

Desarrollo de Aplicaciones Web

DOCENTE: Daniel López Lozano



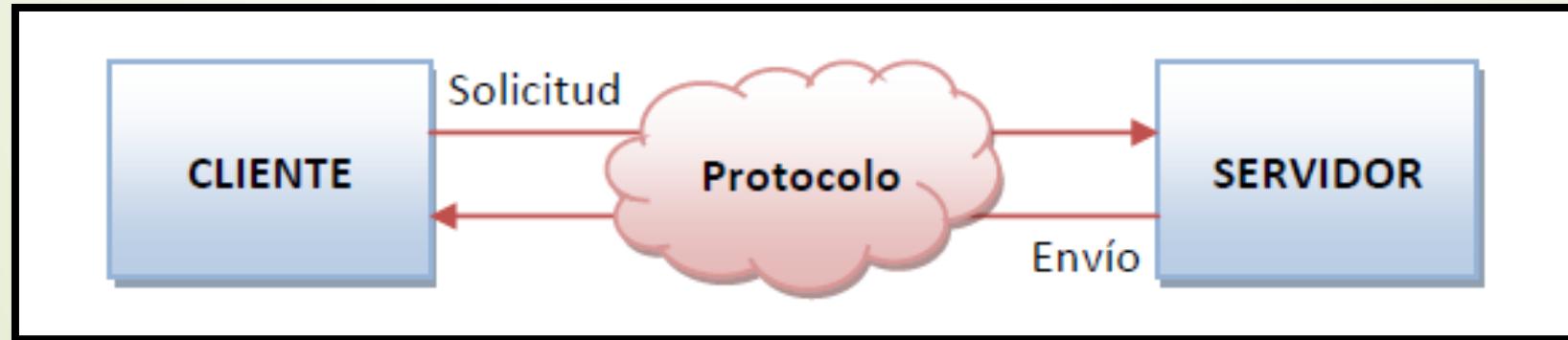


Tema 1. Introducción al Desarrollo en Entorno Cliente

Índice de contenidos

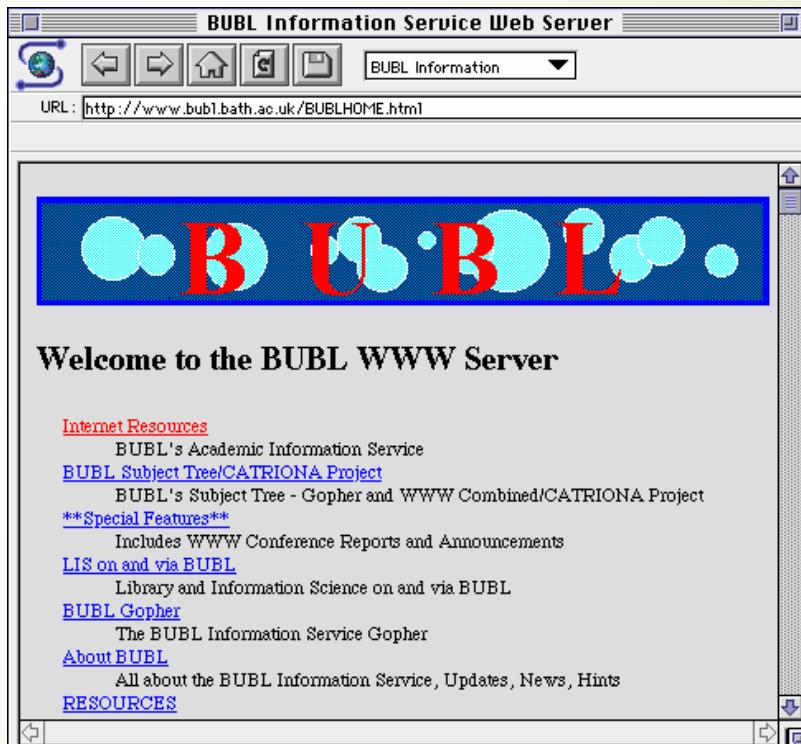
- Definición de Front-End y Back-End.
- Introducción a JavaScript:
 - Integrar JavaScript en HTML.
 - Sintaxis.
 - Variables y Operadores.
 - Entrada y Salida.
 - Uso de la POO.

- La configuración arquitectónica más habitual es la denominada Cliente-Servidor.
- El cliente es un componente **consumidor** de servicios.
- El servidor es un proceso **proveedor** de servicios.

