

JAVASCRIPT



JS

Contenido

★ Objeto Date()

- getDate()
- getMonth()
- getDay()
- getFullYear()

★ Métodos de strings

- length
- indexOf()
- slice()
- trim()
- split()
- toString()

Javascript al igual que
muchos lenguajes,
tiene un objeto para
generar fechas

Objeto Date()

¡Lo primero!

Para poder comenzar a trabajar con el objeto **Date()**, lo primero que tenemos que hacer es crear una **instancia** del mismo.

```
var myOwnDate = new Date();
```

```
// myOwnDate ahora es un objeto con varias  
propiedades para acceder
```

Objeto Date()

getDate()

Método que retorna el número del **día del mes** actual.

```
var actualDayNumber = myOwnDate.getDate();
```

```
// actualDayNumber - Number - del 1 al 31
```

Objeto Date()

getMonth()

Método que retorna el número del **mes actual**.

```
var actualMonth = myOwnDate.getMonth();
```

```
// actualMonth - Number - del 0 al 11
```

```
// 0 => Enero / 11 => Diciembre
```

Objeto Date()

getDay()

Método que retorna el **día de la semana** en formato *Number*.

```
var actualDayName = myOwnDate.getDay();
```

```
// actualDayName - Number - del 0 al 6
```

```
// 0 => Domingo / 6 => Sábado
```

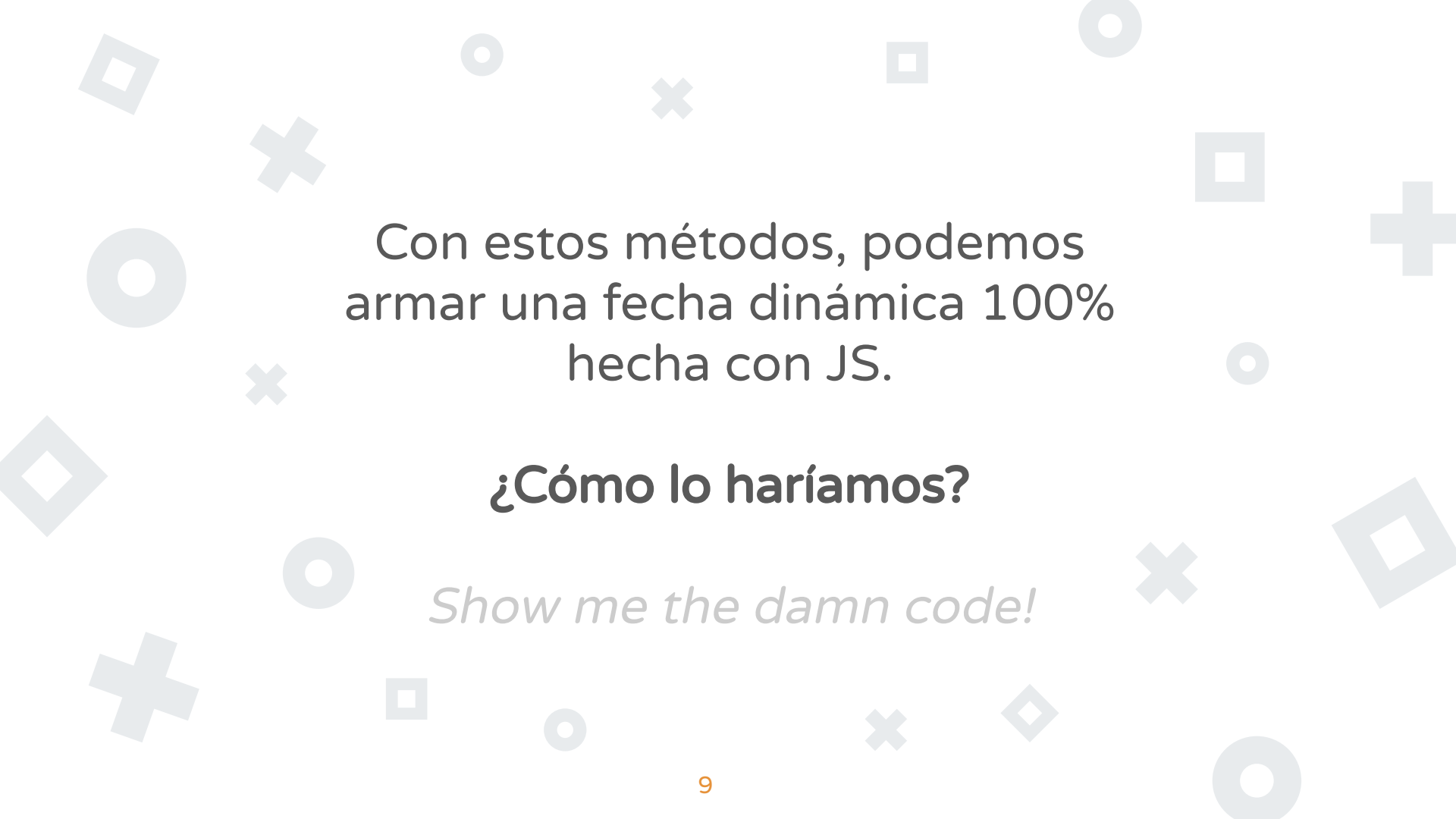
Objeto Date()

getFullYear()

