

# DESARROLLO WEB EN ENTORNO CLIENTE

## UD1 - SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE PROGRAMACIÓN

- CICLO FORMATIVO DE GRADO SUPERIOR EN DESARROLLO DE APLICACIONES WEB
- I.E.S. HERMENEGILDO LANZ – 2022/2023
- PROFESOR: **Isaías Fernández Lozano**
- [ifernandez@ieshlanz.es](mailto:ifernandez@ieshlanz.es)



# Resultados de Aprendizaje y Criterios de Evaluación

**RA1. Selecciona las arquitecturas y tecnologías de programación sobre clientes Web, identificando y analizando las capacidades y características de cada una**

CRITERIOS de evaluación del RA1	%UD	%Curso
a) Se han caracterizado y diferenciado los modelos de ejecución de código en el servidor y en el cliente Web.	15%	5%
b) Se han identificado las capacidades y mecanismos de ejecución de código de los navegadores Web.	15%	
c) Se han identificado y caracterizado los principales lenguajes relacionados con la programación de clientes Web.	15%	
d) Se han reconocido las particularidades de la programación de guiones y sus ventajas y desventajas sobre la programación tradicional.	15%	
e) Se han verificado los mecanismos de integración de los lenguajes de marcas con los lenguajes de programación de clientes Web.	20%	
f) Se han reconocido y evaluado las herramientas de programación sobre clientes Web.	20%	

# Objetivos Didácticos de la unidad 1



Diferenciar los lenguajes y la distribución de la lógica entre back-end y front-end



Conocer las características de los lenguajes de script y framework más usados

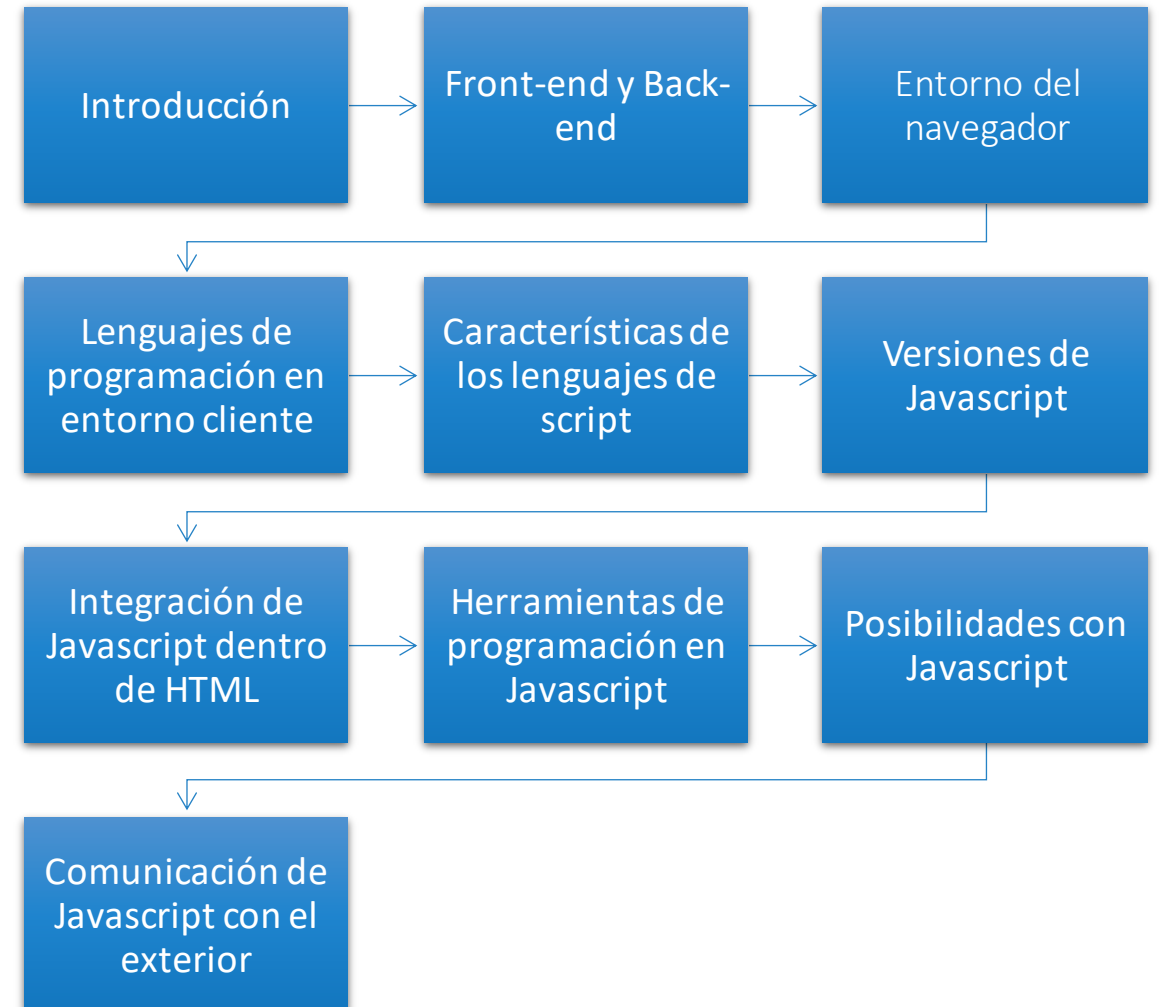


Aprender a integrar JavaScript en el código HTML



Entender las posibilidades de un lenguaje de programación en entorno cliente

# Índice





# Introducción

El W3C es un consorcio que desarrolla estándares para que Internet funcione a largo plazo. [www.w3c.es](http://www.w3c.es)

Profesión desarrollador web muy demandada → Auge de la computación en la nube vs computación *on-premises* (ejemplo AWS, Azure).

