

UD3. Utilización de los objetos predefinidos del lenguaje

Desarrollo Web en Entorno Cliente

2º DAW - Curso 2021/2022

IES Doñana - Sanlúcar de Barrameda (Cádiz)

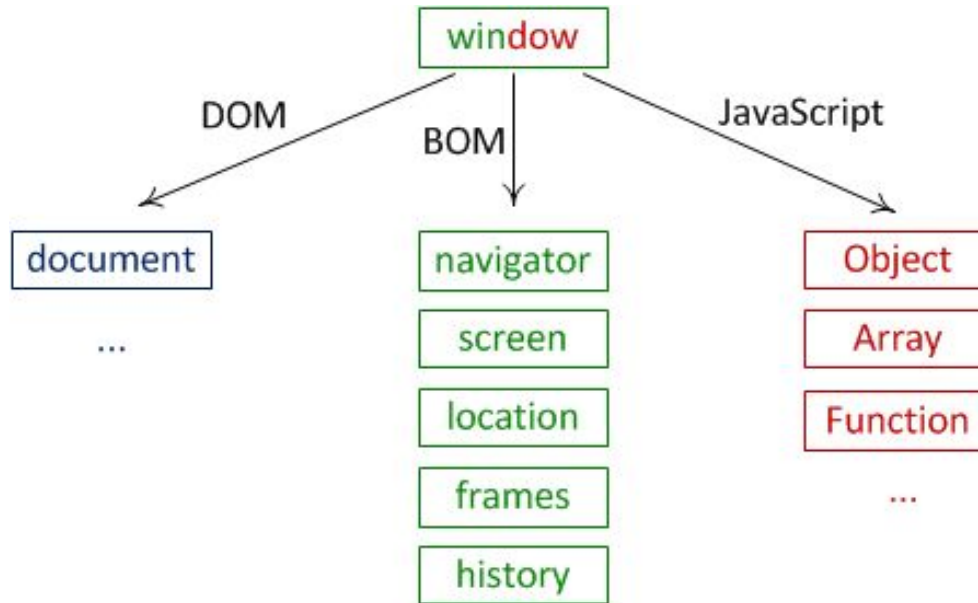
0. Objetivos de la unidad

RA3. Escribe código, identificando y aplicando las funcionalidades aportadas por los objetos predefinidos del lenguaje.

- a) Se han identificado los **objetos predefinidos del lenguaje**.
- b) Se han analizado los **objetos referentes a las ventanas del navegador y los documentos Web** que contienen.
- c) Se han escrito sentencias que utilicen los objetos predefinidos del lenguaje para cambiar el **aspecto del navegador y el documento** que contiene.
- d) Se han generado **textos y etiquetas** como resultado de la ejecución de código en el navegador.
- e) Se han escrito sentencias que utilicen los objetos predefinidos del lenguaje para **interactuar con el usuario**.
- f) Se han utilizado las características propias del lenguaje en documentos compuestos por **varias ventanas y marcos**.
- g) Se han utilizado «**cookies**» para almacenar información y recuperar su contenido.
- h) Se ha **depurado y documentado** el código.

0. Introducción

Ecosistema de objetos en JavaScript:



