,
  - An object itself.

The representation of an object in js is shown below:

```
() object_rep.js X
D: > DEV_projects > guvi codekata > () object_rep.js > ...
1  const info = {
2      name : "Mukeshkhanna",
3      age : 21,
4      education : {
5          class12 : "RASI INTERNATIONAL SCHOOL",
6          class11 : "RASI INTERNATIONAL SCHOOL",
7          B_tech : "SASTRA DEEMED UNIVERSITY"
8      },
9      marks : ["85%", "92.6%", "73.5%"],
10     greet : function(name) {
11         console.log('hello Im ' + name);
12     }
13 };
14
```

As we can see, the object is named as info inside the object we have:

- Name – String
- Age – Number
- Education – Object
- Marks – Array
- Greet – function

Hence, we can store different types of data types in a single object.

## The internal representation of objects in JS:

When key-value pairs are mentioned the first data structure that comes to mind is HashMap/HashTable. Just like any other language js follows this too.

The key-value pairs are stored in a hash map where a key is hashed to point to a value in the table/map. When we try to access a property in the object the engine performs a hash lookup in the table and fetches the required key-value pair.

Objects in JS are dynamic in nature, when we add a property the engine adds a key-value pair in the table. In the same way when we remove or modify a property the engine does the change accordingly to the table.

NOTE:

An important point to note is that there cannot be more than one property with the same name, when we try to use the same key more than one time the second key will be overwritten. So, we have to be careful when we name properties.

For the above snippet the key-value pair is shown below:

key	value
"name"	"Mukeshkhanna"
"age"	21
"education"	Another object representing education
"marks"	["85%","92.6%","73.5%"]
"greet"	greet(name)