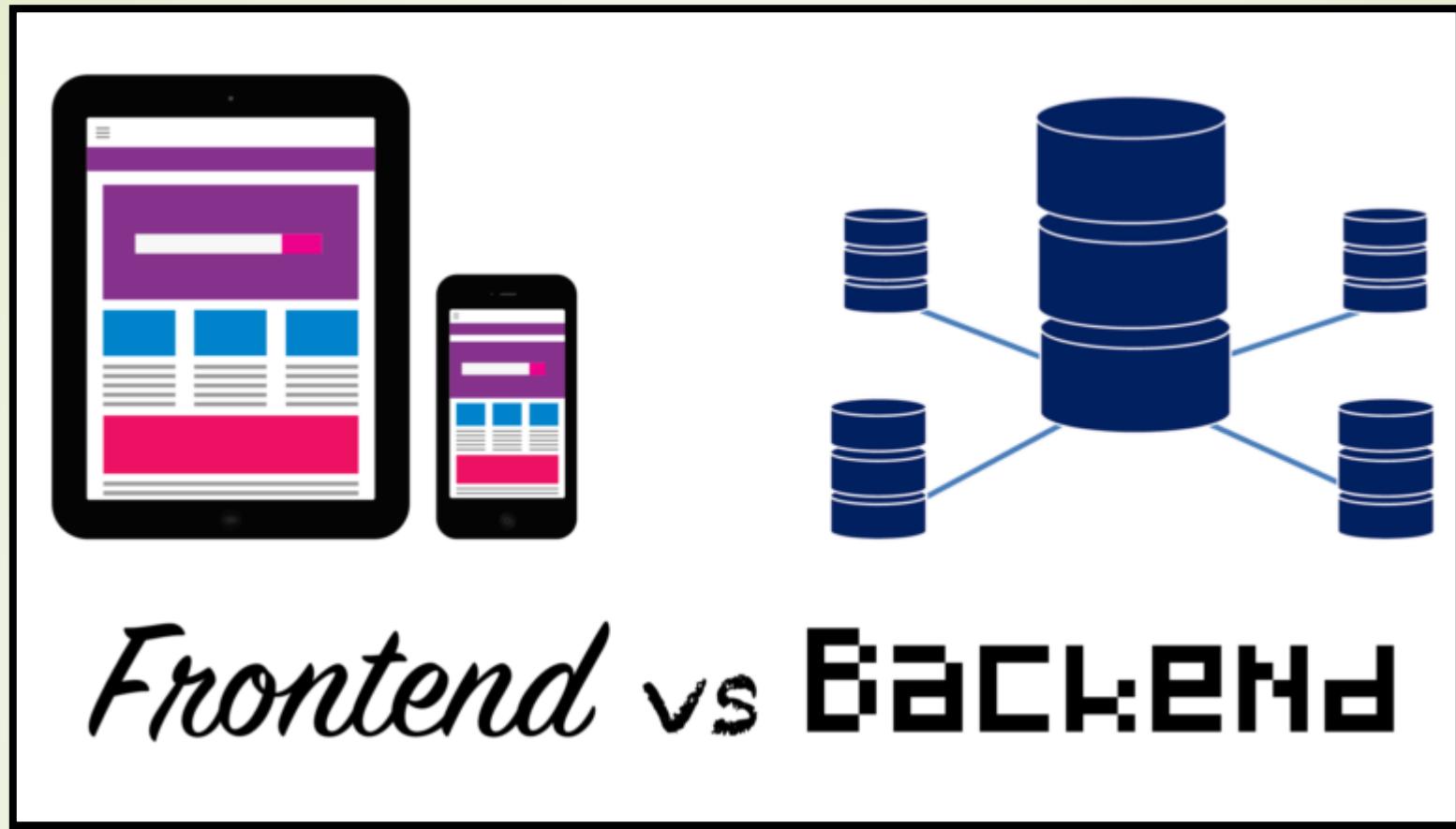


- ❑ El componente software que se utiliza en el cliente es el navegador web.
- ❑ Permite acceder al contenido ofrecido por los servidores de Internet sin la necesidad de que el usuario instale un nuevo programa.
- ❑ El navegador web permite a un usuario acceder y visualizar a un recurso de un servidor Web mediante una dirección URL (Universal Resource Locator).
- ❑ En la actualidad dicho recurso puede incluir programas ejecutables.

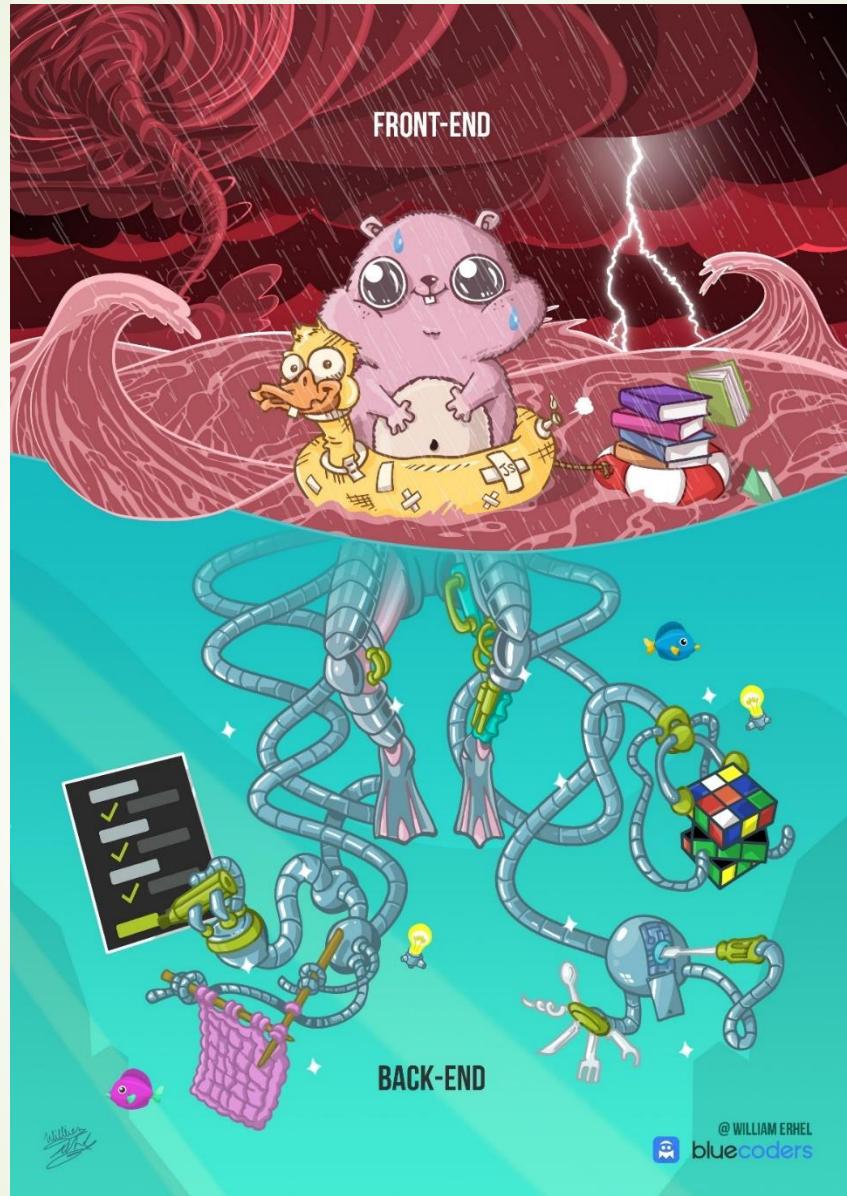
- Existen multitud de navegadores web disponibles:
 - Mosaic.
 - Netscape Navigator (después Communicator).
 - Internet Explorer.
 - Mozilla Firefox.
 - Google Chrome.
 - Safari.
 - Opera.

