

Fundamentos de Programación

JavaScript
Objetos Predefinidos:
`String`, `Math`, `Date`

IES La Nía



Objeto

- Un objeto permite agrupar datos en una entidad y asociarle comportamiento
 - Similar a una entidad EER
- Puede contener atributos y operaciones
 - Atributos → propiedades → variables
 - Operaciones → métodos → funciones
- Para acceder a las propiedades y a los métodos, hay que usar el operador .
- En JavaScript todo es un objeto

String

- Las cadenas son objetos String
- Propiedad **length** → longitud de la cadena
- Métodos:
 - cadena.**toUpperCase()** / **toLowerCase()** → Transforma la cadena a minúsculas/MAYÚSCULAS
 - cadena.**charAt**(índice) → Devuelve el carácter que ocupa la posición índice (0-index)

Ejemplo String

```
var nombre = "Bruce Wayne"; // string

console.log(nombre);
console.log(typeof(nombre));
console.log(nombre.toUpperCase());
console.log(nombre.toLowerCase());

console.log(nombre.length); // es una propiedad

console.log(nombre.charAt(0));

console.log(nombre.charAt(nombre.length));
console.log(nombre.charAt(nombre.length-1));
```

Ejercicios



- 1) (ej301numVocales) Escribe una función que devuelva el número de vocales de una frase
- 2) (ej302toCani) EsCribE uNa FuNclón qUe TrAnSfOrMe uNa CaDeNa eN **CaNi**.
- 3) (ej303palindromo) Escribe una función que devuelva un booleano indicando si una palabra es palíndroma (se lee igual de izquierda a derecha que de derecha a izquierda, por ejemplo, “*Dábale arroz a la zorra el abad*”).

Buscando caracteres

- `cadena.indexOf(carácter)` → devuelve la posición de la primera ocurrencia de *carácter*
 - Si no la encuentra → -1
- `cadena.lastIndexOf(carácter)` → devuelve la posición de la última ocurrencia de *carácter*

```
var nombre = "Batman";
console.log(nombre.indexOf("a")); // 1
console.log(nombre.indexOf("2")); // -1
console.log(nombre.lastIndexOf("a")); // 4
```

Ejercicios

- 4) (ej304dosVeces) Escribe una función que sin recorrer la cadena, devuelva un booleano indicando si la frase contiene dos o más veces una determinada letra.

```
function dosVeces(frase, letra)
```

- 5) (ej305miIndexOf) Escribe una función que realice lo mismo que indexOf, pero permitiendo que la letra esté en minúsculas o mayúsculas.

```
function miIndexOf(frase, letra)
```

Ejercicios



- 6) (ej306anagrama) Escribe una función que reciba dos palabras, y averigüe si son *anagramas* (tiene el mismo número de letras pero en diferentes posiciones).
Por ejemplo, cosa y saco.

Ejercicios

7) (`ej307contrasena`) Escribe un función que a partir de una contraseña, devuelva el nivel de protección:

- MD → muy débil: contiene solo números y tiene menos de 8 caracteres.
- D → débil: contiene solo letras y tiene menos de 8 caracteres.
- F → fuerte: contiene letras y números y al menos 8 caracteres.
- MF → muy fuerte: contiene letras y/o números, caracteres especiales y al menos 8 caracteres.

Recortando cadenas

- `cadena.substring(inicio)` → Devuelve una subcadena a partir de la posición *inicio* de la *cadena*
- `cadena.substring(inicio, fin)` → Devuelve la subcadena comprendida entre las posiciones *inicio* y *fin* de la *cadena*
 - `cadena.slice(inicio, fin)` → Similar a `substring`, pero permite que *fin* sea negativo para comenzar a contar desde el final
 - `cadena.substr(inicio, long)` → Similar a `substring`, pero a partir de *inicio*, recorta *long* caracteres
- `cadena.trim()` → elimina los espacios del principio y final de la *cadena*

