

## Práctica

### 1. bucles

En javascript puedes calcular una potencia  $x^n$  de la forma `x**n`, o a través del objeto Math:  
`Math.pow(x, n)`

a) Haz los siguientes ejercicios utilizando en cada uno de ellos un bucle diferente (`for`, `while` y `do-while`).

- Utiliza un bucle para calcular  $2^{53}$  (máxima precisión entera para JS).
- Muestra también  $2^{53} + 1$ .
- Con un bucle calcula  $2^{128}$  (número de IPv6 disponibles)
- Calcula el factorial de 128.

b) Haz los mismos cálculos utilizando números `BigInt` y utilizando un bucle diferente en cada caso.

### 2. objeto Math

En JavaScript, el objeto `Math` es una librería cuyos métodos son las clásicas funciones matemáticas, por ejemplo:

```
document.write("<h1>", Math.random(), "</h1>")
```

mostraría un número aleatorio comprendido entre 0 y 1 (en Java, equivaldría a la clase `java.lang.Math`).

Realmente, el número 1 no saldrá nunca, ya que los números aleatorios se generan en el intervalo [0, 1).

[Math en MDN](#)

a. Muestra en la página web un número aleatorio en el rango de 0 a 6.

b. o mismo de antes, pero eliminando la parte decimal del número aleatorio. Es decir, debes generar un número cualquiera del conjunto: { 0, 1, 2, 3, 4, 5 }

El objeto `Math` tiene multitud de funciones estáticas para realizar cálculos matemáticos. Algunos de ellos son:

<code>Math.random()</code>	// número aleatorio entre 0 y 1
<code>Math.ceil(x)</code>	// redondea al entero superior
<code>Math.floor(x)</code>	// redondeo al entero inferior
<code>Math.round(x)</code>	// redondeo al entero más próximo
<code>Math.trunc(x)</code>	// trunca (elimina la parte decimal)
<code>Math.max(x, y, z, ...)</code>	// devuelve el máximo de los argumentos
<code>Math.min(x, y, z, ...)</code>	// devuelve el mínimo de los argumentos
<code>Math.PI</code>	// 3.141592653589793

### 3. objeto Date

En JavaScript, el objeto **Date** se utiliza para instanciar objetos que almacenan una fecha y hora. por ejemplo:

```
document.write("<h1>", new Date(), "</h1>")
```

mostraría la fecha y hora actual (en Java, equivaldría a la clase **java.util.Date**).

[Date en MDN](#)

Algunas formas de utilizarlo podrían ser:

```
let d1 = new Date()
let d2 = new Date(2025, 11, 23)
let d3 = new Date("2025/12/23")
let d4 = new Date(2025, 11, 23, 23, 59, 59)
let d5 = new Date("2025/12/31 23:59:59")
```

- Visualiza las fechas de **31 de febrero**, y **32 de marzo** de este año.
- Las fechas se almacenan en la memoria como un número, el número de milisegundos transcurridos desde ... ¿qué fecha? (**Unix epoch**)
- Si restas 2 fechas, por ejemplo **d3 - d1** ¿qué obtienes?
- ¿Cuántos días quedan para que acabe el año?
- ¿Cuántos años tienes? ¿y meses? ¿y días?

Algunos métodos de los objetos **Date**:

<code>date.getDate()</code>	// día del mes (1-31)
<code>date.getDay()</code>	// día de la semana (0-6)
<code>date.getMonth()</code>	// mes (0-11)
<code>date.getFullYear()</code>	// año (yyyy)
<code>date.getHours()</code>	// hora (0-23)
<code>date.getMinutes()</code>	// minutos (0-59)
<code>date.getSeconds()</code>	// segundos (0-59)
<code>date.getMilliseconds()</code>	// milisegundos (0-999)

### 3. objeto **String**

En javascript podemos encerrar una cadena de caracteres entre comillas simples ('), comillas dobles (" ) y acentos graves (` ).

[String en MDN](#)

- Cuando un texto se encierra entre acentos graves se denomina **'template string'** (o plantilla de texto). ¿Qué ventaja tienen los **template string** respecto a las otras dos formas de crear cadenas?
- ¿Qué método utilizas para obtener el 5º carácter de una cadena? (piensa en java)  
Si decimos que un String es casi como un Array, a lo mejor se te ocurre otra forma ...
- ¿Para qué se utiliza el método **indexOf(n)** ? (piensa en java)
- ¿Qué variable miembro almacena la longitud de la cadena? (piensa en java)

e. ¿Qué operador se utiliza para encadenar varios strings? (piensa en java)

f. Podemos utilizar los operadores de comparación (`<`, `<=`, `==`, `!=`, `==`, `!==`, `>=`, `>`) para comparar strings, por ejemplo:

```
if(s1 == s2)
    document.write(`"${s1}" es igual que "${s2}"`);
else
    document.write("ambas cadenas son distintas");
```

Dados los string "gato" y "Gato" ¿cuál es mayor? ¿Sabrías explicar por qué?.

g. El método `slice([inicio [, fin]] )` (rodaja, rebanada) devuelve un substring de un string. Extrae la última palabra de una cadena, por ejemplo "Escorial" en:

```
let ciudad = "San Lorenzo de El Escorial";
```

También existe el método `substring(índiceA [, índiceB])`, pero `slice` tiene la ventaja de admitir índices negativos.

h. Échale un vistazo a los siguientes métodos de objetos `String`:

```
indexOf(searchString [, fromIndex]), lastIndexOf(searchString [, fromIndex])
startsWith(searchString [, position]), endsWith(searchString [, position])
includes(searchString [, position])
repeat(count)
padEnd(count [, string]), padStart(count [, string])
trim(), trimStart(), trimEnd()
toUpperCase(), toLowerCase()
slice([inicio [, fin]] )
replace(pattern, replacement), replaceAll(pattern, replacement)
```

i. Con la siguiente instrucción:

```
let edad = prompt("¿Cuántos años tienes?")
```

solicitamos al usuario una entrada de texto en una ventana emergente.

[prompt en MDN](#)

- Crea un script que solicite una frase al usuario y muestre las palabras primera y última de esa frase.

P.ej. si el usuario introduce: "Madrid es la capital del reino", debe mostrar 'Madrid' y 'reino'.