Desarrollador Web. Varias especializaciones:

Diseñador	Desarrollador en cliente	Desarrollador en servidor	Expertos en usabilidad	Arquitectos de datos...
-----------	--------------------------	---------------------------	------------------------	-------------------------

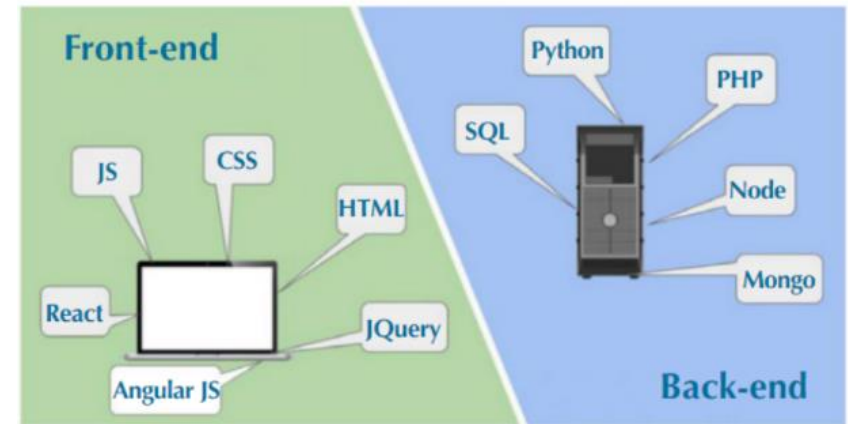
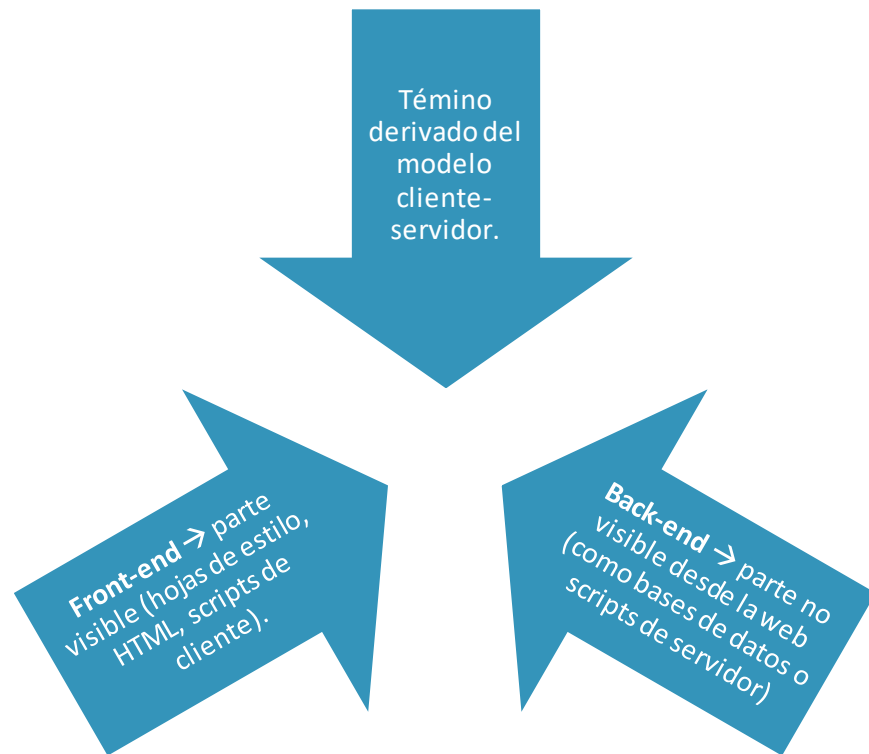
# Aplicaciones Web

1. Aplicaciones para ser ejecutadas en Navegadores (Browsers)
2. Tecnologías de una app web. Generalmente ambas:
  - En lado **cliente** → código accesible al usuario. El centro es HTML. Código interpretado por el navegador.
  - En lado **servidor** → código queda oculto. Código interpretado por el servidor web.
3. Arquitectura de 3 niveles:
  - Capa de **presentación** (o del cliente, o de navegador). Relacionada con el front-end. HTML, CSS, JavaScript
  - Capa **lógica** (o de servidor, o de aplicación). PHP, ASP, Python, Java, C, Swift (e incluso puede ser JavaScript gracias a tecnologías como *node.js*)
  - Capa de **datos** (o de negocio). SGBD, archivos multimedia, etc



