



CICLO 4a

[FORMACIÓN POR CICLOS]

Desarrollo de **APLICACIONES WEB**

JavaScript Introducción



**UNIVERSIDAD
DE ANTIOQUIA**

Facultad de Ingeniería



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA](#)

JavaScript

Es un lenguaje de programación liviano usado para construir páginas web interactivas, insertar texto dinámico en HTML y obtener información sobre el usuario de un PC, entre otras bondades.

Nota importante: no está relacionado con Java a pesar de algunas similitudes de tipo sintáctico y de nombre.





[Esta foto](#) de Autor desconocido está bajo licencia CC BY

JavaScript

Existen al menos *tres* cosas geniales sobre JavaScript:

- Su completa integración con HTML y CSS
- Las cosas simples se hacen de manera simple
- Está soportado por la mayoría de los navegadores y habilitado de forma predeterminada



Beneficios de JavaScript

Usabilidad

Se puede modificar una página sin necesidad de publicar de nuevo en el servidor

Eficiencia

Es posible realizar cambios rápidos y pequeños sin esperar por el servidor

Manejador de eventos

Responde a acciones de usuario como clics y teclado

Seguridad

Tiene acceso a datos privados del servidor

El cliente no puede ver el código

Compatibilidad

No está sujeto a compatibilidad de navegadores



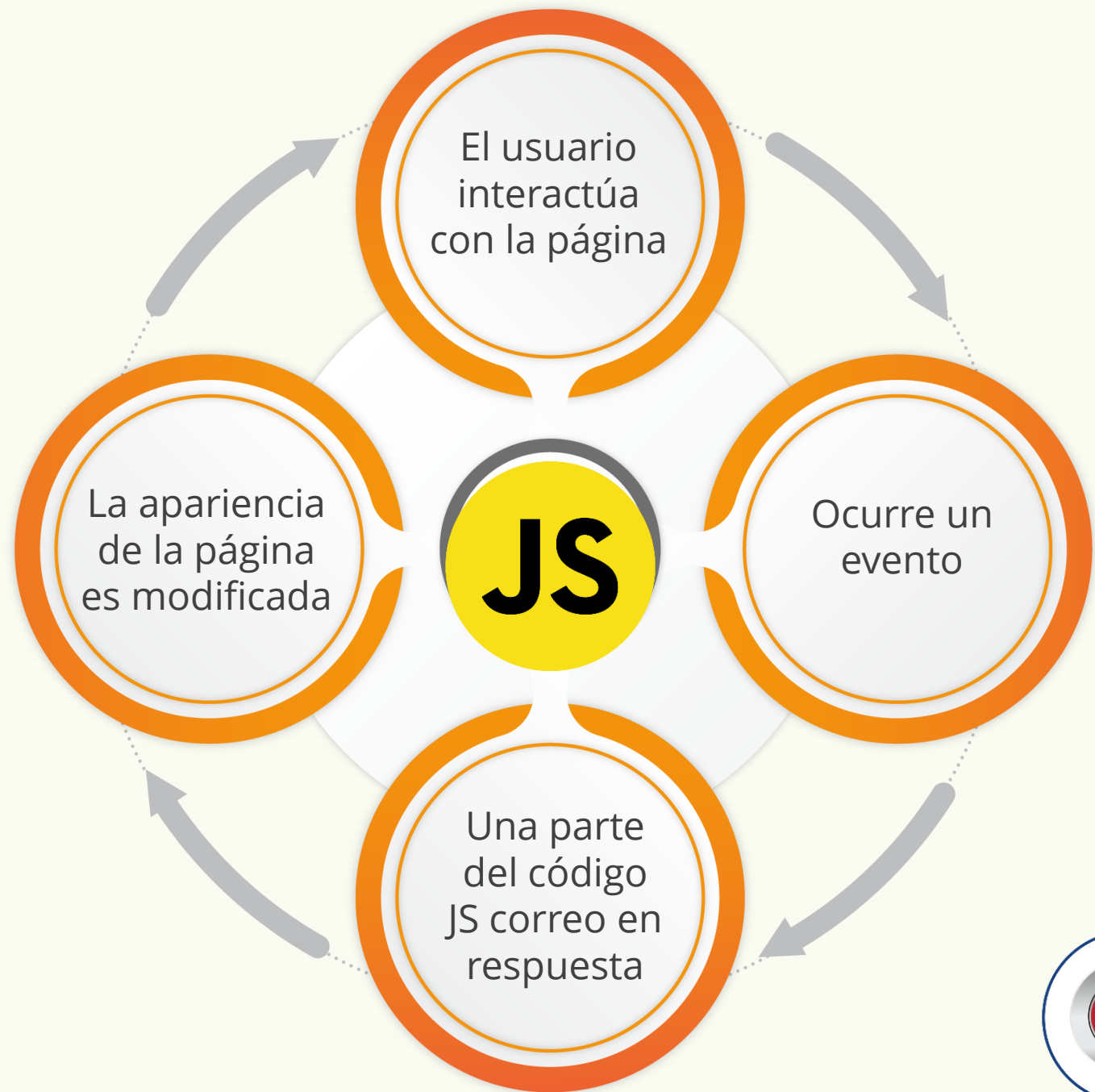
Vinculando un archivo JavaScript

```
<script src="filename" type="text/javascript"> </script>
```