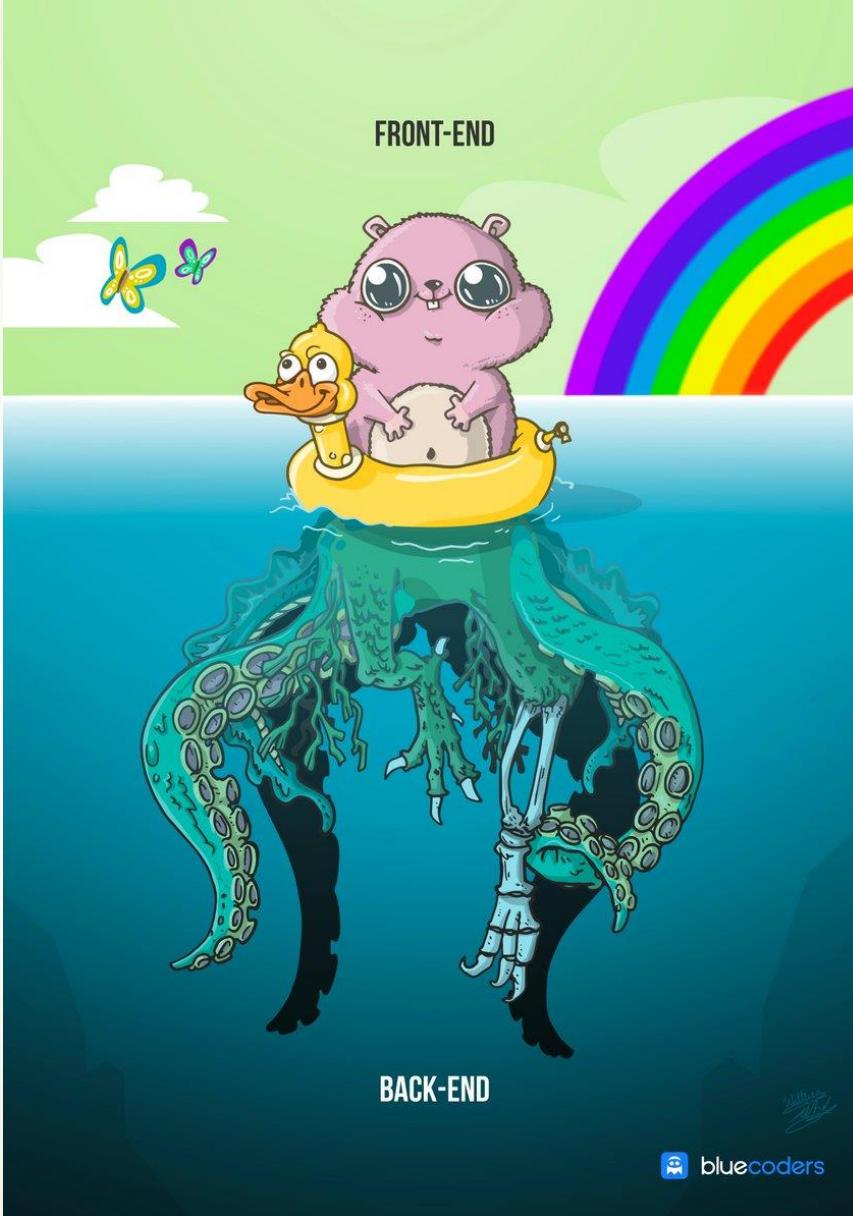


- En toda aplicación web existen dos partes distinguibles. Se trata del front-end y el back-end.
- El front-end es la parte del software que interactúa con los usuarios y el back-end es la parte que procesa la información que entra desde el front-end.
- El front-end sea el responsable de recolectar los datos de entrada del usuario, y los transforma ajustándolos a las especificaciones que demanda el back-end para poder procesarlos, devolviendo generalmente una respuesta que el front-end recibe y expone al usuario.



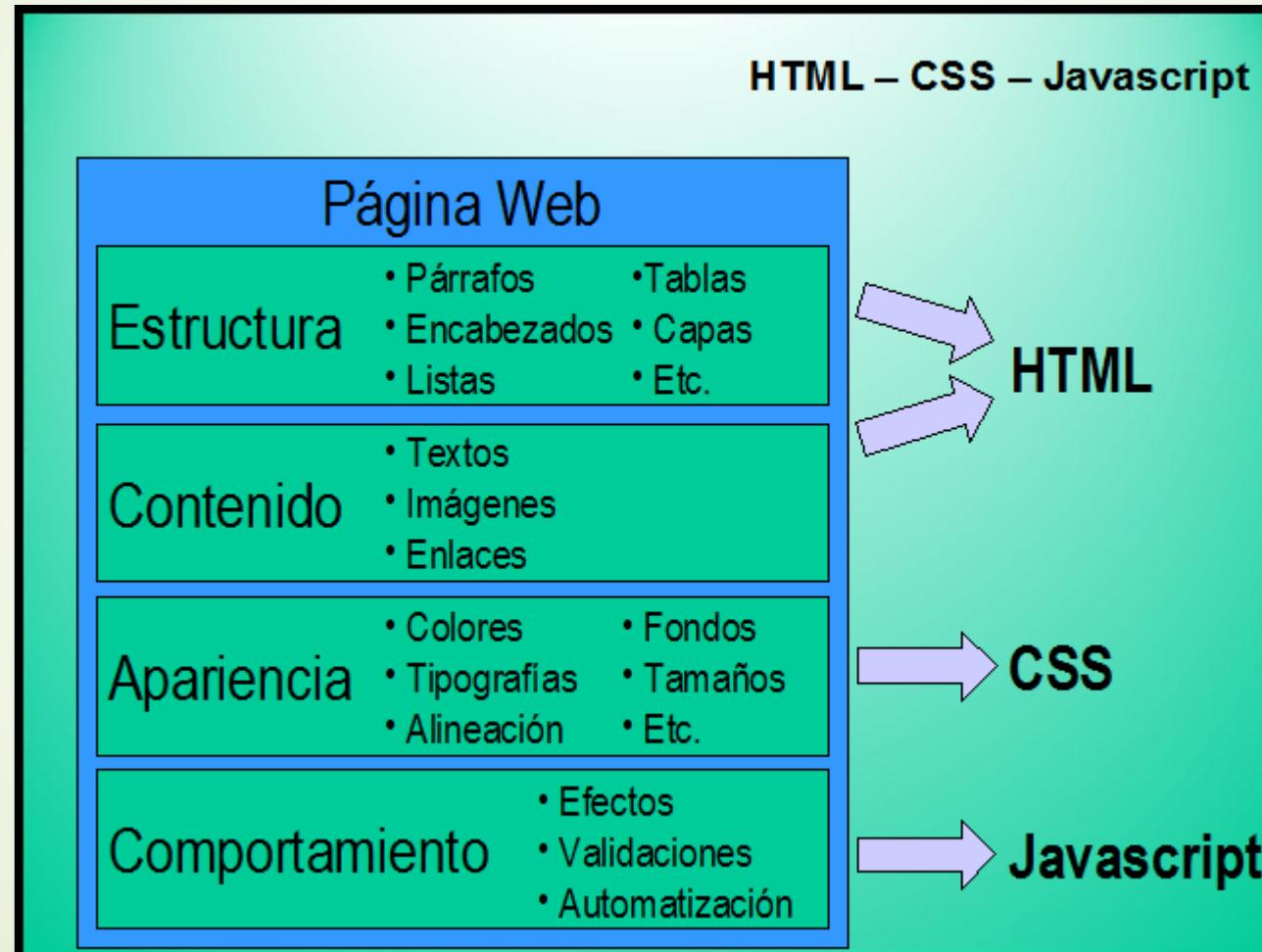
- El back-end es el encargado de procesar la información, almacenarla y recuperarla correctamente, dando un valor añadido a los datos recogidos por el front-end



