

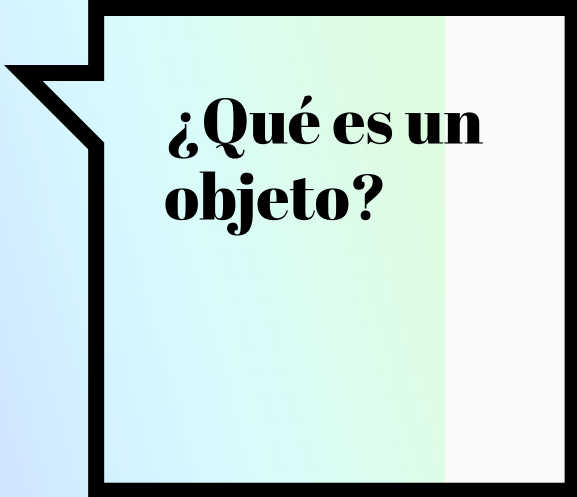
# Programación Web

Clase 4 - Objetos



**1.**

**Objetos**



## **¿Qué es un objeto?**

Es un tipo de dato en el que podemos guardar gran cantidad de información relacionada.

Básicamente, es cualquier cosa que podamos pensar como un elemento, con sus características propias.

Por ejemplo, un auto podría ser un objeto, y sus propiedades podrían ser el fabricante, el modelo, el año de fabricación, etc.



## **Propiedades y acciones**

Además de tener propiedades, los objetos tienen acciones relacionadas. A estas acciones las llamamos métodos.

Por ejemplo, el auto podría tener métodos como acelerar, frenar, girar hacia derecha o izquierda.

## Object literal

Tenemos varias formas de definir un objeto. Una de ellas es asignandolo a una variable. Lo hacemos escribiendo llaves.

Dentro de las llaves, podemos crear propiedades (como variables pero no hace falta declararlas) y asignarles valores. Estas propiedades se separan con coma.

```
var perro = {  
  nombre = "Doug",  
  raza = "Golden",  
  edad = 2  
};
```

## Propiedades de objetos

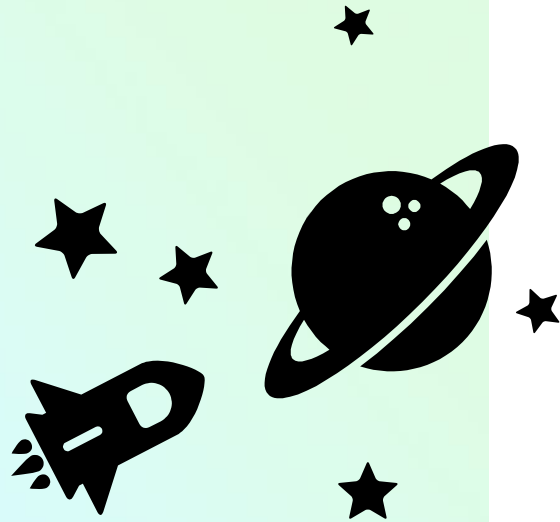
Para acceder a las propiedades de un objeto, podemos llamarlo usando el nombre del objeto + un punto (.) + el nombre de la propiedad.

De esta forma, podemos incluso crear nuevas propiedades para el objeto, y asignarles valores.

```
console.log(perro.nombre); // imprime "Doug"  
perro.genero = "macho"; // nueva propiedad  
perro.edad = 3; // nuevo valor
```

# Actividad

Creen un objeto "persona" y agreguenle propiedades con sus datos. Nombre, apellido, edad, si estudian o no (booleano), peliculas favoritas (3 en un array). Impriman la info en la consola.



## Propiedades con corchetes

Tenemos otra forma de acceder a las propiedades de un objeto, utilizando corchetes.

```
var perro = {  
  nombre = "Doug",  
  edad = 2  
};
```

```
console.log(perro["edad"]); // 2  
perro["edad"] = 3;
```



## Recorriendo propiedades

Utilizando esta sintaxis con corchetes, podemos recorrer nuestro objeto utilizando un **for in**.

```
var perro = {  
  nombre = "Doug",  
  edad = 2  
};  
  
for (propiedad in perro) {  
  alert(propiedad); // imprime "nombre" y "edad"  
  alert(perro[propiedad]); // imprime "Doug" y 2  
}
```

## Tipos de objetos

**Nativos:** Objetos que vienen pre-definidos por Javascript.

**Objetos del Host:** los provee el ambiente en el que estemos desarrollando. El ambiente que vamos a manejar nosotros, por ejemplo, es el browser.