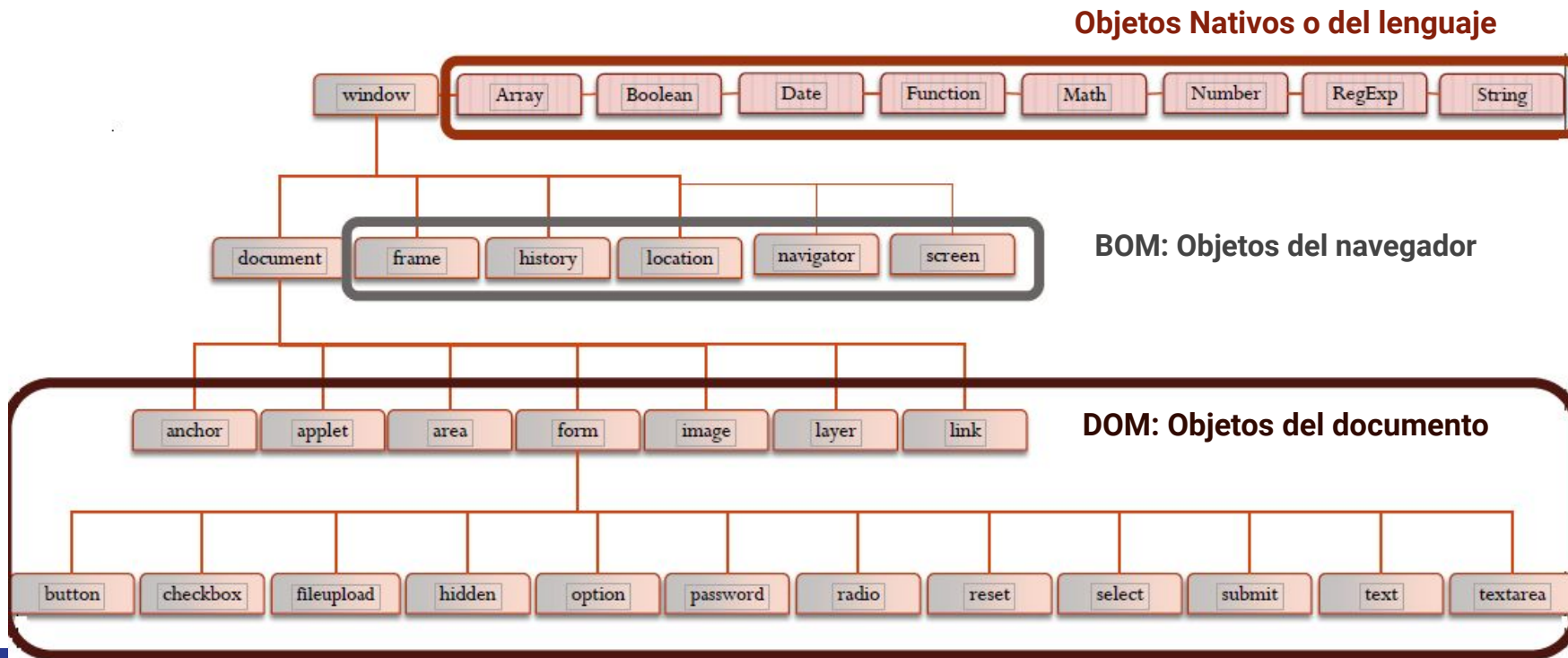
0. Introducción

Ecosistema de objetos en JavaScript:

- **Objetos Nativos o del lenguaje** : String, Number, Boolean, Date, Math, RegExp, Array, Function, Object.
- **Objetos del navegador (BOM)**: Window, Navigator, Screen, Location, History
- **Objetos del documentos (DOM)**:
 - Document, Events, Elements
 - Anchor, Area, Base, Body, Button, Form, Frameset, Image, Input, Link, Meta, Object, Option, Select, Style, Table, Textarea
- **Objetos definidos por el usuario: UD4**

0. Introducción

Ecosistema de objetos en JavaScript:



ÍNDICE


1. Objetos nativos de JS

- 1.1. Introducción a los objetos JS
- 1.2. El objeto String
- 1.3. El objeto RegExp
- 1.4. El objeto Number
- 1.5. El objeto Math
- 1.6. El objeto Date

2. Objetos del navegador (BOM)

- 2.1. El objeto Window
- 2.2. El objeto Navigator
- 2.3. El objeto Screen
- 2.4. El objeto History
- 2.5. El objeto Location
- 2.6. Gestión de ventanas

3. Objetos del documento (DOM)

- 3.1. El objeto Document
 - 3.2. Cookies
- 

1. Objetos nativos de JS

1.1. Introducción a los objetos

JavaScript proporciona una serie de objetos definidos nativamente que no dependen del navegador.

Un objeto se compone de:

- **Propiedades** → características
 - Constructor
 - Prototype: permite añadir propiedades y métodos al objeto
 - Ej: `Boolean.prototype.nuevaFuncion= function(){//código}`
- **Métodos** → funciones específicas que se pueden realizar con los objetos
 - `toString()`: Devuelve una representación como string del objeto.
 - `valueOf()`: Devuelve el valor primitivo
- **Eventos** → situaciones que pueden llegar a realizarse o no. Cada objeto reconoce una serie de eventos

1. Objetos nativos de JS

1.1. Introducción a los objetos

Para crear un objeto se utiliza la palabra clave **new**. Ejemplo:

- `var miObjeto = new Object();`

Aunque no se recomienda usar este constructor para los siguientes objetos:

- **String:** `var cadena = " ";`
- **Number:** `var numero = 3.14;`
- **Boolean:** `var condicion = true;`
- **Array:** `var arr=[];`
- **RegExp:** `var patron=/ /;`
- **Function:** `var foo = function();`
- **Object:** `var miObj = { };`
- **Math** no se puede declarar con "new" porque es un objeto global
- Solo **Date** se declara con "new"