Ejemplo Recortando...

```
var nombre = "Batman";
console.log(nombre.indexOf("a")); // 1
console.log(nombre.indexOf("2")); // -1
console.log(nombre.lastIndexOf("a")); // 4

var frase = "Yo soy Batman";
console.log(frase.substring(3)); // "soy Batman"
console.log(frase.substring(3,6)); // "soy"

console.log(frase.slice(3)); // "soy Batman"
console.log(frase.slice(3,-3)); // "soy Bat"

console.log(frase.substr(3,3)); // "soy"
console.log(frase.substr(3,7)); // "soy Bat"

var fraseConEspacios = "    susto    o muerte    "; // "susto    o muerte"
console.log(fraseConEspacios.trim());
```

Ejercicios



9)(ej309miSubstr) Escribe una función que haga lo mismo que `substr` pero haciendo uso de `substring`.

```
function miSubstr(frase, inicio, long)
```

10)(ej310negSubstring) Escribe una función que realice lo opuesto a `substring`, es decir, que devuelva lo que queda de la cadena tras recortar la subcadena

```
function negSubstring(frase, inicio, fin)
```

Código de un carácter

- Cada letra se representa mediante un código numérico (ASCII, UTF-8)
- `cadena.charCodeAt(pos)` → Devuelve el código de la letra que ocupa la posición `pos`
 - `"hola".charCodeAt(0); // 104`
- `String.fromCharCode(num)` → Devuelve la letra cuyo código es `num`
 - `String.fromCharCode(104) // h`

Ejercicios

11) (ej311esLetraABC) Escribe una función que indice si un carácter es una letra del abecedario

```
function esLetraABC(letra)
```

12) (ej312miIsNaN) Escribe una función que realice lo mismo que la función `isNaN`, pero haciendo uso de `charCodeAt`

```
function miIsNaN(frase)
```

No se puede utilizar `parseInt`

- Para ello, primero crear una función que averigüe si un carácter es un número

Problema Olimpiada



Problema 3: Número del César (20 puntos)

Debes implementar un programa que permita codificar o decodificar una cadena de texto utilizando el cifrado de César.

En este cifrado cada letra de la cadena de texto se sustituye por otra letra que se encuentra ' n ' posiciones adelante en el alfabeto, dando la vuelta si es necesario.

Por ejemplo, si n es 2, todas las A se transformarían en C, las B en D, las C en E, etc.

Para descifrar se realizaría la operación contraria. Se calcularía la letra que está ' n ' posiciones por detrás.

El programa solicitará si se desea cifrar o descifrar. A continuación, preguntará el número de César. Por último preguntará la cadena de texto a cifrar o descifrar.

Como salida, el programa mostrará la cadena resultante.

Ejemplo de ejecución:

```
¿Quieres cifrar o descifrar? (c/d): c
```

```
Número del César: 3
```

```
Cadena: abcdefghijklmnopqrstuvwxyz
```

```
Resultado: defghijklmnopqrstuvwxyzabc
```

Math

- *JavaScript* ofrece el objeto `Math` como un conjunto de utilidades para realizar operaciones matemáticas.
- Ofrece constantes: `pi`, `e`, ... y operaciones
- Siempre se escribe el nombre del objeto, el punto, y la propiedad/método
 - `Math.PI`, `Math.pow(2, 3)`

Operaciones matemáticas

- Potencia \rightarrow `Math.pow(base, exp)`
- Raíz cuadrada \rightarrow `Math.sqrt(num)`
- Redondear \rightarrow `Math.round(num)`,
`Math.ceil(num)`, `Math.floor(num)`
- Obtener el mayor \rightarrow `Math.max(num1, num2, ...)`
- Obtener el menor \rightarrow `Math.min(num1, num2, ...)`
- Número aleatorio en 0 y 1 \rightarrow `Math.random()`
- Más información en
https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Math

Ejercicios

13) (ej313ecuacion2g) Escribe una función que resuelva una ecuación de segundo grado:

```
function ecuacion2g(a,b,c)
```

27) (ej314digitoAleat) Escribe una función que devuelva de forma aleatoria un dígito

```
function digitoAleat()
```

27) (ej315letraAleat) Escribe una función que devuelva de forma aleatoria una letra

```
function letraAleat()
```

Ejercicios / String y Math



8) (ej308generadorPWD) Crea una aplicación generadora de contraseñas. Para ello, debe pedir la longitud de la contraseña y cuantos números y caracteres especiales debe contener. El resto se rellenará con letras.

