# Objects

## Introduction

Like arrays, objects can have one or more elements.

But unlike arrays, indices in objects don’t change – each value has their own reference key.

A reference key is a unique identifier that corresponds with the value.

The relationship between the items on the list is different in an object than they would be in an array.

This is because in an object each value is independent and there isn’t the need for a fixed order.

Objects are associative arrays.

Below shows the basic syntax of an object:

const artist = {

firstName: "Michael",

lastName: "Jackson",

birthYear: 1958,

numberOneHits: 13

};

## Properties and Values

Objects contain properties, which shows the characteristics of an object. Below is an example of a bicycle expressed as an object:

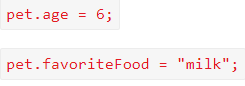
Note the commas after each property is listed.

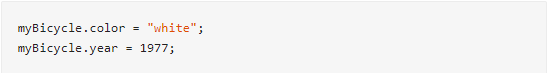
They are listed like elements of an array.

These are unlike semicolons which are used to separate statements to be runned one after another.

****

Properties can be reassigned by typing the objectname, immediately followed by “.<propertyname>”, as seen below:



Updating properties

Syntax for an **Array of Objects**:

const garden = [

{

color: 'red',

flower: 'rose',

},

{

color: 'blue',

flower: 'violet',

petals: 6

}

];

const contacts = [

{

firstName: "Sanju",

lastName: "Asho",

phone: "(512) 355-0453",

email: "sanjuasho@email.com",

},

{

firstName: "Jane",

lastName: "Doe",

phone: "(312) 641-2203",

email: "janedoe@email.com",

},

{

firstName: "Haru",

lastName: "Ito",

phone: "(415) 604-4219",

email: "haruito@email.com",

}

];

function listContacts(contactArray) {

for (let i = 0; i < contactArray.length; i++) {

console.log(contactArray[i].firstName + " " + contactArray[i].lastName)

}

}

### Methods

To write a method, we give our project a property that has a **function** as its value.

To call a method, type:

<objectName>.<methodName>()

Within the methods, we use **this** to refer to the object. This makes it easy to refer to the ***properties*** involved in the method.

