

# Objects and its internal representation in Javascript:

JavaScript, the language of the web, is renowned for its flexibility and object-oriented nature. At the heart of this flexibility the concept of objects.

An object in javascript is a collection of key - value pairs.Each key serves as an identifier ,its associated value can be of any data types,and including other objects.

This key -value structure allows developers to model real-world entities,making javascript an expressive and dynamic language.

Object Syntax:

```
Data_type Object_name=new Object();
```

## Ways of creating Object:

### 1.Literal Notation:

The most straightforward way to create an object is through literal notation, encapsulating properties within curly braces.

```
Data_type object_name ={ key1:value1,  
                        key2:value2,  
                        Key3:value3,.....}
```

Ex:

```
let person={Name:Prakash,  
            Age:22,  
            City:Tamil nadu  
            }
```

## 2.Constructor Function:

Constructor functions are useful when creating multiple instances of similar objects.

Example:-

```
function person(name,age,city){  
    this.name=name;  
    this.age=age;  
    this.city=city;  
  
}  
  
let Prakash=new Prakash("Prakash",22,"Tamil nadu")
```

## Accessing Object Properties:

Accessing properties is fundamental to working with objects. JavaScript provides two primary methods: dot notation and bracket notation.

```
console.log(person.name);  
console.log(person["city"]);
```

## Modifying and Extending Objects

JavaScript objects are mutable, allowing dynamic modifications and additions.

```
person.age=23;  
person.city="Trichy";
```

## Deleting object property syntax:

```
delete objectName.objectProperty;
```

## The Internal Representation of Objects

JavaScript manages objects by allocating memory for their properties and utilizing property descriptors to define attributes like writability and configurability.

Understanding this internal representation empowers developers to write more efficient code.

## **Conclusion:**

Objects are the backbone of JavaScript programming, providing a robust means to structure data and functionality.

Whether you're a novice using literal notation or an aficionado embracing the elegance of classes, mastering objects is pivotal for unleashing the full potential of JavaScript.

In the upcoming installments, we'll delve deeper into advanced object-oriented concepts, exploring prototypes, inheritance, and practical use cases that showcase the true power of objects in JavaScript development.

Stay tuned for more insights into the fascinating world of JavaScript programming!