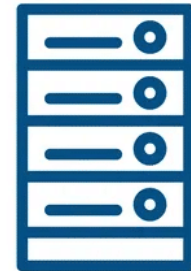
¿Qué es una  
Aplicación Web

# Front-end y back-end



## Front End

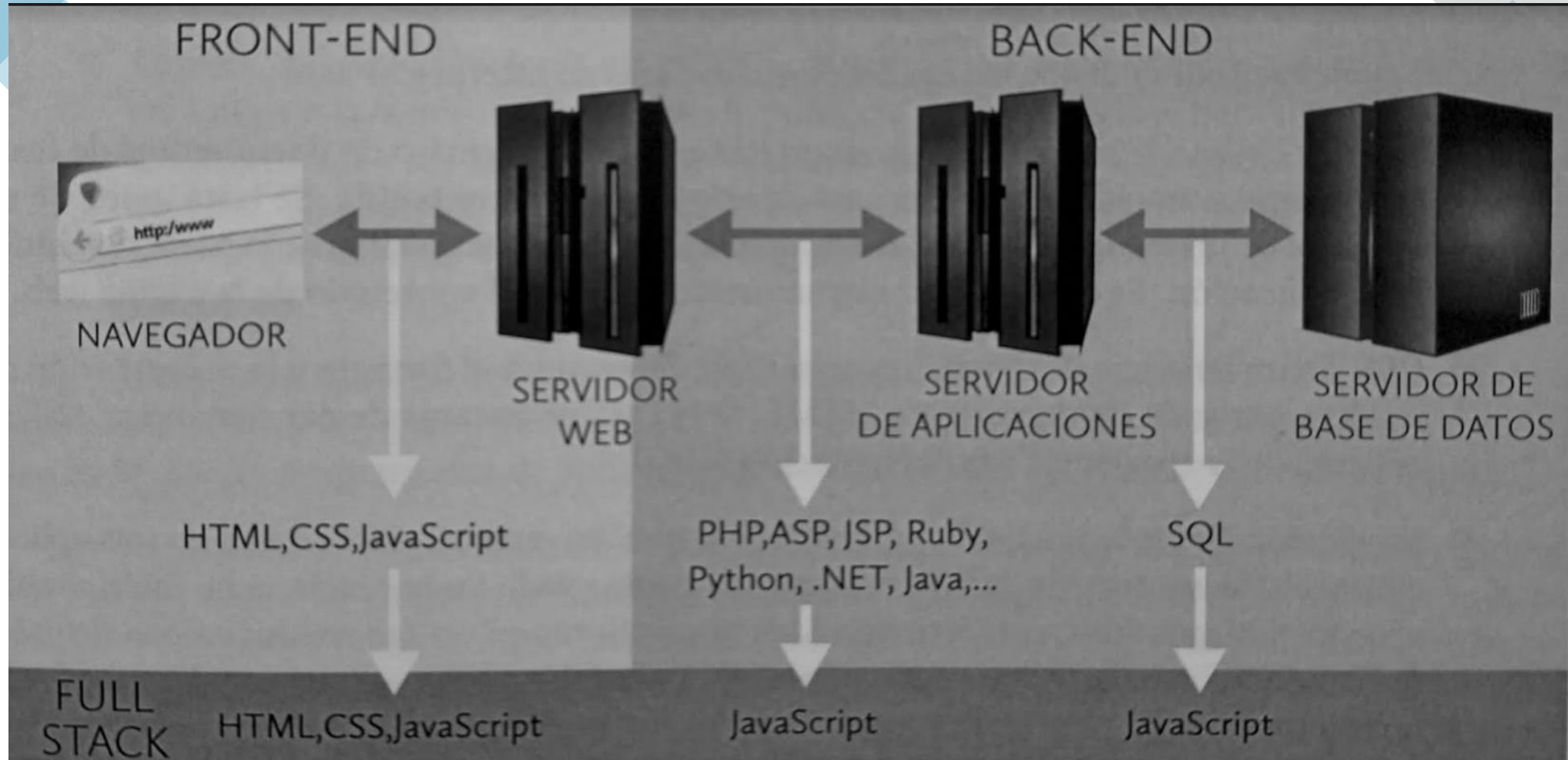
- Markup and web languages such as HTML, CSS and Javascript
- Asynchronous requests and Ajax
- Specialized web editing software
- Image editing
- Accessibility
- Cross-browser issues
- Search engine optimisation



## Back End

- Programming and scripting such as Python, Ruby and/or Perl
- Server architecture
- Database administration
- Scalability
- Security
- Data transformation
- Backup

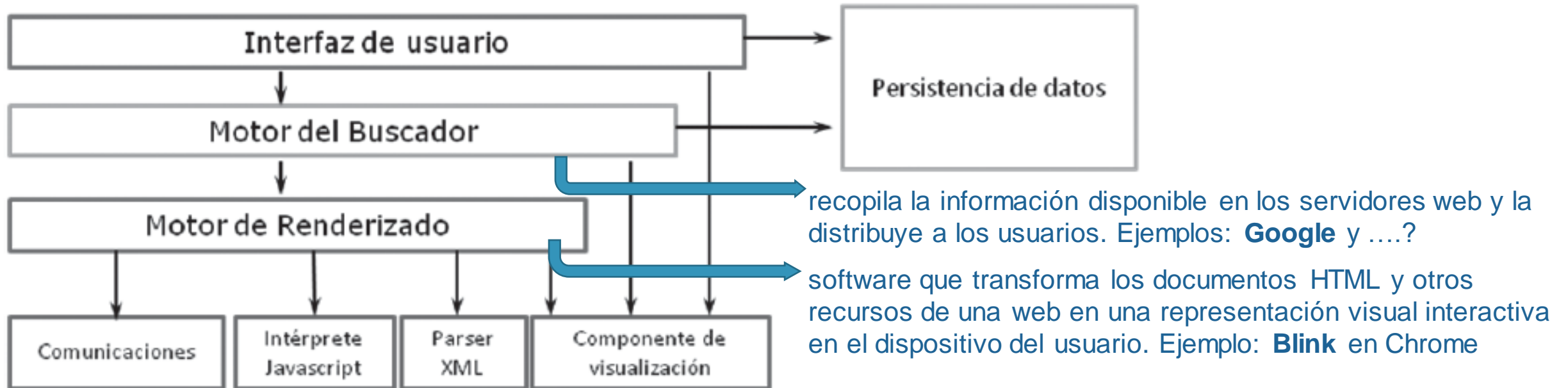
## Funcionamiento Front-end Back-end



[Full stack development](#) → uso de javascript en todas las capas. Mitiga el problema de los programadores de tener que especializarse en front-end o back-end



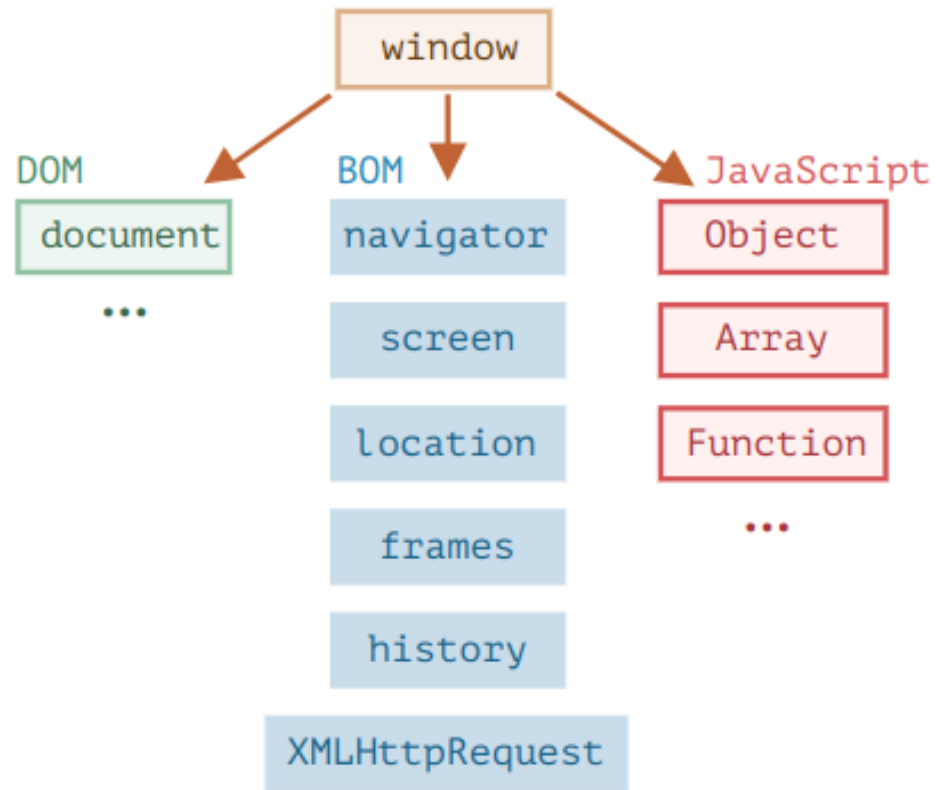
# Mecanismo de ejecución de código en Navegadores Web