1. Objetos nativos de JS

1.1. Introducción a los objetos

```
//1. Definir y crear un objeto simple utilizando un literal.
var personal = {
  nombre:"Ada",
  apellido:"Lovelace",
  ano: 1815};

//2. Definir y crear un objeto simple utilizando la palabra new.
var persona2 = new Object();
persona2.nombre = "Charles";
persona2.apellido = "Babbage";
persona2.ano = 1791;

//3. Definir un constructor de un objeto, y crear objetos del
    tipo construido.
function Persona (nom, ape, an){
  this.nombre = nom;
  this.apellido = ape;
  this.ano = an;
}
var ada = new Persona ("Ada", "Lovelace", 1815);
var babbage = new Persona ("Charles", "Babbage", 1791);

/* This (palabra reservada):
   - Hace referencia al propietario de la función que está
     invocándose.
   - Hace referencia al objeto donde la función es un método.*/
```

1. Objetos nativos de JS

1.2. El objeto String

Permite manipular cadenas de texto.

Propiedades y métodos del objeto String:

- Siempre devuelven otro string
- En rojo, wrapped methods

Métodos			
<code>anchor()</code>	<code>fixed()</code>	<code>localeCompare(string)</code>	<code>strike()</code>
<code>big()</code>	<code>fontcolor()</code>	<code>match(regex)</code>	<code>sub()</code>
<code>blink()</code>	<code>fontsize()</code>	<code>replace(searchvalue, newvalue)</code>	<code>substr(start, length)</code>
<code>bold()</code>	<code>fromCharCode(num)</code>	<code>search(searchvalue)</code>	<code>substring(start, end)</code>
<code>charAt(pos)</code>	<code>indexOf(searchvalue, start)</code>	<code>slice(start, end)</code>	<code>sup()</code>
<code>charCodeAt(pos)</code>	<code>italics()</code>	<code>small()</code>	<code>toLowerCase()</code>
<code>concat(string, string, ...)</code>	<code>lastIndexOf(searchvalue, start)</code>	<code>split(separator, limit)</code>	<code>toUpperCase()</code>
<code>startsWith(string, length)</code>	<code>endsWith(string, length)</code>	<code>includes(string)</code>	<code>trim()</code>

1. Objetos nativos de JS

1.2. El objeto String

Ejercicio 3A01:

Realiza un script en el que un usuario introduzca una frase (esta pueda contener todo tipo de caracteres), ir mostrando todos los caracteres uno a uno, excepto si es un número que lo obviará. Usa el método `charAt()`

Ejercicio 3A02:

Solicitar una frase desde teclado, almacenar cada palabra en un array y mostrar los siguientes datos:

- La primera palabra de la frase
- La ultima palabra de la frase
- El número de palabras de la frase



1. Objetos nativos de JS

1.3. El objeto RegExp

Una expresión regular es un objeto que describe un patrón de caracteres.

Sintaxis básica:

- /patron/modificadores
- Modificadores:
 - g: Busca todas las apariciones del patrón en lugar de parar en la primera coincidencia:

```
var str = "Is this all there is?";  
var patt1 = /is/g;           //is,is
```

- i: Ignora mayúsculas y minúsculas en la búsqueda del patrón:

```
var str = "Visit W3Schools";  
var patt1 = /w3schools/i;    //W3Schools
```

- m: Permite la búsqueda multilínea:

```
var str = "\nIs th\nis it?";  
var patt1 = /is/m;           //is
```

1. Objetos nativos de JS

1.3. El objeto RegExp

El objeto RegExp:

- Patrones:
 - Corchetes: Se usan para buscar un rango de caracteres
 - [abc], [^abc], [0-9], [^0-9], (x|y)
 - Metacaracteres: Son caracteres con un significado especial
 - ., \w, \W, \d, \D, \s, \S, \b...
 - Cuantificadores:
 - +, *, ?, {x}, \$, ?=...



1. Objetos nativos de JS

1.3. El objeto RegExp

Ejercicio 3A03

Solicita el ingreso de la coordenada de un punto. El formato a ingresar por teclado es:

- (999,999)
- Los números pueden tener entre 1 y 3 dígitos.

