



Números, fechas y temporizadores

Números - bueno saber

```
console.log(90 === 90.0); // true
console.log(Number('125')); // 125
console.log(+ '125'); // 125
console.log(Number.parseInt('50px')); // 50
console.log(Number.parseFloat('2.5')); // 2
console.log(Number.parseFloat('5.4')); // 5.4

console.log("====Number.isNaN37====")
console.log(Number.isNaN(20)); // false
console.log(Number.isNaN("23")); // false
console.log(Number.isNaN(4 / 0)); // false

console.log("====isFinite====")
console.log(Number.isFinite(20)); // true
console.log(Number.isFinite("23")); // false
console.log(Number.isFinite(4 / 0)); // false

console.log("====isInteger====")
console.log(Number.isInteger(30)); // true
console.log(Number.isInteger(30.0)); // true
console.log(Number.isInteger(30 / 0)); // false
```

Números - bueno saber

```
console.log(Math.sqrt(49)); // 7
console.log(5 * 5); // 25
console.log(5 ** 2); // 25
console.log(8 ** (1 / 3)); // 2
console.log(Math.max(2, 10)); // 10
console.log(Math.min(2, 10)); // 2
console.log(Math.PI); // 3.141592653589793
console.log(Math.random()); // 0-1
console.log(Math.trunc(Math.random() * 10 + 1)); // 1-10
const myRandom = (min, max) =>
    Math.trunc(Math.random() * (max - min) + 1) + min;
console.log(myRandom(10, 20)); // 10-20
```


Números - bueno saber

```
console.log("====rounding====")
console.log(Math.round(40.23)); // 40
console.log(Math.ceil(23.2)); // 24
console.log(Math.floor(23.7)); // 24 esto es para negativos también
// decimal redondeado
console.log((2.7).toFixed(0));
console.log((2.7).toFixed(3)); // 2.700
console.log((1.768).toFixed(2)); // 1.77
console.log((2.876).toFixed(2)); // 2.88
// Números grandes
console.log(Number.MAX_SAFE_INTEGER); // 9007199254740991
```

Fechas

El objeto JavaScript Date nos permite trabajar con fechas. Por defecto, Javascript usa la zona horaria del navegador y muestra una fecha como texto (full string).

`new Date()` crea un objeto nuevo con la fecha y hora actual. Por ejemplo:

```
const d = new Date();
```

Dechas

```
const now = new Date();
console.log(now);
console.log(new Date("November 12, 2021"));
console.log(new Date(2021, 10, 25, 10, 43, 20));
// año, mes, día, hora, minutos, segundos
// mes: 0-11
console.log(new Date(2021, 10, 25)); // año, mes, día
console.log(now.getFullYear());
console.log(now.getMonth());
console.log(now.getDate());
console.log(now.getDay());
console.log(now.getHours());
console.log(now.getMinutes());
console.log(now.getSeconds());
console.log(now.toISOString());
console.log(now.getTime()); // estampa
// puedes usar setters
now.setMonth(1);
```

```
const day1 = new Date(2021, 10, 25);  
const day2 = new Date(2024, 5, 25);  
  
const calcNumOfdays = (d1, d2) =>  
    Math.abs(d1 - d2) / (1000 * 60 * 60 * 24);  
  
const days = calcNumOfdays(day1, day2);  
  
console.log(days);
```

El método `setTimeout()` llama a una función desde de un número de milisegundos.

1 segundo = 1000 milisegundos.

Notas:

`setTimeout()` se ejecuta solo una vez.

Si necesitas varias ejecuciones, usa `setInterval()` en vez.

Usa el método `clearTimeout()` para prevenir que la función empiece.

setTimeout

```
const arr = ["red", "blue", "green", "black"];

const colors = () => {
  const random = Math.floor(Math.random() * 4); // 0-4
  document.body.style.backgroundColor = arr[random];
};

setTimeout(colors, 3000); // llama a colors después de 3 segundos
```

setTimeout

```
const arr = ["red", "blue", "green", "black"];

const colors = range => {
  const random = Math.floor(Math.random() * range); // 0-range
  document.body.style.backgroundColor = arr[random];
};

setTimeout(colors, 3000, 4); // llama a colors después de 3 segundos
```

El método `setInterval()` llama a la función en intervalos específicos (en milisegundos).

El método `setInterval()` continúa llamando la función hasta que `clearInterval()` es llamada, o que la ventana sea cerrada.

1 segundo = 1000 milisegundos.

Ejemplo:

Muestra "Hello" cada segundo (1000 milisegundos):

```
setInterval(() => element.innerHTML += "Hello", 1000);
```

setInterval

```
const timer = document.getElementById("timer");

const counting = () =>
  timer.innerText = Number(timer.innerText) + 1;
};

setInterval(counting, 1000);
```

Crea una página de html que muestre cada segundo un número entre 0 y 100


```
const range = 100;

const nums = () => {
  document.body.innerText =
    Math.floor(Math.random() * range); // 0-100
};

setInterval(nums, 1000); // llama a la función nums cada segundo
```

A nighttime photograph of a city skyline, featuring several tall skyscrapers with illuminated windows. The buildings are set against a dark blue sky. In the foreground, a red speech bubble with a white outline contains the word "wawiwa" in white, lowercase, sans-serif font. The speech bubble is positioned on the left side of the image, pointing towards the right.

wawiwa

¿Preguntas?