**Figura 1.2.** *Arquitectura de referencia de un navegador web*

# Entorno del Navegador.

## Especificaciones:




Cada plataforma donde se ejecuta JavaScript aporta sus propios objetos y funcionalidades.

Los navegadores Web proporcionan un medio para controlar y gestionar las páginas Web.

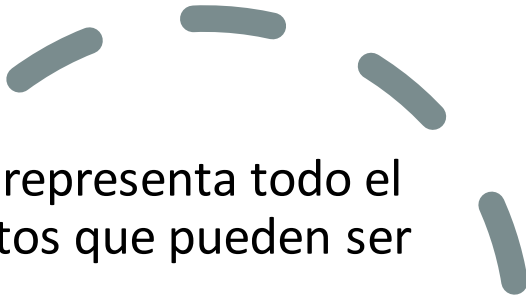
Lo que tenemos cuando JavaScript se ejecuta en un navegador web:

Hay un objeto “raíz” llamado window que tiene dos roles:


- Primero, es un objeto global para el código JavaScript, es decir proporciona variables y funciones que están disponibles en cualquier lugar..
- Segundo, representa la “ventana del navegador” y proporciona métodos para controlarla.



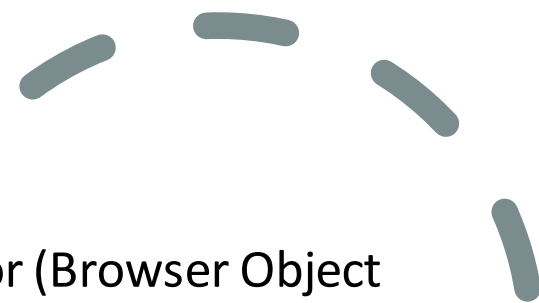
# DOM (Modelo de Objetos del Documento)

- 
- Document Object Model, o DOM, representa todo el contenido de la página como objetos que pueden ser modificados.
  - El objeto document es el punto de entrada a la página. Con él podemos cambiar o crear cualquier cosa en la página.

```
1 // cambiar el color de fondo a rojo
2 document.body.style.background = "red";
3
4 // deshacer el cambio después de 1 segundo
5 setTimeout(() => document.body.style.background = "", 1000);
```



## BOM (Modelo de Objetos del Navegador)



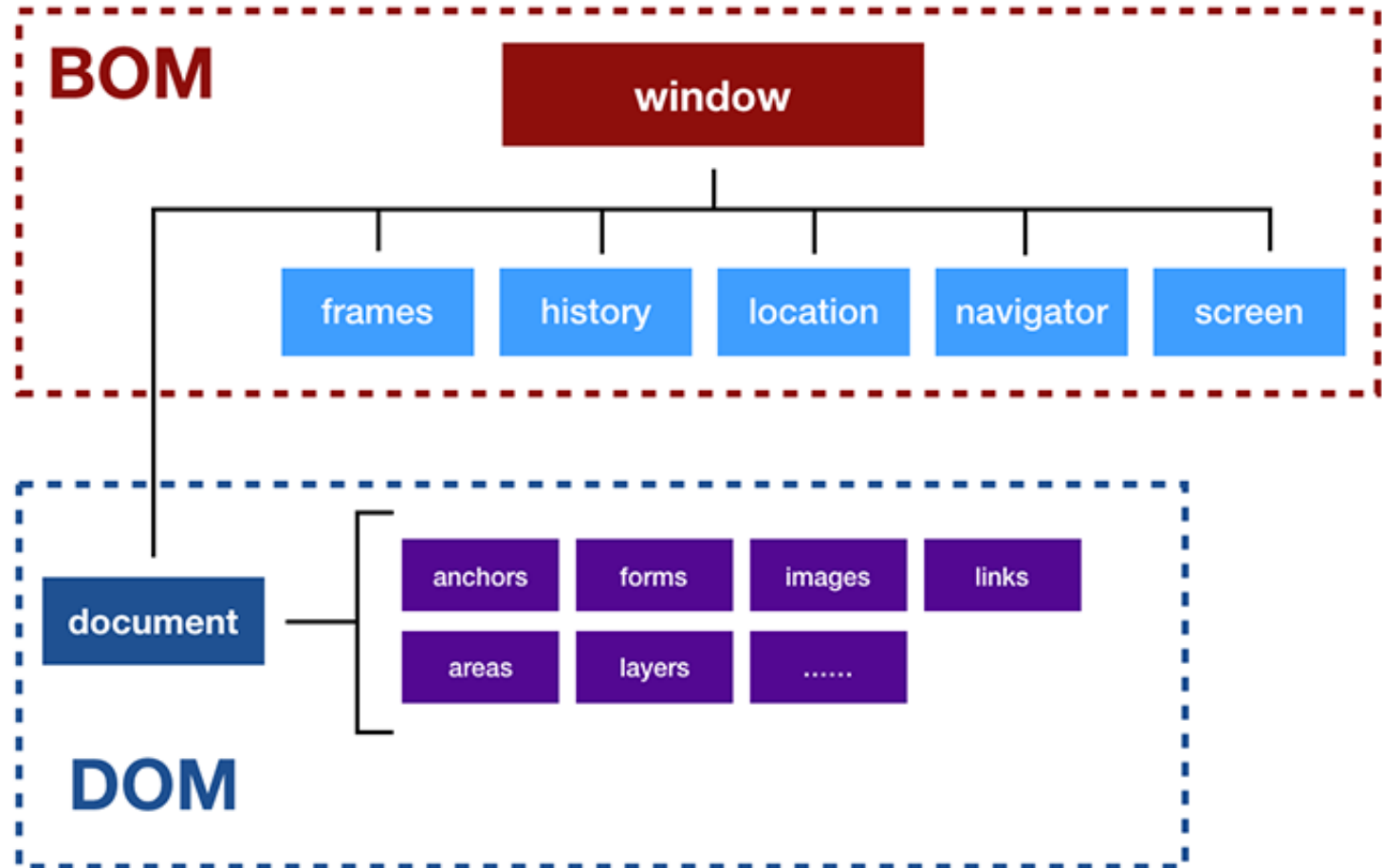
El Modelo de Objetos del Navegador (Browser Object Model, BOM) son objetos adicionales proporcionados por el navegador (entorno host) para trabajar con todo excepto el documento.