Usa el método `test()`



1. Objetos nativos de JS

1.4. El objeto Number

El objeto Number permite realizar tareas relacionadas con tipos de datos numéricos:

Propiedades
MAX_VALUE
MIN_VALUE
NaN
NEGATIVE_INFINITY
POSITIVE_INFINITY

Devuelve String

Method	Description
toString()	
toExponential()	<pre>var x = 9.656; x.toExponential(2); // returns 9.66e+0</pre>
toFixed()	<pre>var x = 9.656; x.toFixed(0); // returns 10 x.toFixed(2); // returns 9.66</pre>
toPrecision()	<pre>var x = 9.656; x.toPrecision(); // returns 9.656 x.toPrecision(2); // returns 9.7</pre>
valueOf()	Returns a number as a number

1. Objetos nativos de JS

1.5. El Objeto Math

El objeto Math permite realizar operaciones matemáticas complejas en JavaScript:

Propiedades
E
LN2
LN10
LOG2E
LOG10E
PI
SQRT1_2
SQRT2

Métodos		
abs()	exp()	random()
acos()	floor()	round()
asin()	log()	sin()
atan()	max()	sqrt()
ceil()	min()	tan()
cos()	pow()	sign()

1. Objetos nativos de JS

1.5. El Objeto Math

Ejercicio 3A04

Genera 10 números aleatorios enteros entre 0 y 20.



1. Objetos nativos de JS

1.6. El Objeto Date

El objeto Date permite trabajar con fechas y tiempos en JavaScript.

Los objetos Date se crean `new Date()`

- `new Date();`
- `new Date(milliseconds);`
- `new Date(dateString);`
- `new Date(year, month, day, hours, minutes, seconds, milliseconds);`

Particularidades:

- Origen tiempos milisegundos: 1 enero de 1970
- Días de la semana y meses empiezan en 0 (domingo y enero)

Ejemplo:

```
let fActual = new Date();
let fFin = new Date(2022, 00, 01);
alert("La fecha actual es: " + fActual);
alert("El fin de año es el: " + fFin);
let tRestante = fFin - fActual;
alert("Quedan " + tRestante + " milisegundos para que termine el año");
```

1. Objetos nativos de JS

1.6. El Objeto Date

Métodos				
<code>getDate()</code>	<code>getTime()</code>	<code>getUTCMonth()</code>	<code>setMonth()</code>	<code>setUTCMonth()</code>
<code>getDay()</code>	<code>getTimezoneOffset()</code>	<code>getUTCSeconds()</code>	<code>setSeconds()</code>	<code>setUTCSeconds()</code>
<code>getFullYear()</code>	<code>getUTCDate()</code>	<code>parse()</code>	<code>setTime()</code>	<code>toDatestring()</code>
<code>getHours()</code>	<code>getUTCDay()</code>	<code>setDate()</code>	<code>setUTCDate()</code>	<code>toLocaleDateString()</code>
<code>getMilliseconds()</code>	<code>getUTCFullYear()</code>	<code>setFullYear()</code>	<code>setUTCFullYear()</code>	<code>toLocaleTimeString()</code>
<code>getMinutes()</code>	<code>getUTCHours()</code>	<code>setHours()</code>	<code>setUTCHours()</code>	<code>toLocaleString()</code>
<code>getMonth()</code>	<code>getUTCMilliseconds()</code>	<code>setMilliseconds()</code>	<code>setUTCMilliseconds()</code>	<code>toTimeString()</code>
<code>getSeconds()</code>	<code>getUTCMinutes()</code>	<code>setMinutes()</code>	<code>setUTCMinutes()</code>	<code>toUTCString()</code>
<code>now()</code>	<code>toJSON()</code>			

1. Objetos nativos de JS

1.6. El Objeto Date

Ejercicio 3A05

Muestra los meses, días y años que tienes, haciendo uso de los métodos del objeto Date