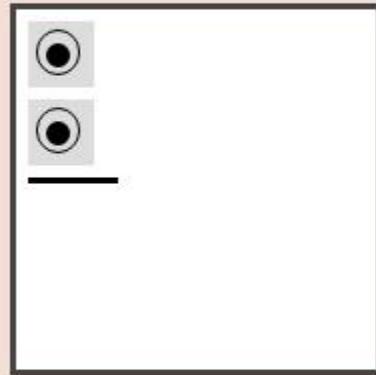


- Las tecnologías front-end y backend poseen características que las hacen complementarias más que adversarias.
- Por ejemplo, para recoger información de un formulario y grabarla en una base de datos (back-end), tendrá más sentido **chequear los datos** en el front-end antes de enviarlos al servidor.
- Sera mucho más **eficiente** y el usuario se sentirá menos **frustrado** al llenar los datos en el formulario.

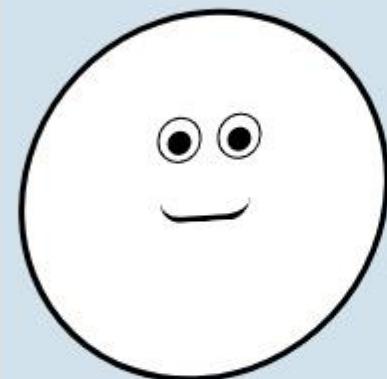
- El front-end se divide en diversos lenguajes cada uno con sus funciones.



HTML



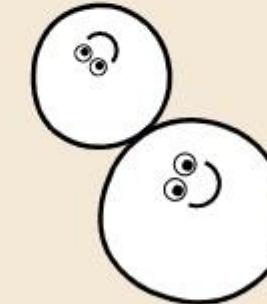
HTML + CSS



HTML + CSS + JS



Click here to PLAY!



Renacimiento de JavaScript con la UX



- ❑ Vamos a usar JavaScript porque es el lenguaje de script más utilizado en el lado del cliente, y está soportado mayoritariamente por todas las plataformas.
- ❑ Brendan Eich que trabajaba en el navegador Netscape desarrollo un lenguaje de programación llamado **Mocha**, que posteriormente se denominó **LiveScript**.
- ❑ Coincidiendo con la popularidad del lenguaje Java acabo denominándose JavaScript.

- ❑ ¿Entonces que es ECMA Script?
- ❑ Debido a que Microsoft (JScript) y otros sacaron sus propias versiones, los autores originales desarrollaron un estándar para la ECMA que es el adoptado en la actualidad por todos los navegadores.
- ❑ De hecho TypeScript que ha sido desarrollado también por Microsoft es compatible con JavaScript, porque los dos se basan en ECMA Script.

- ❑ El objetivo de JavaScript es manipular el código HTML y CSS, borrando, mostrando, modificando y añadiendo elementos y su presentación como respuesta a una interacción del usuario con la aplicación web.
- ❑ Existen 3 formas de incluir JavaScript en nuestro código HTML:
 - ✓ Incrustado en etiquetas HTML.
 - ✓ La etiqueta <script>.
 - ✓ Ficheros externos js.

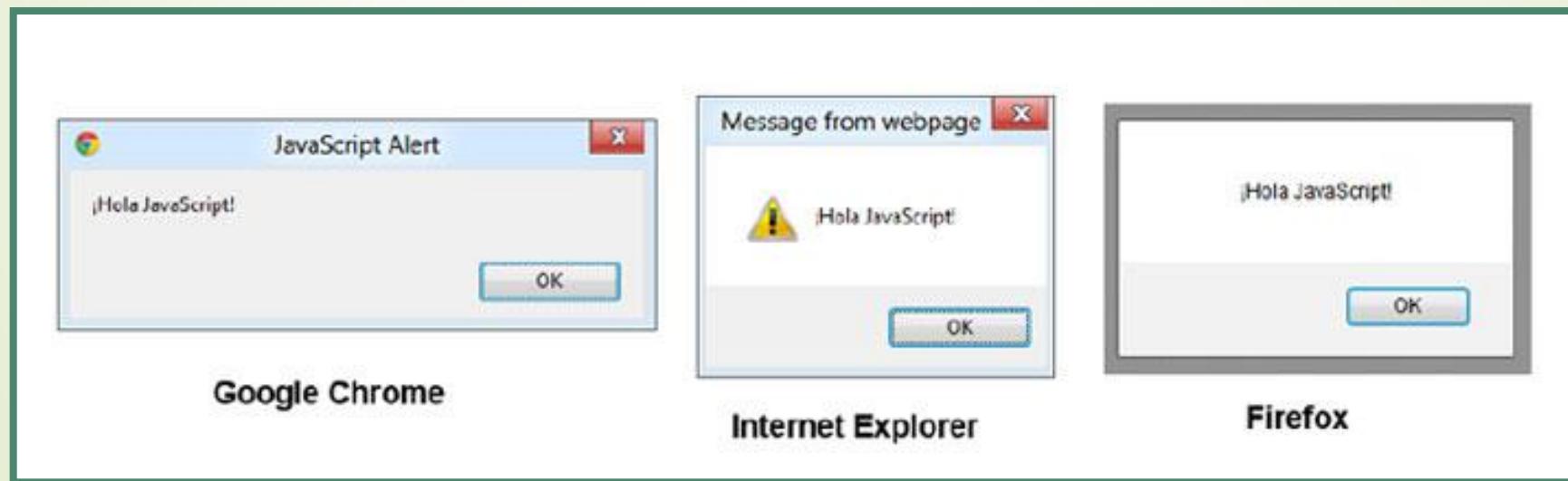
Incrustado en etiquetas HTML

```
<body><p onclick="alert('Hola Mundo')">Texto de prueba.</p></body>
```