Por ejemplo:

- El objeto [navigator](#) proporciona información sobre el navegador y el sistema operativo. Hay muchas propiedades, pero las dos más conocidas son: `navigator.userAgent`: acerca del navegador actual, y `navigator.platform`: acerca de la plataforma (ayuda a distinguir Windows/Linux/Mac, etc.).
- El objeto [location](#) nos permite leer la URL actual y puede redirigir el navegador a una nueva.



# DOM vs BOM



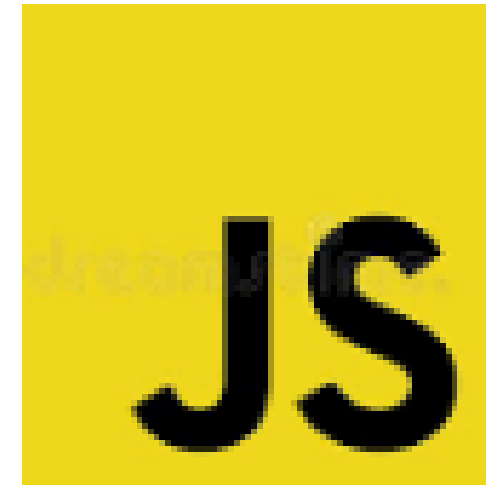
# Lenguajes de programación en entorno cliente

- **HTML**. Es en realidad, un lenguaje de marcas. Definirá el contenido del documento. El navegador leerá e interpretará este contenido, junto con las etiquetas (Parser XML) y lo mostrará al cliente (Componente de visualización). VENTAJA: el contenido debe verse igual en los diferentes navegadores. Versión actual: HTML5.
- **CSS**. Lenguaje de diseño gráfico que define la presentación del documento para hacer que la web sea atractiva al usuario. Se dedica al aspecto de la web. Actual: CSS3
- **Javascript**. Agrega contenido dinámico a las páginas. Es considerado un lenguaje de programación. Usado principalmente en front-end pero también en back-end.

Actualmente, las empresas no programan directamente sobre JavaScript, sino sobre un **framework** de este.

# JavaScript

- Lenguaje de programación **interpretado (ya que es un script)**. El propio navegador va **ejecutando sentencia a sentencia sin necesidad de compilar**.
- Se utiliza en el lado del **cliente**, es decir, se ejecuta en el navegador web (salvo excepciones), lo que permite mejoras en la interfaz de usuario.
- Es **orientado a objetos y basado en prototipos**, aunque algo diferente de la orientación a objetos tradicional. Podemos clonar objetos sin necesidad de instanciar.
- Es Imperativo y estructurado. Podemos ordenarle lo que queremos que haga, y es estructurado porque utiliza las tres estructuras básicas del lenguaje estructurado: Secuencia, Selección e interacción.
- Es de **tipado débil y dinámico**. Una misma variable puede tener valores de diferentes tipos a lo largo de su vida.
- Permite programar **comportamiento** de las páginas web.
- Podemos escribir código JavaScript en nuestro documento HTML o en documentos aparte y llamarlos desde la cabecera HTML.




ORACLE®



# Características

---

En el lado del  
cliente. 



Interpretado



Orientado a  
objetos y  
basado en  
prototipos



Imperativo y  
estructurado



Objetual, con  
tipificación  
débil y  
dinámica






# Motor o intérprete de JavaScript

- JavaScript puede ejecutarse no solo en los navegadores, sino también en servidores o incluso en cualquier dispositivo que cuente con un programa especial llamado [El motor o intérprete de JavaScript](#).
- El navegador tiene un motor embebido a veces llamado una “Máquina virtual de JavaScript”.
- Diferentes motores tienen diferentes “nombres en clave”. Por ejemplo:
  - [V8](#) – en Chrome, Opera y Edge.
  - [SpiderMonkey](#) – en Firefox.





## ¿Cómo trabajan los Motores de JavaScript?



Los motores son software complicado, pero los fundamentos son fáciles.

- El motor (embebido si es un navegador) lee (“analiza”) el script.
- Luego convierte (“compila”) el script a lenguaje de máquina.
- Por último, el código máquina se ejecuta, muy rápido.
- El motor aplica optimizaciones en cada paso del proceso. Incluso observa como el script compilado se ejecuta, analiza los datos que fluyen a través de él y aplica optimizaciones al código máquina basadas en ese conocimiento.