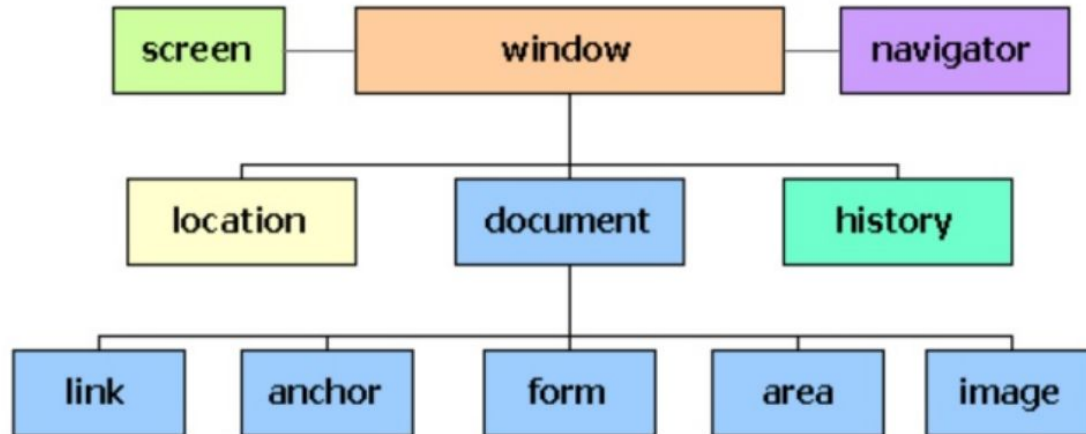


2. Objetos del navegador (BOM)

Existe otro tipo de objetos que permiten manipular diferentes características del navegador en sí mismo, conformando lo que se denomina el modelo de objetos del navegador (BOM).

El BOM (Browser Object Model) es una convención específica implementada por los navegadores para poder usar métodos y propiedades de forma uniforme.

Representa al navegador, no solo al documento, y su elemento raíz es el objeto window



2. Objetos del navegador (BOM)

2.1. El Objeto Window

- Se considera el objeto más importante de JavaScript.
- Representa una ventana abierta en un navegador en la que se muestra un documento del DOM
- Permite gestionar las ventanas del navegador, controlando aspectos como las dimensiones de la ventana, las barras laterales de desplazamiento, la barra de estado...
- Todos los objetos que contenga el código de la ventana se hacen miembros del objeto window:
 - Funciones: son sus métodos
 - Variables globales :son sus propiedades

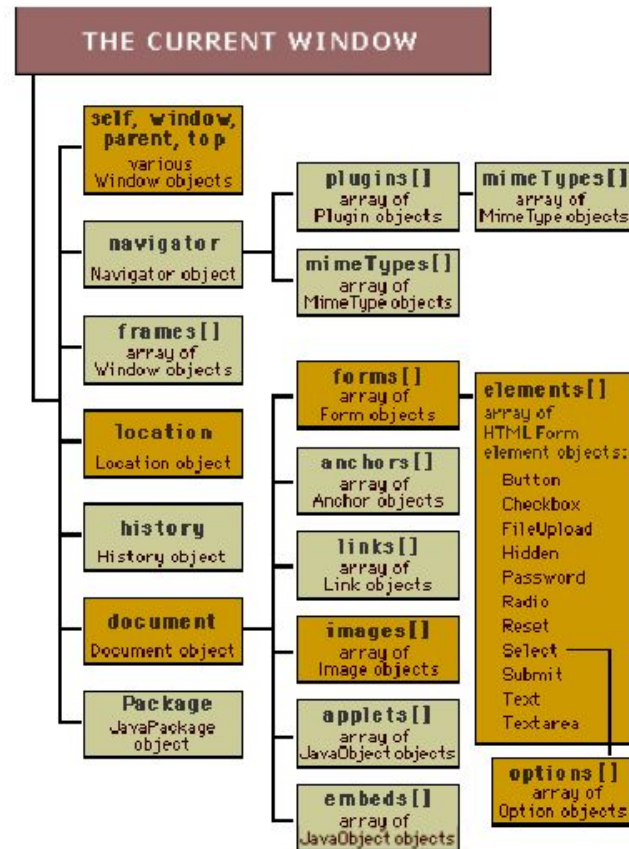


2. Objetos del navegador (BOM)

2.1. El Objeto Window

- Cada una de las pestañas abiertas tendrá un objeto window asociado, pero métodos como window.resizeTo se aplicarán a la ventana de la aplicación y no a la pestaña específica.
- Es un objeto implícito, con lo cual no es necesario nombrarlo para acceder a los objetos que se encuentran debajo de su nivel de jerarquía:

`window.document.cookie == document.cookie`



2. Objetos del navegador (BOM)

2.1. El Objeto Window

Métodos y propiedades del objeto Window

Ojo: NO todos los métodos son compatibles con todos los navegadores. Se comprueba en W3Schools

Métodos			Propiedades		
alert()	clearInterval()	setInterval()	closed	location	pageYoffset
getComputedStyle()	clearTimeout()	setTimeout()	defaultStatus	screenLeft / screenX	parent
blur()	moveTo()	scrollBy()	document	screenTop / screenY	screen
close()	moveBy()	scrollTo()	frames	name	scrollbars
confirm()	open()	stop()	history	opener	self
find()	prompt()	resizeBy()	innerHeight	outerHeight	status
focus()	resizeTo()	getSelection()	innerWidth	outerWidth	toolbar
matchMedia()	print()		length	pageXoffset	top
			localStorage	sessionsStorage	