- Se usan para añadir comportamiento a los elementos HTML directamente.
- Mezcla el HTML con el JavaScript y complica el mantenimiento de la página.

alert

- ❑ Muestra una ventana con el mensaje que le hayamos indicado y es una forma básica de comunicación.



Etiqueta <script>

```
<head>
    <script type="text/javascript">
        alert("Hola Mundo");
    </script>
</head>
```

- Se recomienda incluir las en la etiqueta head, aunque veremos algunas excepciones necesarias.

Ficheros externos .js

Fichero hmtl:

```
<head>
    <script type="text/javascript" src="prueba.js"></script>
</head>
```

Fichero prueba.js:

```
alert("Hola mundo");
```

- Es la que utilizaremos nosotros en clase

- La sintaxis de JavaScript en la parte básica es similar a la de C, C++, Java y otros lenguajes relacionados.
- No se tiene en cuenta espacios en blanco y saltos de línea.
- Distingue las mayúsculas y minúsculas.
- Se pueden incluir comentarios con // y /* */
- Existen palabras reservadas del lenguaje.

- ❑ No es necesario establecer el tipo de datos en la declaración de las variables que se ajusta automáticamente.
- ❑ Los tipos de datos básicos existentes son:
 - ✓ Numérico, incluyendo los números decimales.
 - ✓ Lógico o booleano, true o false.
 - ✓ Tipo cadena o string, que almacena texto.

- El nombre de la variable solo puede contener letras, números y los símbolos “_” y “\$”, además de no empezar por numero. Para declarar variables se usa la palabra let.
- Existen distintas maneras de declarar variables.

```
let primera;
let puntuacion,record,jugador;
let correcta="ok";
let _correcta2=8;
let $correctisima=4.78;
let correcta_de_mas=true;
let 1incorrecta="blue";
let puntuacion=0,record=5000,jugador="Juan";

alert(primer);
alert(puntuacion);
...
```

- Si probamos el código anterior vemos que primera no tiene valor y muestra undefined.
- Existen operadores aritméticos: +,-,*/,%, ++,--,+=,-=, ...
- De comparación: ==,!<,>,<,>=,<=
- Lógicos: !(NOT), &&(AND) y || (OR)

```
Let cantidad,precio,total;
cantidad=100;
precio=5;
total=cantidad*precio;
alert(total);
```

Más ejemplos de operadores matemáticos

```
alert(4+6-2);
let num1=4,num2=5.4;
let suma=num1+num2;
let aumenta_cinco=num2+5;
let quita=num1-3;
let escrito="3";
let resultado=num1+escrito;
alert(resultado);
alert(num1+4);
alert(num1);
let num2=num2+7;
alert(num2);
```

¿Cual es el valor de cada variable al ejecutar el código?

```
let A,B,C;  
A=5;  
B=7;  
C=4;  
A=A + B + C;  
B=C / 2;  
A=A*B + A/C;  
C=A+(B-C)-B;
```

- Otros operadores de asignación que incluyen operandos.

Operador	Utilización	Expresión equivalente
<code>+=</code>	<code>op1 += op2</code>	<code>op1 = op1 + op2</code>
<code>-=</code>	<code>op1 -= op2</code>	<code>op1 = op1 - op2</code>
<code>*=</code>	<code>op1 *= op2</code>	<code>op1 = op1 * op2</code>
<code>/=</code>	<code>op1 /= op2</code>	<code>op1 = op1 / op2</code>
<code>%=</code>	<code>op1 %= op2</code>	<code>op1 = op1 % op2</code>

¿Cuál es el valor de cada variable al ejecutar el código?

```
x=0;  
y=4.5;  
z=3*y;  
x=x+1;  
x++;  
x--;  
y=y+5.6;  
y+=5.6  
x=x+y;  
x+=y;  
z=4%3;  
z=x%6;
```

```
x=3;  
alert(x+4);  
x=x+4;  
alert(x);
```

¿Cual es el valor de cada variable al ejecutar el código?

```
Let A,B,C,D,E,F,G,H;  
A=5;  
B=3;  
C=(A>B);  
D=(A<B);  
E=(A>=B);  
F=(A<=B);  
G=(A==B);  
H=(A!=B);  
alert(A);  
...
```

```
4==4  
4>3  
x=5  
x<=7  
x<5  
edad>=18  
x==5  
jornada=="media"  
jornada!="completa"  
x%2==0
```

Operadores lógicos o booleanos

AND: sirve para **UNIR**

OR: sirve para **ELEGIR**

NOT: sirve para **EXCLUIR**

¿Cual es el valor de cada variable al ejecutar el código?

```
A=5;  
B=3;  
C=6;  
D=1;  
E=(C>A && D<B);  
F=(C>A || D>B);  
G=( !F );
```

Tipo booleano en el mundo real

```
continuar=true;
!continuar

x>=1 && x<=10
x>45 && sexo=="masculino"
x>45 || sexo=="masculino"
letra=="A" || letra=="E" || letra=="I" || letra=="O" || letra=="U"
continuar && credito>0

encontrado=false;
fin=true;
encontrado || fin
```