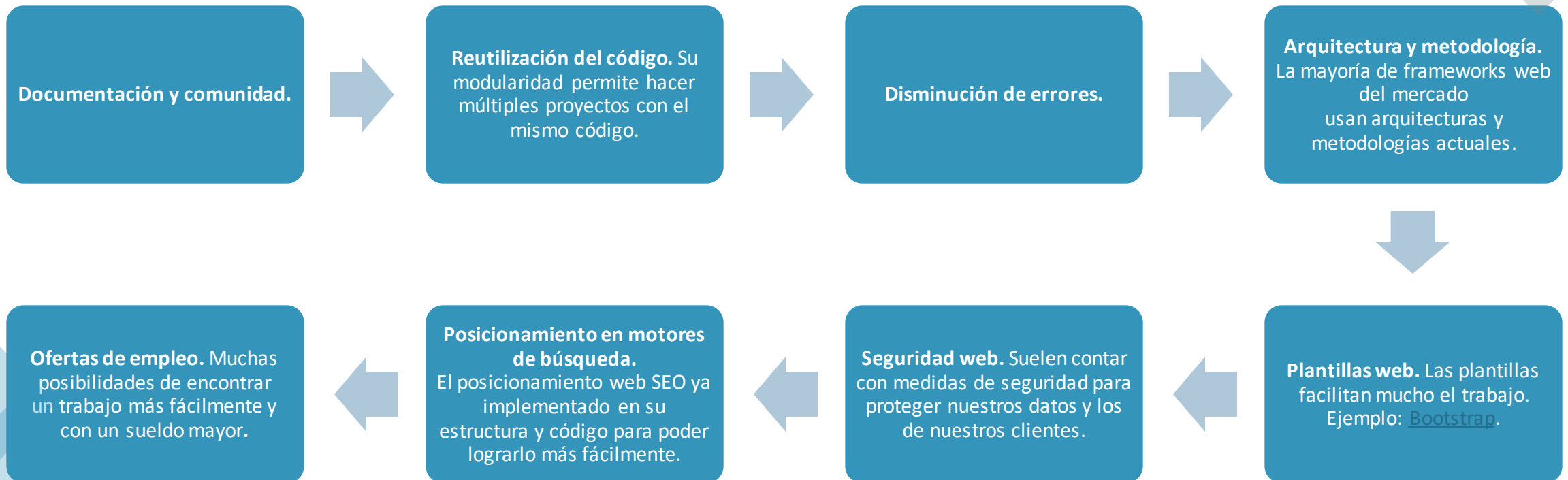
# Lenguajes por encima de JavaScript

- Las herramientas modernas hacen la conversión (Transpilación) muy rápida y transparente, permitiendo a los desarrolladores codificar en otros lenguajes y convertirlo automáticamente.
- Ejemplos de tales lenguajes:
- [CoffeeScript](#) Es una “sintaxis azucarada” para JavaScript. Introduce una sintaxis corta, permitiéndonos escribir un código más claro y preciso. Usualmente desarrolladores de Ruby prefieren este lenguaje.
- [TypeScript](#) se concentra en agregar “tipado estricto” (“strict data typing”) para simplificar el desarrollo y soporte de sistemas complejos. Es desarrollado por Microsoft.
- [Flow](#) también agrega la escritura de datos, pero de una manera diferente. Desarrollado por Facebook.
- [Brython](#) es un transpilador de Python a JavaScript que permite escribir aplicaciones en Python puro sin JavaScript.
- [Kotlin](#) es un lenguaje moderno, seguro y conciso que puede apuntar al navegador o a Node.

# Framework web

Conjunto de **herramientas, estilos y librerías** dispuestas a través de una estructura o esqueleto base, para el desarrollo de aplicaciones web más escalables y sencillas de mantener.

## Ventajas de los frameworks web





# Frameworks más usados de Javascript



Angular



React



Vue

ReactJS (<https://reactjs.org>). Creado por Facebook.

- Librería Javascript
- Orientado a componentes que gestionan su propio estado y que al agruparse permiten crear interfaces.
- Utiliza un DOM virtual que se mapean al DOM del navegador

Angular (<https://angular.io>). Creado por Google.

- Ha dejado de ser librería y es una plataforma de desarrollo.
- Curva de aprendizaje pronunciada.
- Actualmente se programa en TypeScript (superconjunto de JavaScript creado por Microsoft) y que usa el patrón *reactivo* RxJS.

Vue.js (<https://vuejs.org>).

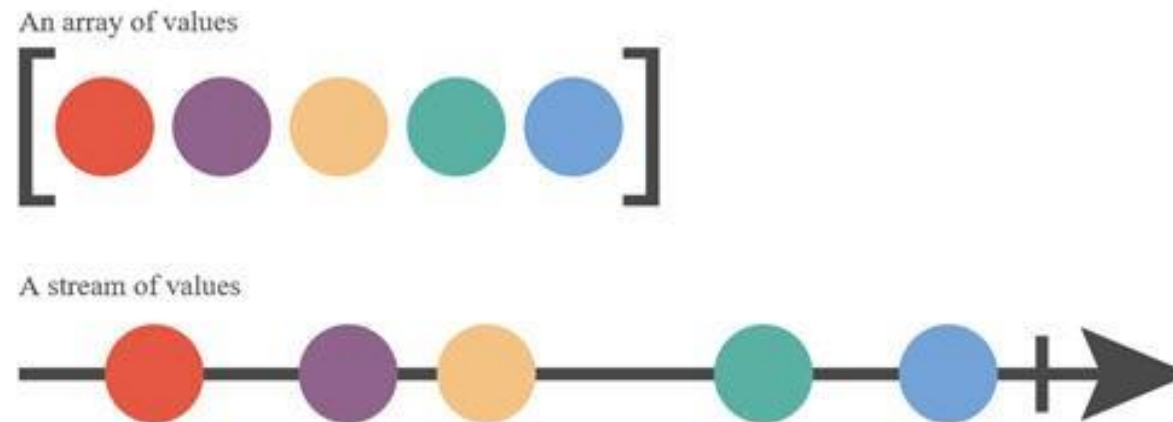
- Trata de aprovechar las ventajas de los dos anteriores.
- Mejora curva de aprendizaje
- Utiliza la técnica de DOM virtual por sus ventajas

# Programación reactiva.

## Idea general:

---

- **Programación clásica:** yo pido los datos al servidor, a un sensor, a la base de datos, etc
- **Programación reactiva:** el software se actualiza solo cuando las fuentes de datos de las que dependen se modifican. *“cada vez que haya un cambio infórmame”...*



