

# Objetos

## Teoría

### Cómo crear un objeto en javascript?

Simplemente escribiendo dos llaves `{ }` ya tenemos un objeto en javascript. Es usual (pero no obligatorio) guardar los objetos dentro de alguna variable:

#### Ejemplo

```
const persona = {}
```

### Cómo agregarle propiedades a un objeto?

Simplemente haciendo `nombreDelObjeto.nombreDeLaPropiedad = algunValor ...`:

- Si la propiedad no existía, creamos la propiedad con el valor dado.
- Si la propiedad ya existía, actualizamos su valor.

#### Ejemplo

```
const persona = {}  
persona.nombre = 'mariano'  
persona.edad = 32
```

También podemos utilizar la siguiente sintaxis:

```
const persona = {}  
persona['nombre'] = 'mariano'  
persona['edad'] = 32
```

Esta sintaxis nos da la ventaja de que como el argumento que pasamos entre corchetes es un string, este puede ser obtenido de diversas maneras, o incluso recibido por parámetro.

### Cómo acceder a las propiedades a un objeto?

Al igual que para agregar, existen dos maneras de acceder a una propiedad de un objeto:

- Utilizando la notación: `objeto.propiedad`
- Utilizando la propiedad: `objeto[propiedad]`

Cualquiera de las dos variantes me devuelve el mismo resultado, que es el valor de la propiedad en cuestión.

#### Ejemplo

```
console.log(persona.nombre)
console.log(persona['edad'])
```

## Cómo quitarle propiedades a un objeto?

Para quitarle una propiedad a un objeto usaremos la palabra reservada **delete** y a continuación la propiedad que queremos borrar, accediéndola desde su objeto contenedor:

#### Ejemplo

```
delete persona.edad
```

## Propiedades anidadas

Es posible declarar propiedades que sean a su vez también objetos.

#### Ejemplo

```
const persona = {}
persona.nombre = 'mariano'
persona.edad = 32
persona.dirección = {}
persona.direccion.calle = 'rivadavia'
persona.direccion.numero = 1234
persona.telefonos = []
persona.telefonos.push('15-1234-5678')
persona.telefonos.push('15-1234-5000')
```

## Declaración e inicialización en un solo paso

Nótese que al representar al objeto (mostrarlo) éste aparece como una serie de pares de datos, separados por comas, todo dentro de las llaves que delimitan al objeto. Estos pares de datos representan siempre una clave y un valor, es decir, el nombre del par, y el valor asociado a ese nombre. El valor asociado puede ser cualquier cosa, incluyendo (no exclusivamente) números, strings, objetos, arrays, etc.

Es posible declarar un objeto en js ya inicializándolo con sus valores, todo en una sola operación. En este caso, se debe separar cada clave de cada valor usando dos puntos ( `:` ), y cada par clave/valor del siguiente usando comas ( `,` ).

### Ejemplo

```
const persona = {  
  nombre: 'mariano',  
  edad: 32,  
  direccion: {  
    calle: 'rivadavia',  
    numero: 1234  
  },  
  telefonos: ['15-1234-5678', '15-1234-5000']  
}
```