

OBJECTS



OBJECTS



INTRODUCTION

JavaScript object is a non-primitive data-type that allows you to store multiple collections of data.

JAVASCRIPT OBJECT DECLARATION

The Syntax To Declare An Object Is:

```
const object_name = {  
  key1: value1,  
  key2: value2  
}
```

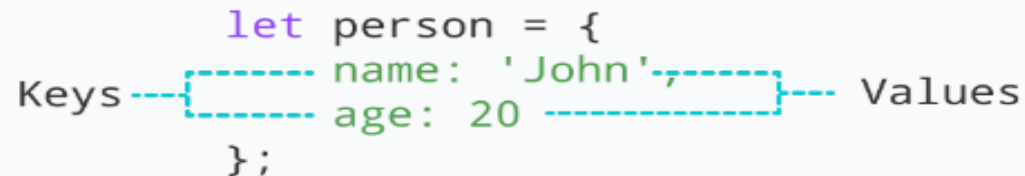
Here, an object `object_name` is defined. Each member of an object is a **key: value** pair separated by commas and enclosed in curly braces {}

JAVASCRIPT OBJECT PROPERTIES

- In JavaScript, "key: value" pairs are called **properties**. For example,

```
let person = {  
  name: 'John',  
  age: 20  
};
```

Here, `name: 'John'` and `age: 20` are properties.



The diagram shows the code `let person = { name: 'John', age: 20 };` with a dashed green box around the `name: 'John', age: 20` part. A dashed line from the word "Keys" points to the `name` and `age` keys, and another dashed line from the word "Values" points to the `'John'` and `20` values.

```
let person = {  
  name: 'John',  
  age: 20  
};
```

JavaScript object properties

ACCESSING OBJECT PROPERTIES

- We can access the **value** of a property by using its **key**.

1. Using dot Notation

Here's the syntax of the dot notation.

```
objectName.key
```

For example,

```
const person = {  
  name: 'John',  
  age: 20,  
};  
  
// accessing property  
console.log(person.name); // John
```

ACCESSING OBJECT PROPERTIES

- We can access the **value** of a property by using its **key**.

2. Using bracket Notation

Here is the syntax of the bracket notation.

```
objectName["propertyName"]
```

For example,

```
const person = {  
  name: 'John',  
  age: 20,  
};  
  
// accessing property  
console.log(person["name"]); // John
```

JavaScript Nested Objects

An object can also contain another object. For example,

```
// nested object
const student = {
  name: 'John',
  age: 20,
  marks: {
    science: 70,
    math: 75
  }
}

// accessing property of student object
console.log(student.marks); // {science: 70, math: 75}

// accessing property of marks object
console.log(student.marks.science); // 70
```

- **JavaScript Object Methods**
- In JavaScript, an object can also contain a function. For example,

JavaScript this Keyword

To access a property of an object from within a method of the same object, you need to use the `this` keyword. Let's consider an example.

```
const person = {  
  name: 'John',  
  age: 30,  
  
  // accessing name property by using this.name  
  greet: function() { console.log('The name is' + ' ' + this.name); }  
};  
  
person.greet();
```





THANK YOU



M.LATIF
KHARMANGI129@GMAIL.COM