**Propios:** Los que definamos.

*Nativos:*

*Number, String, Array, Boolean, etc*

*Host:*

*document, console, Element, etc*

*Propios:*

*perro, persona, los que definamos*



**2.**

**Métodos**



## **¿Qué son los métodos?**

Los métodos son acciones (funciones) que asignamos a los objetos.

Volviendo a nuestro ejemplo de un perro, el perro hace ciertas acciones, como ladrar.

Podríamos entonces querer linkear esa acción a nuestro objeto, en lugar de tenerla como una función independiente, dado que los objetos sirven justamente para almacenar y agrupar propiedades y acciones relacionadas.

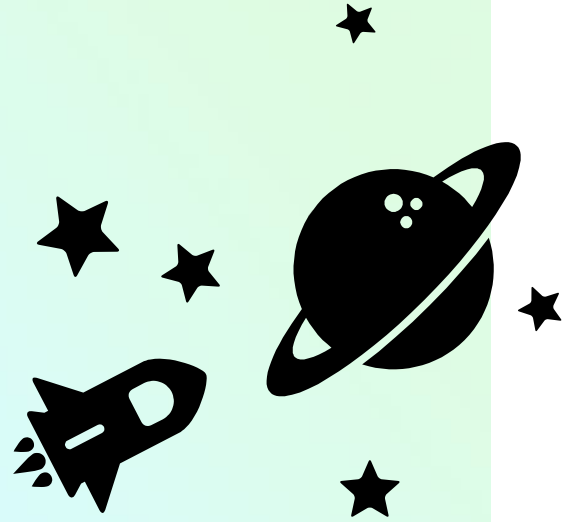
## Pasar de función a método

```
// Como función:  
function ladrar() {  
    console.log("guau");  
}  
ladrar();  
  
// Como método:  
var perro = {  
    ladrar: function() {  
        console.log("guau");  
    }  
};  
perro.ladrar();
```

# Actividad

Crear una calculadora como un objeto. Debería tener los métodos sumar y restar, y deberíamos poder accederlos como `calculadora.sumar()` y `calculadora.restar()`

Puede tomar los parámetros que quieran.





**3.**  
**this**

## Entendiendo this

This es una variable que cambia de valor con respecto al scope.

Se refiere siempre al objeto dueño del método en el que la usamos.

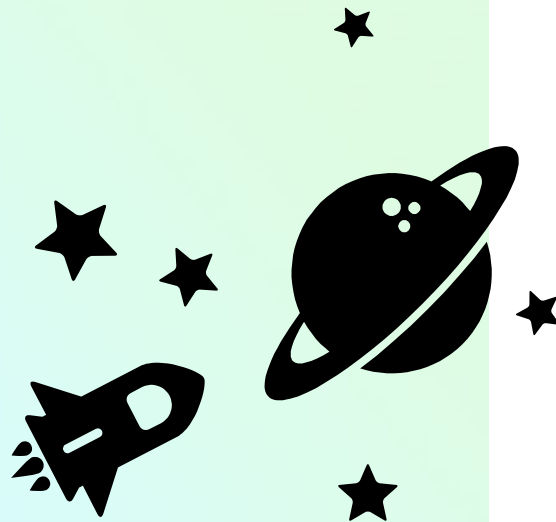
```
var producto = {  
  precio: 20,  
  iva: function() {  
    return this.precio * 1.21;  
  }  
};
```

En nuestro ejemplo, toma producto.precio



# T.P. Nº 4

Crear una función que agregue películas. El objeto película. Deberá tener ID, título, descripción, año, duración, actores, director. Deberá tener métodos para editar todas sus propiedades, menos su ID, y para los actores tendrá que ir agregando uno a uno.



# Hasta la próxima clase!

¿Preguntas?

Enviar a

[namorcamila@gmail.com](mailto:namorcamila@gmail.com)