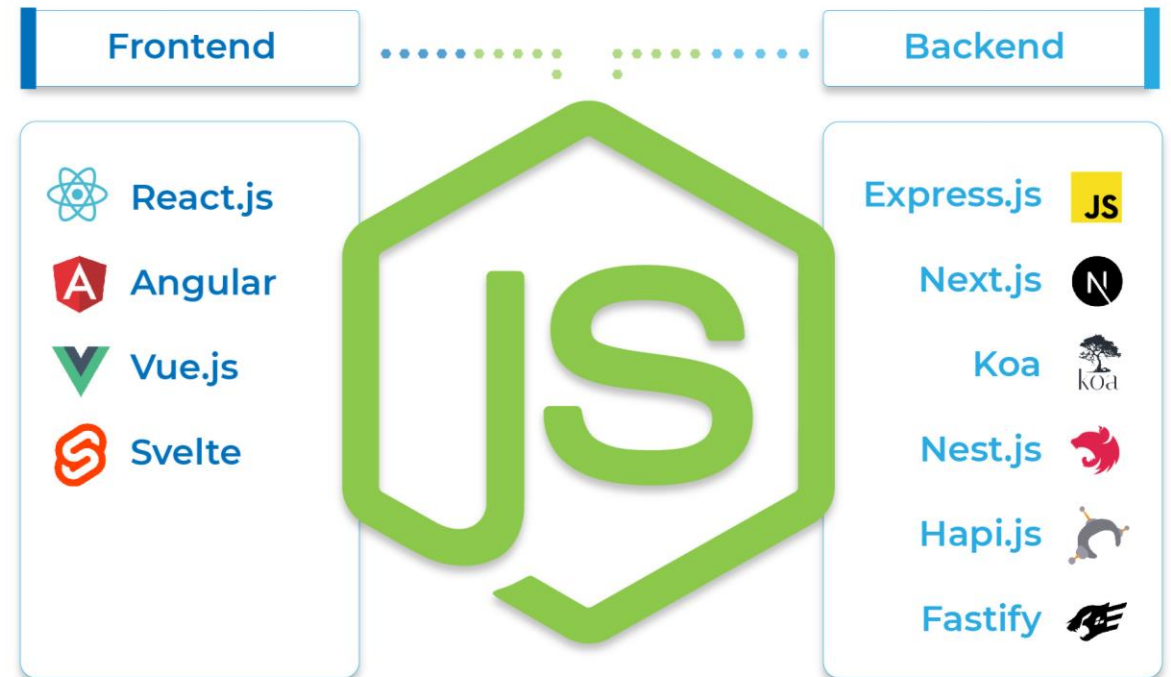
# Otros frameworks

## Node.js

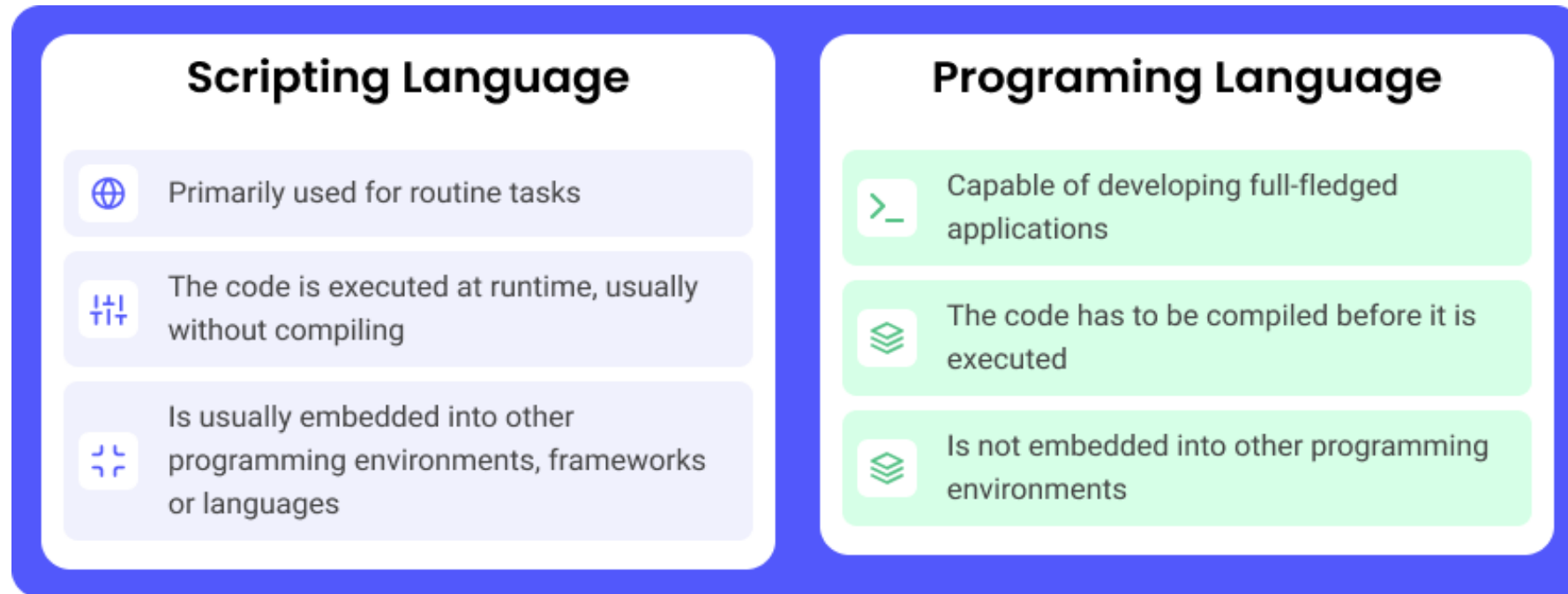
es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la *capa del servidor*, basado en JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google



### — Node.js Frameworks —



# Lenguajes de Script vs Lenguajes de Programación



Algunos lenguajes de programación son: Java, C, C++, Swift, Pascal, etc.

Algunos lenguajes de scripting son: JavaScript, Shell, Perl, PHP, Python, Ruby, etc.

## ACTIVIDAD

¿CUALES CONSIDERAS QUE SON LAS VENTAJAS Y DESVENTAJAS DE LOS LENGUAJES DE SCRIPTING VS LOS LENGUAJES DE PROGRAMACION TRADICIONAL?



# Versiones de Javascript

JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997.

ECMAScript is the official name of the language.

ECMAScript versions have been abbreviated to ES1, ES2, ES3, ES5, and ES6.

Since 2016 new versions are named by year (ECMAScript 2016 / 2017 / 2018).