2. Objetos del navegador (BOM)

2.1. El Objeto Window

Propiedades básicas:

- Nombre de la ventana: Por defecto no tiene pero se le puede asignar

```
window.name = "Mi Ventana";
```

- Tamaño de la ventana con toolbar y scrollbar:

```
window.outerHeight;  
window.outerWidth;
```

- Tamaño del viewport (sin toolbar y scrollbar):

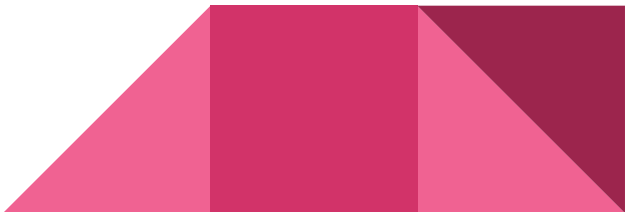
```
window.innerHeight;  
window.innerWidth;
```

- Scroll horizontal y Vertical

```
window.pageXOffset; window.scrollX;  
window.pageYOffset; window.scrollY;
```

- Distancia desde la esquina superior izquierda:

```
window.screenX;  
window.screenY;
```



2. Objetos del navegador (BOM)

2.1. El Objeto Window

Propiedades con iframes:

- **window.frame:** Devuelve todos los elementos iframe de la ventana
- **window.frameElement:** Devuelve el iframe en el que la ventana está insertada. Si no está insertada en un iframe devuelve null.
- **window.length:** Devuelve el número de frames de la ventana



2. Objetos del navegador (BOM)

2.1. El Objeto Window

Propiedades con otras ventanas:

- **nuevaVentana.closed:** Devuelve true si la ventana está cerrada
- **nuevaVentana.opener:** Devuelve referencia con la ventana que creo la ventana actual.
- **nuevaVentana.parent:** Devuelve la ventana padre de la ventana activa
- **nuevaVentana.self:** Referencia a la ventana actual



2. Objetos del navegador (BOM)

2.1. El Objeto Window

Métodos:

- **window.open():** Abre una nueva ventana.
`window.open(URL, name, specs, replace)`
- **window.close():** Cierra una ventana.
`window.close()`
- **window.resizeBy():** Redimensiona una ventana un número de píxeles respecto a su tamaño actual. También esta `resizeTo` con valores totales.
`resizeBy(width, height)`
- **windows.moveBy():** Mueve una ventana una determinada cantidad de pixeles. `moveTo()` desplaza la ventana a una posición concreta
`window.moveBy(x, y)`
- **Muchos más: scrollBy()....**

2. Objetos del navegador (BOM)

2.1. El Objeto Window

Métodos - Instrucciones de tiempo:

- **window.setTimeout():** Ejecuta función pasados una determinada cantidad de ms.

```
setTimeout(function, milliseconds, param1, param2, ...)
```

- **window.clearTimeout():** Cancela ejecución de un timeout (retardo).

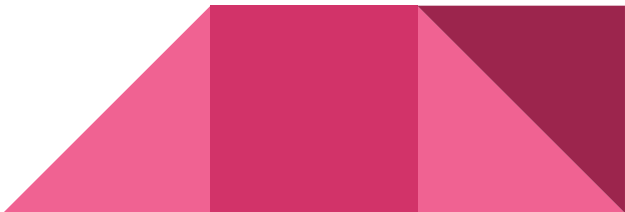
```
clearTimeout(id_of_settimeout)
```

- **window.setInterval():** Ejecuta una función cada cierto intervalo de ms.

```
setInterval(function, milliseconds, param1, param2, ...)
```

- **windows.clearInterval():** Cancela la ejecución de un setInterval.

```
clearInterval(var)
```



ÍNDICE

1. Objetos nativos de JS

- 1.1. Introducción a los objetos JS
- 1.2. El objeto String
- 1.3. El objeto RegExp
- 1.4. El objeto Number
- 1.5. El objeto Math
- 1.6. El objeto Date

2. Objetos del navegador (BOM)

- 2.1. El objeto Window
- 2.2. El objeto Navigator
- 2.3. El objeto Screen
- 2.4. El objeto History
- 2.5. El objeto Location
- 2.6. Gestión de ventanas

3. Objetos del documento (DOM): cookies

- 3.1. El objeto Document
 - 3.2. Cookies
- 