Método que retorna el **año completo** en formato Number.

```
var actualFullYear = myOwnDate.getFullYear();
```

```
// actualFullYear - Number
```

Con estos métodos, podemos
armar una fecha dinámica 100%
hecha con JS.

¿Cómo lo haríamos?

Show me the damn code!

Objeto Date()

Podemos construir una fecha determinada

El objeto **Date()**, nos permite crear un objeto de tipo Date. Recibe 3 parámetros: **Año completo, Mes, Día.**

```
var myBirthDate = new Date(1998, 4, 11);
```

// myBirthDate ahora es un objeto de tipo Date. Podemos usar los mismos métodos vistos. La fecha de ésta instancia es el 11 de mayo de 1998.

¡A practicar!

Ejercicios Objeto Date



```
// To Do  
console.log("Practice Time");
```



Métodos de Strings

Tal como los Arrays, los Strings en JS son susceptibles a ejecutar sobre ellos varios métodos.

Métodos de Strings

.length

Retorna la cantidad de caracteres de un *String* (en un *Array* la cantidad de elementos). Sirve también en *funciones* ¿WTF?

```
var randomWord = "Breaking Bad Rules";  
console.log(randomWord.length)  
// 18
```

Métodos de Strings

.indexOf()

Retorna un *Number*, que será la primer posición encontrada de un *String* dado. O retorna **-1** si no se encuentra nada.

```
var randomWord = "Breaking Bad Rules";  
console.log(randomWord.indexOf("Bad")); // 9  
console.log(randomWord.indexOf("hola")); // -1
```

Métodos de Strings

.match()

Este método busca dentro de un *String* una cadena que coincida con una expresión regular suministrada, y retorna esos "matcheos" como un Array.

```
var randomWord = "Hello world, hello world, hello world, hello world";  
var matches = randomWord.match(/world/g);  
console.log(matches); // ["world", "world", "world", "world"]
```

Métodos de Strings

.slice()

Extrae una parte del *String* y retorna dicha cadena en un nuevo *String*. Recibe como parámetros 2 números, *desde* - *hasta*.

```
var randomWord = "Breaking Bad Rules";  
var extracted = randomWord.slice(8, 18);  
console.log(extracted); // "Bad Rules"  
var otherExtracted = randomWord.slice(-5, randomWord.length);  
console.log(otherExtracted); // ¿?
```


Métodos de Strings

.trim()

"**Poda**" los espacios vacíos antes y después de la ocurrencia del 1er y último caracter y retorna el String sin dichos espacios.

```
var randomWord = "    Hello world!    ";  
var withOutSpaces = randomWord.trim();  
console.log(withOutSpaces); // "Hello world!"
```

Métodos de Strings

.split()

Toma un *String* y lo divide en varios strings. Retorna un *Array*. Toma como parámetro el valor de corte. El valor de corte es muy **IMPORTANTE**.

```
var randomWord = "And Bingo was his name, oh!";  
var arrayOfString = randomWord.split(" ");  
console.log(arrayOfString);  
// ["And", "Bingo", "was", "his", "name,", "oh!"]
```

Métodos de Strings

.replace()

Este método permite reemplazar un "pedazo" de cadena de texto, dentro de un String completo. Recibe dos parámetros. String a buscar, String de reemplazo.

```
var randomWord = "We love Python!";  
var finalWord = randomWord.replace("Python", "Javascript");  
console.log(finalWord); // "We love Javascript!"
```

¿Qué pasa si queremos cambiar todas las coincidencias de texto?

Métodos de Strings

.toString()

Dado un *Number* o *Array*, retorna el valor absoluto en formato *String*.

```
var numberA = 347;
var numberB = 743;
console.log(numberA + numberB); // 1090
console.log(numberA.toString() + numberB.toString()); // "347743"
var myArray = ["Hello", "world!"];
console.log(myArray.toString()); // "Hello,world!" - Y ¿si usamos join()?
```

¡A practicar!

Ejercicios Métodos de
Strings



```
// To Do  
console.log("Practice Time");
```