

JAVA SCRIPT

(OBJECT AND ITS INTERNAL FUNCTION)

JAVASCRIPT OBJECTS:

A JavaScript object is a collection of named values having state and behavior (properties and method)". For example: Person, car, pen, bike, Personal Computer , Washing Machine etc. Objects are important data types in javascript. Objects are different than primitive datatypes (i.e. number, string, boolean, etc.). Primitive data types contain one value but Objects can hold many values in form of Key: value pair. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

variables are containers for data values. This code assigns a simple value (bullet) to a variable named car.

```
let bike = "bullet";
```

Objects are variables too. But objects can contain many values. This code assigns many values (bullet, 350, black) to a variable named car:

```
const bike = {type:"bullet", model:"350", color:"black"};
```

The values are written as name:value pairs (name and value separated by a colon).

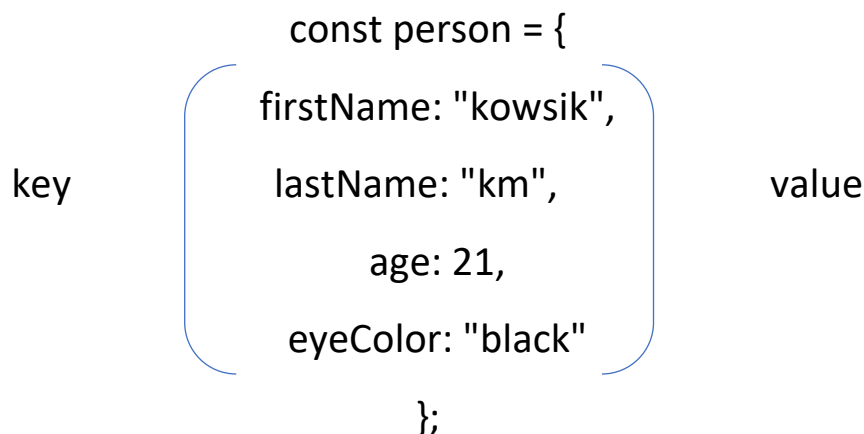
OBJECT DEFINITION

Objects, in JavaScript, is it's most important data-type and forms the building blocks for modern JavaScript. These objects are quite different from JavaScript's primitive data-types(Number, String, Boolean, null, undefined and symbol) in the sense that while these primitive data-types all store a single value each (depending on their types)

You define (and create) a JavaScript object with an object literal:

```
const person = {firstName:"kowsik", lastName:"km", age:21, eyeColor:"black"};
```

Spaces and line breaks are not important. An object definition can span multiple lines:



The diagram illustrates the structure of an object literal definition. It shows the code `const person = {` followed by a block of four key-value pairs enclosed in curly braces, and finally `};`. A blue rounded rectangle highlights the four key-value pairs. To the left of this rectangle is the label 'key' and to the right is the label 'value'. The key-value pairs are: `firstName: "kowsik",`, `lastName: "km",`, `age: 21,`, and `eyeColor: "black"`.

```
const person = {  
  firstName: "kowsik",  
  lastName: "km",  
  age: 21,  
  eyeColor: "black"  
};
```

ACCESSING OBJECTS PROPERTIES

DOT NOTATION:

`objectName.key`

EXAMPLE:

```
const person = {  
  name: 'kowsik',  
  age: 20,  
};  
console.log(person.name);
```

OUTPUT: kowsik

BRACKET NOTATON:

`objectName["propertyName"]`

EXAMPLE:

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

OUTPUT: kowsik

JAVASCRIPT NESTED OBJECTS

(An object can also contain another object.)

EXAMPLE:

```
const student = {  
  name: 'kowsik',  
  age: 20,  
  marks: {  
    science: 70,  
    math: 75  
  }  
}
```

```
console.log(student.marks);
```

OUTPUT: {science: 70, math: 75}

```
console.log(student.marks.science);
```

OUTPUT: 70

JAVASCRIPT OBJECT METHODS

(In JavaScript, an object can also contain a function.)

EXAMPLE:

```
const person = {  
  name: 'Sam',  
  age: 30,  
  greet: function() { console.log('hello') }  
}  
person.greet();
```

OUTPUT: hello