Tipo booleano en el mundo real

```
hermano=(nombre=="Miguel");
arquimedes=(frase=="Eureka");
suspenso=!aprobado;
seguir=!fin;
correcto=(mes>=1 && mes<=2);
comprar_coche=(ahorros>precio_coche || loteria==true);
grado_sup=(bachillerato==true || prueba_acceso==true);
alert(isNaN(34));
alert(isNaN("dani"));
esNumero(!isNaN(10));
```

Operaciones básicas de string

```
let texto="Un texto con 'comillas' dentro";
let otro_texto='Otro texto tambien con "comillas" dentro';
let numero_letras=text.length;

let prueba1="Hola "+ "Mundo";
let prueba2="Nota: "+ 10;
let prueba3="7"+5;
```

- JavaScript cuenta con el tipo **Array** como un tipo básico.
- Un **Array** es una **colección de variables**, que se engloba en una misma variable
- Los **Arrays** son **dinámicos** por lo que pueden cambiar de tamaño y pueden incluir cualquier tipo de datos.
- Los **Arrays** empiezan a numerarse en 0 y tiene la propiedad **length** como el tipo **string**.
- De hecho un **string** es un caso particular de un **Array**.

- De manera práctica son conjuntos de datos almacenados bajo el mismo nombre.
- Una representación gráfica de un array seria la siguiente

Posición	0	1	2	3
Valor	25	36	21	58

- Para crear el array representado en la imagen anterior es necesario este código.

```
Let numeros=[];
numero[0]=25;
numero[1]=36;
numero[2]=21;
numero[3]=58;
```

- Un array en JavaScript no tiene un tamaño predefinido, con lo que puede tener tantos valores como sea necesario.

Operaciones básicas de un Array

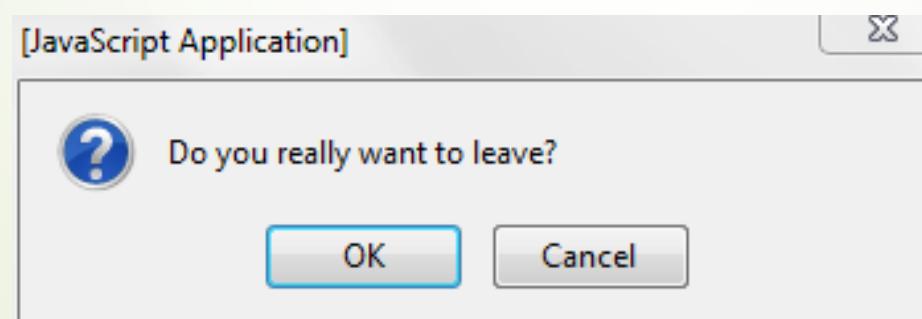
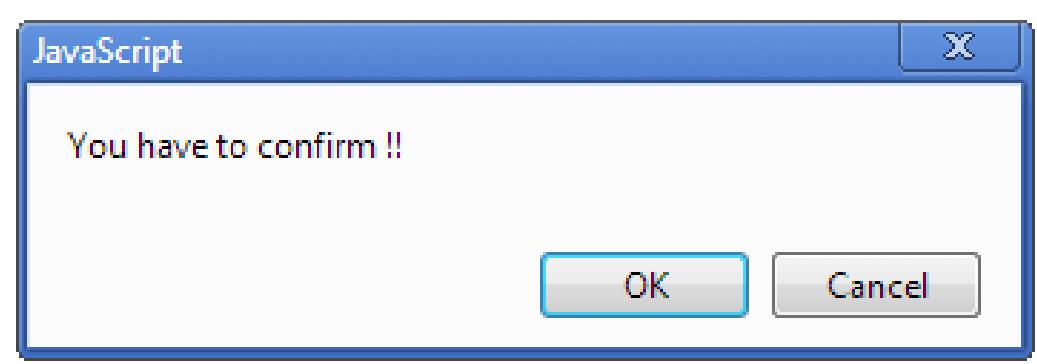
```
let nombre_array=[];
let dias_lectivos=["Lunes","Martes","Miercoles","Jueves","Viernes"];
alert(dias[0]);
alert(dias.length);
let variado=["Dia",45,32.78,true];
let elMartes=dias_lectivos[1];

//un string tambien es un array
let nombre="Pepe"
alert(nombre[2]);
```

- Ya hemos visto alert para mostrar información, también existen mas métodos para pedir y mostrar información.
- La instrucción prompt además de mostrar un mensaje permite recoger información del usuario



- La orden confirm es parecida a la anterior pero ofreciendo la respuesta si o no.



Entrada y salida de datos

```
let respuesta=confirm("Esta satisfecho con la aplicacion");
alert(respuesta);
let puntuacion=prompt("Puntue la aplicacion del 1 al 10");
alert("Su puntuacion ha sido de "+ puntuacion+" gracias por su visita");

let numero1,numero2,suma,resta;
numero1=prompt("Introduzca un numero");
numero2=prompt("Introduzca un numero");
suma=numero1+numero2;
alert(suma);
resta=numero1-numero2;
alert(resta);
```

Entrada de números con parseInt y parseFloat

```
//parseInt transforma un numero string a numero entero  
numero1=parseInt(prompt("Introduzca un numero sin decimales"));  
numero2=parseInt(prompt("Introduzca un numero sin decimales"));  
//parseFloat transforma un numero string a numero decimal  
numero1=parseFloat(prompt("Introduzca un numero con decimales"));  
numero2=parseFloat(prompt("Introduzca un numero con decimales"));
```

- **document** es una variable de tipo objeto que tiene un método llamado **write** que permite escribir código html en el **body** de la página.
- Por lo tanto cualquier etiqueta html que incluyamos se interpretará en el navegador. La idea es mezclar información de las variables con el código HTML.

```
document.write("Hola mundo");
document.write("<h1>Hola mundo</h1>");
let nota=7;
document.write("Tu nota es un <strong>" + nota + "</strong>");
```