Una vez generada, mostrar al usuario

- (ej308R) Con los mismos datos, mostrar 3 contraseñas diferentes

Ejercicios

14) (ej316adivina) Escribe un programa que pida números al usuario del 0 al 10 hasta que adivine el número que el ordenador ha fijado.

- (ej316R1) Reescribir la función para que además informe al usuario de si el número a adivinar es mayor o menor
- (ej316R2) Permite tres niveles de dificultad:
Fácil (del 1 al 10), **Medio** (del 1 al 100), **Difícil** (del 10 al 1000)
- (ej316R3) Al finalizar mostrar la cantidad de intentos necesarios y:
 - Si acierta a la primera, responder: “Menudo crack”
 - Entre 2 y 4 intentos: “Flipo”
 - Entre 6 y 8 intentos: “No está mal”
 - Más de 9 intentos: “Igual esto no es lo tuyo”

Problema Olimpiada



Problema 4: Piedra, Papel, Tijera (20 puntos)

Debes implementar un programa que permita jugar al juego de Piedra, Papel, Tijera contra el ordenador.

El ordenador deberá calcular cuál de las tres opciones quiere. Esto lo debe hacer de manera aleatoria.

Pista: La generación aleatoria de la opción del ordenador se puede realizar generando un número aleatorio, por ejemplo del 0 al 2. Si es 0 será Piedra, si es 1 será Papel y si es 2 será Tijera.

El programa le debe preguntar al usuario cuál es su opción (¡sin mostrar la suya!). Esto puede hacerse indicándole al usuario que introduzca un único carácter, como por ejemplo podéis ver en los ejemplos de ejecución. A continuación comprobará quién de los dos ha ganado y lo mostrará por pantalla.

Ejemplo de ejecución:

```
¿Piedra, papel o tijera? (p/a/t): p
El ordenador ha elegido: Papel
¡Has perdido!
```

Otro ejemplo de ejecución:

```
¿Piedra, papel o tijera? (p/a/t): t
El ordenador ha elegido: Papel
¡Has ganado!
```

Date

- *Timestamp Epoch* (desde el 1/1/70)

```
var hoy = new Date();  
var nochevieja = new Date(2014, 11, 31);  
var cenaNochevieja = new Date(2014, 11, 31,  
22, 30, 0);
```

- Obtener información mediante:

- `getFullYear()`: devuelve el año de la fecha con cuatro dígitos
- `getMonth()`: número del mes del año (de 0 a 11)
- `getDate()`: número de día del mes

Comparando fechas

- Podemos usar `< o >`

```
var cenaPreNochevieja = new Date(2014, 11, 30, 22, 30, 0);
var cenaNochevieja = new Date(2014, 11, 31, 22, 30, 0);
console.log( cenaPreNochevieja < cenaNochevieja ); // true
```

- Para comparar hay que pasar el objeto a su *timestamp* mediante `fecha.getDate()`

```
var cenaNochevieja = new Date(2014, 11, 31, 22, 30, 0);
var cenaNochevieja2 = new Date(2014, 11, 31, 22, 30, 0);
console.log( cenaNochevieja === cenaNochevieja2 ); // false
console.log( cenaNochevieja.getTime() ===
cenaNochevieja2.getTime() ); // true
```

Ejercicios

- 17) (ej317fechaQuincena) Escribe una función que a partir de una fecha, devuelva la fecha dentro de 15 días.
- 18) (ej318diaSemana) Escribe una función que a partir de una fecha, devuelva el día de la semana (por ejemplo, Lunes).
- 19) (ej319diasVida) Escribe una función que a partir de una fecha de nacimiento, devuelva los días vividos.