

# Objects and its internal representations in Javascript

In JavaScript, objects are a fundamental data type and a versatile construct used to represent and manipulate data. Here are some points about objects and their internal representation in JavaScript:

## Key-Value Pairs:

- Objects in JavaScript are collections of key-value pairs, where each key is a string (or symbol) and each value can be any data type, including other objects.

## Dynamic Properties:

- Objects in JavaScript are dynamic, meaning you can add or remove properties at runtime. This flexibility allows for the creation of complex and adaptable data structures.

## Property Access:

- You can access object properties using either dot notation (`object.property`) or bracket notation (`object['property']`). Bracket notation is useful when the property name is dynamic or includes special characters.

## Object Literal Notation:

- The most common way to create objects is using literal notation, which involves defining the object and its properties in a concise and readable syntax.

```
let user = {  
  name: "suresh",  
  age: 23,  
  email: "suresh@gmail.com"  
}
```

## Object Methods:

- Objects can contain functions as properties, known as methods. These methods can be called to perform specific actions related to the object.

```
const calculator = {  
  add: function(a, b) {  
    return a + b;  
  },  
}
```

```
subtract: function(a, b) {  
    return a - b;  
}  
};
```