Entrada y salida con JavaScript

```
document.write("Bienvenido a mi pagina web");
document.write("<h1>Bienvenido a mi pagina web</h1>");
var mensaje=prompt("Escriba una frase:");
document.write("<h1>" +mensaje+ "</h1>");
var nivel=prompt("¿Que nivel de encabezado desea?(1-6)");
document.write("<h" +nivel+ ">Bienvenido a mi pagina web</h" +nivel+ ">");
```

Crear HTML desde un array

Posición	0	1	2	3
Valor	León	Seat	5	V6

```
let coche=[];
coche[0]="León";
coche[1]="Seat";
coche[2]=5;
coche[3]="V6";

document.write("<ul>"+
              "<li>"+coche[0]+ "</li>"+
              "<li>"+coche[1]+ "</li>"+
              "<li>"+coche[2]+ "</li>"+
              "<li>"+coche[3]+ "</li>"+
              "</ul>");
```

EJERCICIO

Crear el siguiente array como el de la imagen de abajo y mostrarlo dentro de una tabla HTML de forma parecida:

0	1	2	3	4	5	6
Jorge	Pérez	35	1.77	80	Moreno	Soltero

Importancia del orden en JavaScript

```
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script type="text/javascript">
        alert("Hola");
        document.write("Hola");
    </script>
</head>
<body>
    <h1>Pagina de bienvenida</h1>
    <a href="http://www.google.es">Si quieres ir a google</a>
    <div><input type="button" value="Pulsar boton"></div>
</body>
<script type="text/javascript">
    document.write("Adios");
</script>
```

- Aunque no lo creamos hemos estado usando programación orientada a objetos en JavaScript sin darnos cuenta.
- `alert`, `prompt` y `confirm` son ordenes perteneciente al objeto `window` (se puede omitir).
- Por supuesto `document.write` que indica que el método `write` pertenece al objeto `document`.
- JavaScript es un lenguaje orientado a objetos (más o menos) y por tanto necesitamos familiarizarnos con sus objetos.

¿Qué es la POO?



¿Qué es la POO?

- Propiedades (Cómo es):
 - Tiene puertas
 - Tiene ruedas
 - Tiene ventanas
 - Tiene motor
 - Etc.



Objeto=Coche

- Métodos (¿Qué puede hacer?):
 - Arrancar
 - Acelerar
 - Frenar
 - Girar
 - Etc.