## ECMA Script versions

ES1	Jun 1997		ES6	ES2015
ES2	Jun 1998		ES7	ES2016
ES3	Dec 1999		ES8	ES2017
ES5	Dec 2009		ES9	ES2018
ES5.1	Jun 2011		ES10	ES2019
			ES11	ES2020
			ES.Next	

# Integración de Javascript dentro de HTML

## 2 Opciones:

1. Integrar el código JavaScript dentro de los archivos HTML.
2. Tener separado del HTML el código JavaScript en archivos con extensión js.



**Figura 1.2**

Código JavaScript dentro y fuera de un archivo HTML.



Lo más limpio y eficaz es tener el código JavaScript fuera de los archivos HTML por las ventajas que ofrece.

## Opción 1 integrar código en el HTML

Detalle del archivo *index.html*:

```
<!DOCTYPE html>
<html>
<title>Myfpschool</title>
<head>
<script>
  function diAlgo()
  {
    alert("Hola");
  }
</script>
</head>
<body>

<script>
diAlgo();
</script>

</body>
</html>
```

## Opción 2 – incluir llamada al archivo .js

Contenido del archivo *index.html*:

```
<!DOCTYPE html>
<html>
<title>Myfpschool</title>
<head>
<script src="script.js"></script>
</head>
<body>
</body>
</html>
```

A continuación, se muestra el contenido del archivo *script.js*:

```
function diAlgo()
{
    alert("hola");
}
diAlgo();
```

# Herramientas de programación Javascript

- **Editores de texto** sencillos como Notepad.
- Herramientas **online**. Ejemplos.
  - ✓ Coding Ground de Tutorialspoint  
[https://www.tutorialspoint.com/online\\_javascript\\_editor.php](https://www.tutorialspoint.com/online_javascript_editor.php)
  - ✓ Codepen io. <https://codepen.io/> → vista con HTML y CSS

- **IDE's y Sistemas de Control de Versiones**. Ejemplos:



- ✓ Visual Studio Code de Microsoft. <https://code.visualstudio.com/>



- ✓ Atom: <https://atom.io/>



- ✓ Sublime Text: <https://www.sublimetext.com/>

Los 3 cuentan con integración en Github para el control de versiones



# Posibilidades con Javascript

Modificación del  
contenido de  
una página web

Cambio de  
atributos de  
objetos HTML

Cambio del estilo  
CSS



# Modificación del contenido de una página web

```
<!DOCTYPE html>
<html>
<body>

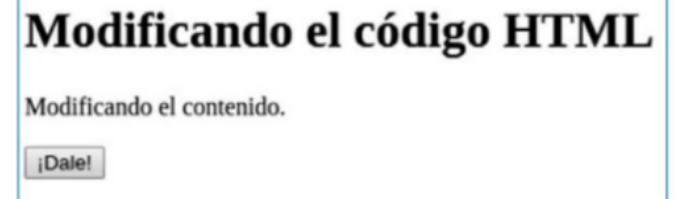
<h1>Modificando el código HTML</h1>

<p id="prueba">Modificando el contenido.</p>

<button type="button"
onclick="document.getElementById('prueba').innerHTML = 'CAMBIANDO el
contenido!'">
¡Dale!</button>

</body>
</html>
```

¿Qué realiza el código javascript?



# Cambiar atributos de Objetos HTML

```
<!DOCTYPE html>
<html>
<body>

<h1>Cambio de imágenes con JavaScript</h1>



<p>Haz click sobre las letras para cambiarlas.</p>

<script>
function cambiaPic() {
    var image = document.getElementById('myFPImage');
    if (image.src.match("green")) {
        image.src = "http://myfpschool.com/wp-content/uploads/2016/06/my-black.jpeg";
    } else {
        image.src = "http://myfpschool.com/wp-content/uploads/2016/06/my-green.jpeg";
    }
}
</script>

</body>
</html>
```

¿Qué realiza el código javascript?



AHORA Cambia el código para que use tus propias imágenes

# Cambiar el estilo CSS

Aprende JavaScript con MyFPSchool!

¡Dale!

**Figura 1.7**

Aspecto de la página antes de pulsar el botón.

Aprende JavaScript con MyFPSchool!

¡Dale!

**Figura 1.8**

Aspecto de la página después de pulsar el botón.

¿Qué código javascript puedo usar?

```
<!DOCTYPE html>
<html>
<body>
<p id="mytxt">Aprende JavaScript con MyFPSchool!</p>
<button type="button" onclick="myFunction()">¡Dale!</button>
<script>
function myFunction() {
    var x = document.getElementById("mytxt");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>
</body>
</html>
```

# Comunicación de Javascript con el exterior

1. Escribir en la consola del navegador
2. Escribir en un elemento con innerHTML
3. Generar código HTML con document.write
4. Mensajes de alerta con window.alert

```
<script>
    console.log("Síntesis, ejemplo de consola");
</script>
```

```
<p id="parrafito"></p>
<script>
document.getElementById("parrafito").innerHTML = 5 + 6;
</script>
```

```
<script>
    document.write("<h2>Buenos días</h2>");
</script>
```

```
<script>
    window.alert("BUENAS NOCHES");
</script>
```



# Ejercicios Propuestos Libro Síntesis

## Diapositiva 26

1. Crea un segundo botón para el ejemplo del apartado 1.6.1 que modifique el título “Modificando el contenido”.
2. Elabora una secuencia de imágenes con al menos 4 fotogramas. Según el usuario vaya haciendo clic sobre las imágenes, la secuencia se irá reproduciendo.
3. Realiza un test de 7 preguntas donde cada pregunta tenga dos botones (verdadero y falso). Dependiendo de si el usuario hace clic en verdadero o en falso la frase se pone verde (acierto) o roja (error).
4. Desarrolla una página web en la que al pulsar un botón se genere un mensaje por consola.
5. Diseña una web con tres botones para que al pulsarlos se genere un mensaje de bienvenida en ruso, español e inglés en la misma página. El mensaje irá en un párrafo (etiqueta <p>). Se pide que los mensajes tengan un color diferente dependiendo del idioma.
6. Modifica el código anterior para que los mensajes de bienvenida se generen con alertas.
7. Cambia el código anterior para que los mensajes de bienvenida se generen en consola.
8. Transforma el código anterior para que los mensajes de bienvenida se generen con el método write().