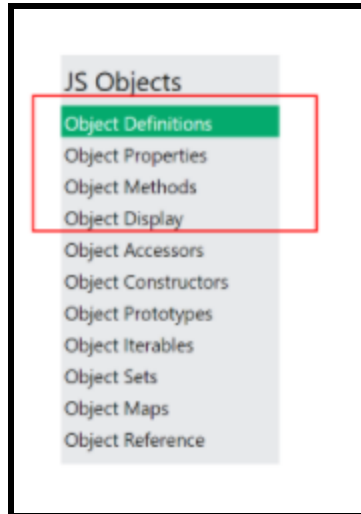


Realiza una **lectura comprensiva** de los apartados: “definición”, “propiedades”, “métodos” y “visualización de objetos” en el enlace [JavaScript Objects](#)



### Definición de objetos y propiedades

Uno de los aspectos más importantes del lenguaje Javascript es el concepto de objeto, puesto que prácticamente todo lo que utilizamos en Javascript, son objetos. Sin embargo, tiene ligeras diferencias con los objetos de otros lenguajes de programación.

En Javascript, existe un **tipo de dato** llamado objeto (*object*). Una primera forma de verlo, es como una variable especial que puede contener más variables en su interior. De esta forma, tenemos la posibilidad de organizar múltiples variables de la misma temática en el interior de un objeto.

En muchos lenguajes de programación, para crear un objeto se utiliza la palabra clave *new*. En Javascript también se puede hacer, pero por el momento evitaremos su uso. En esta parte del curso, por simplicidad, legibilidad y velocidad de ejecución, utilizaremos para definir objetos la **sintaxis “literal de objeto” (sintaxis JSON)**, es decir, crearemos objetos sin necesidad de un constructor o instanciar una clase.

A continuación se muestran varias técnicas para la [visualización de objetos](#):

- a través del **nombre de la propiedad** (puedes usar notación con puntos o con corchetes)

```
const person = {
  firstname: "John",
  lastname: "Doe",
  age: 50,
  eyecolor: "blue"
};
document.getElementById("demo").innerHTML = person.firstname + " is " + person["age"] + " years old.";
```

*ojo! porque hay algunos casos en los que sólo se puede utilizar la notación con corchetes, como por ejemplo cuando se utilizan espacios en el nombre de la propiedad. Es imposible hacerlo con la notación con puntos.*

- iterando por las propiedades del objeto mediante un bucle **for in**

```
const person = {
  fname:"John",
  lname:"Doe",
  age:25
};

let txt = "";
for (let x in person) {
  txt += person[x] + " ";
}

document.getElementById("demo").innerHTML = txt;
```

- haciendo uso de **Object.values ()** -- devuelve un array con los valores del objeto
- usando **JSON.stringify ()** -- transforma un objeto de Javascript en una cadena con formato JSON

## Puesta en práctica:

**Ejer1.** Define el objeto coche con las propiedades puertas(2), ruedas(4) y color("azul"). Muestra en pantalla todos los valores de coche, haciendo referencia a las propiedades del objeto por su **nombre**

**Ejer2.** Define el objeto persona = {nombre:"Juana", apellido:"CaLamidad", edad:34}  
Visualiza las propiedades del objeto iterando por las propiedades del mismo con un bucle **for in**

**Ejer3.** Partiendo de una definición de objeto persona similar a la del Ejer2, **añade** al objeto la propiedad nacionalidad con el valor "Americana" y muestra en pantalla mediante **Object.values ()** todos los valores de persona para comprobar que la propiedad ha sido añadida. Para aprender más: prueba el uso de **Object.keys ()** y de **Object.entries()** y extrae conclusiones.

Ojo!, no se pide definir un objeto persona con 4 propiedades, sino un objeto persona con 3 propiedades, tal y como lo has hecho en el Ejer2, al que posteriormente has de añadir una nueva propiedad.

**Ejer4.** Dada la definición del objeto persona de Ejer2, elimina la propiedad edad y muestra en pantalla el objeto como una **cadena JSON**. Al igual que en el ejercicio anterior la operación de eliminación es a posteriori; comprueba el resultado mostrando en pantalla el objeto como una **cadena JSON**

## Métodos de objeto

Hasta ahora, solo hemos visto cómo crear en Javascript objetos «genéricos» (un tipo de datos llamado object). También hemos visto que es posible añadir y eliminar propiedades a un objeto, que no son más que variables dentro del objeto en cuestión.

Si dentro de una variable del objeto metemos una función (o una variable que contiene una función), tendríamos lo que se denomina un **método de un objeto**. Recuerda que en la tarea10 estudiando funciones, ya adelantamos que "una función definida como la propiedad de un objeto, se llama **método del objeto**". Un método de objeto es una propiedad de objeto que contiene una **definición de función**.

En su interior, la palabra reservada **this** hará referencia al objeto propietario de la función. Esto es, cuando en un método del objeto *persona* escribimos **this.apellido** nos estaremos refiriendo a “la propiedad apellido de éste objeto”.

**Ejer5.** Crea el objeto “persona” con las propiedades nombre(“Camilo”), edad(22) y el método “hablar” mediante una función que retorne “Hola soy” + **Camilo** + ” y tengo “ + **22** + “años”. A continuación accede al método con la sintaxis adecuada para mostrar en pantalla el mensaje generado.

## Objetos anidados - objetos y matrices

**Ejer6.** Los valores o propiedades de un objeto pueden ser otro objeto. Escribe un ejemplo de objetos anidados y accede y muestra los valores del mismo. El ejemplo ha de ser diferente a los propuestos de ejemplo en el tutorial.

**Ejer7.** Las propiedades de los objetos pueden ser matrices y los valores de las matrices pueden ser objetos. Define mediante sintaxis JSON el **objeto alumno** que recoja la información que a continuación se detalla y la muestre en pantalla:

el alumno de nombre Juan Pérez Martínez matriculado en el curso DAM2 ha obtenido las siguientes calificaciones

- ☐ Acceso a datos , RAs: [7,4,7,8,10,8]
- ☐ Desarrollo web , RAs: [6,7,6,8,5,9,7,5]
- ☐ Sistemas de gestión empresarial, RAs: [5, 5, 6, 6, 8]
- ☐ Programación de servicios y procesos, RAs: [5, 5, 4, 6, 8]

Teniendo en cuenta que:

- ☐ La propiedad **calificaciones** del **objeto alumno** es una matriz de objetos con **propiedades** módulo y RAs.
- ☐ RAs es una matriz de enteros que contiene las notas de los resultados de aprendizaje de cada módulo.

Ejemplo:





## JS-Tarea11-Objetos- definición, propiedades, métodos y visualización

*DAM2 - DWEB*

**Ejer8.** Desarrolla la página que permita al usuario añadir las calificaciones de los módulos “Desarrollo de Interfaces” y “Programación multimedia y dispositivos móviles” del alumno Juan Pérez. La página debe también permitir visualizar todas las notas.