Todos usamos JavaScript Console Log  para imprimir los típicos mensajes de consola  en JavaScript. Sin embargo muchas veces nos olvidamos de las posibilidades que la consola de JavaScript nos aporta. Vamos a ver algunos ejemplos que nos ayudarán a clarificar. Para ello empezaremos con la  operación más sencilla a realizar es imprimir un objeto por la consola.

```
var objeto= {nombre:"pedro",edad:20};
console.log(objeto);
```

El resultado se muestra en la consola:




Aunque nos imprimir con claridad los datos del objeto , no es la mejor forma de imprimir los datos de un objeto en pantalla .¿Por qué? porque únicamente imprime la información del objeto y no nos aporta información sobre el contexto. Vamos a ver como mejorarlo.

# JavaScript Console Log

Podemos usar console.log y utilizar el comodín de substitución de objeto , vamos con ello:

```
var objeto= {nombre:"pedro",edad:20};
console.log("imprimiendo desde el metodo 1 %o",objeto);
```

El resultado es mucho más claro:




# JavaScript Console Log Arrays

De igual forma podemos trabajar con arrays de objetos , la forma que tiene de imprimir la información la consola es bastante clara:

```
var array= [{nombre:"pedro",edad:20},{nombre:"ana",edad:30}];
console.log(array);
```

El problema es que a veces los datos que el objeto tiene son muchos y no queda tan claro.




Podemos clarificarlos utilizando console.table():

```
var array= [{nombre:"pedro",edad:20},{nombre:"ana",edad:30}];
console.table(array);
```



| (index) | nombre | edad |
|---|---|---|
| 0 | "pedro" | 20 |
| 1 | "ana" | 30 |

▶ Array(2)

Como podemos ver la consola es una clase muy flexible . Otra opción interesante cuando trabajamos con objetos es la posibilidad de comprimir bloques de consoles.log con console.group().

```
var array= [{nombre:"pedro",edad:20},{nombre:"ana",edad:30}];
var array2= [{nombre:"gema",edad:50},{nombre:"ana",edad:40}];
var array3= [{nombre:"marco",edad:50},{nombre:"ana",edad:40}];
console.group();
console.table(array);
console.table(array2);
console.table(array3);
console.groupEnd();

console.group();
var objeto= {nombre:"pedro",edad:20};
var objeto2= {nombre:"pedro",edad:30};
console.log(objeto);
console.log(objeto2);
console.groupEnd();
```

El resultado como podemos observar se encarga de agrupar los bloques de información.




Pudiendo comprimir el que deseemos:




Todos estos métodos nos serán muy útiles cuando trabajemos con la consola. La consola de JavaScript siempre nos sorprende con sus curiosidades.

Otros artículos relacionados:

1. JavaScript console time y rendimiento
2. JavaScript setTimeout vs setInterval
3. ¿Qué es un JavaScript Bundle?
4. La consola de Javascript