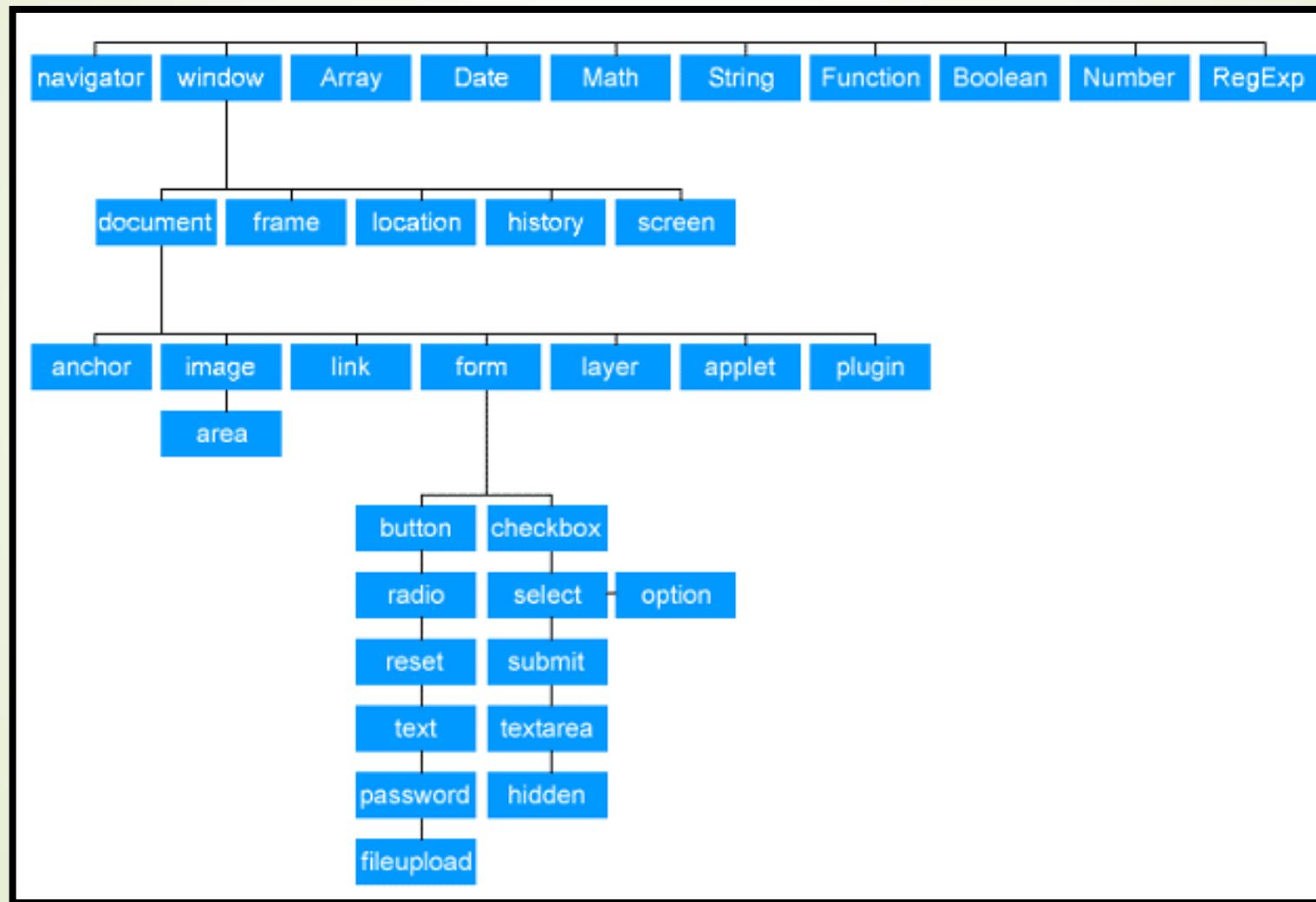
Instancia de coche.
Coche 1



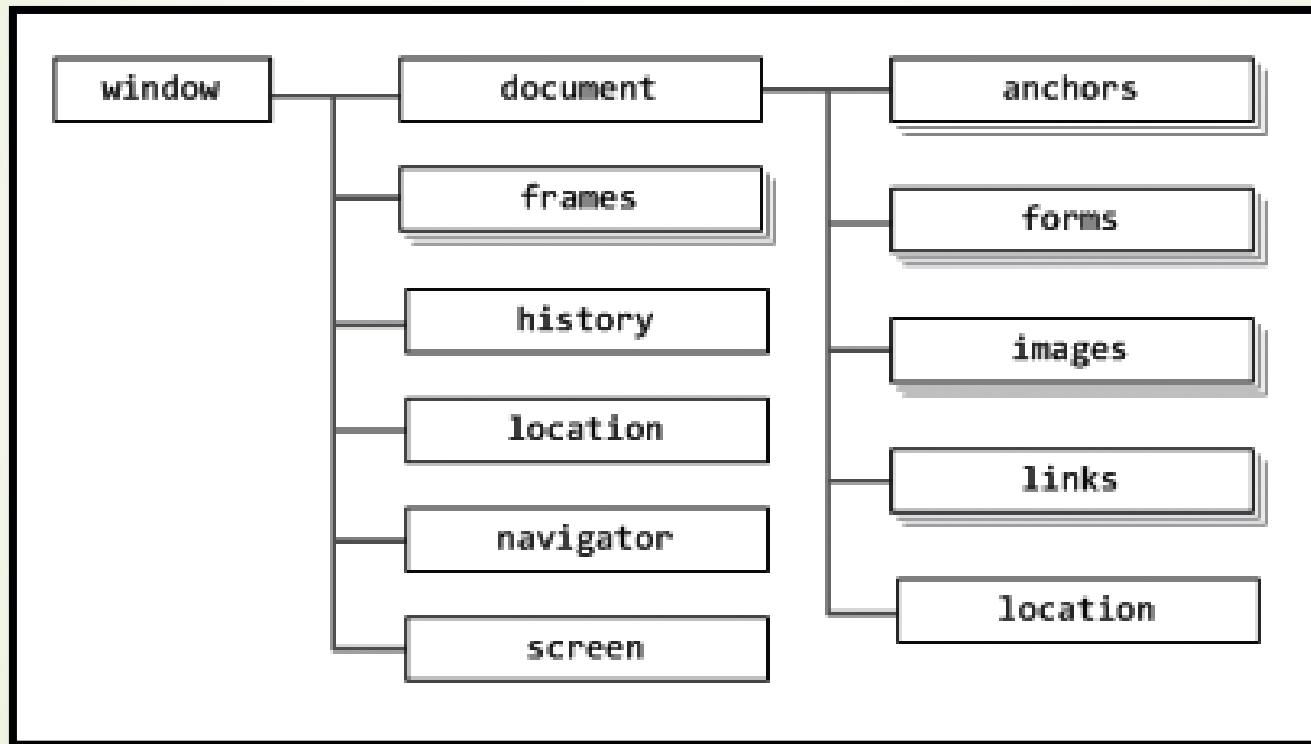
Instancia de coche.
Coche 2

- En la programación orientada a objetos los elementos de un programa van a funcionar conceptualmente igual.

Jerarquía de objetos JavaScript



window y document



Operador . (punto)

```
//Características
document.body
document.title
document.URL
document.links //array de enlaces
document.images //array de imágenes
document.cookie

//Capacidades
document.close();
document.open();
document.getElementByName("a");
document.write("<h1>Hola Mundo</h1>");

//Modificar características
document.title="Titulo cambiado con JavaScript";
document.body.style.backGroundColor="cyan";
document.images[0].style.width="500px";
```

Objeto string

```
let nombre="Daniel,Garcia";
//Caracteristicas
nombre.length;
//Capacidades
"Nombre: " + nombre
nombre.toUpperCase(); //Convierte a mayusculas
nombre.toLowerCase(); //Convierte a minusculas
nombre.indexOf("n"); //Posicion de La Letra n
nombre.split(","); //Separa string por ,
```

Objeto array

```
let colores=["Verde","Azul","Rojo"];
//Características
colores.length;
//Capacidades
colores.indexOf("Rojo");
colores.reverse();
colores.sort();
colores.push("Amarillo"); //Añade al final
colores.unshift("Amarillo"); //Añade al comienzo
colores.pop(); //Saca del final
colores.unshift() //Saca del comienzo
```

Objeto Math

```
//Caracteristicas
Math.PI
Math.E

//Capacidades
Math.sqrt(4); //Raiz cuadrada
Math.pow(2,4); //Potencia base exponente
Math.abs(-7); //Valor absoluto
Math.random(); //Numero aleatorio entre 0 y 1
Math.random() * 100 //Aleatorio entre 0 y 99
Math.random() * 100 + 1 //Aleatorio entre 1 y 100
```

Objetos predefinidos JavaScript

- Number.
- Date.
- window.
- history.
- navigator.
- screen.
- cookie.
- Timer y interval
- etc...

Más información en:

<https://www.arkaitzgarro.com/javascript/capitulo-14.html>

http://www.w3im.com/es/jsref/jsref_obj_array.html

Bibliografía

- ❑ Gauchat, Juan Diego: “El gran libro de HTML5, CSS3 y Javascript”.
Editorial Marcombo. 2012
- ❑ Vara, J.M. y otros: “Desarrollo Web en Entorno Cliente. CFGS”.
Editorial Ra-Ma. 2012
- ❑ “Programación en Javascript”.
Colección de artículos disponibles en la url.
<http://www.desarrolloweb.com/manuales/> Última visita: Septiembre 2017.
- ❑ W3SCHOOL “Manual de referencia y Tutoriales”
<http://www.w3schools.com/>
Última visita Septiembre 2017
- ❑ Comesaña, J.L. : Técnico en Desarrollo de Aplicaciones Web.
<http://www.sitiolibre.com/daw.php>
Última visita Septiembre 2017
- ❑ Píldoras Informáticas. Curso de JavaScript.
<https://www.pildorasinformaticas.es/course/javascript-desde-0/>
Ultima visita Septiembre 2018.