- La etiqueta debe ser ubicada en el encabezado de la página HTML
- El código debe ser almacenado por separado en un archivo .js
- El código JS puede ser ubicado directamente en el cuerpo o cabecera del archivo HTML (como CSS)
- Sin embargo, no es una buena práctica (debería separarse el contenido, la presentación y el comportamiento)



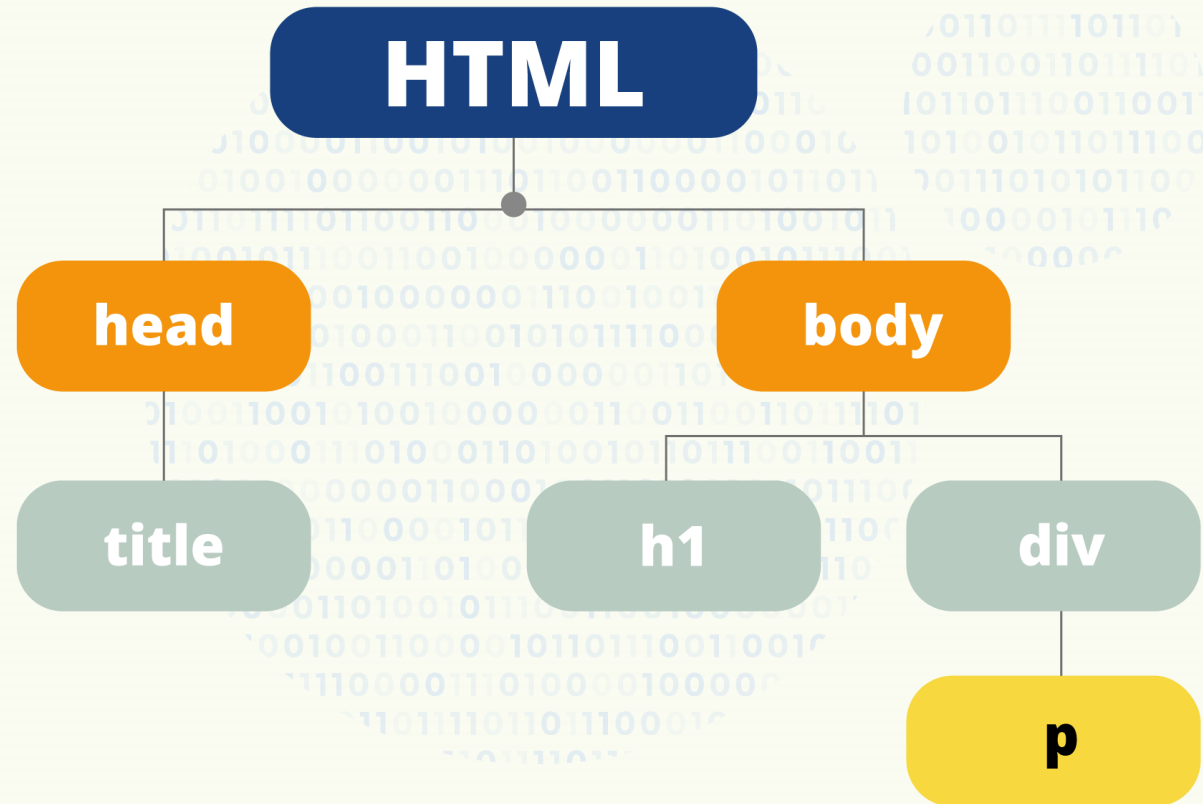
Event-Driven



DOM

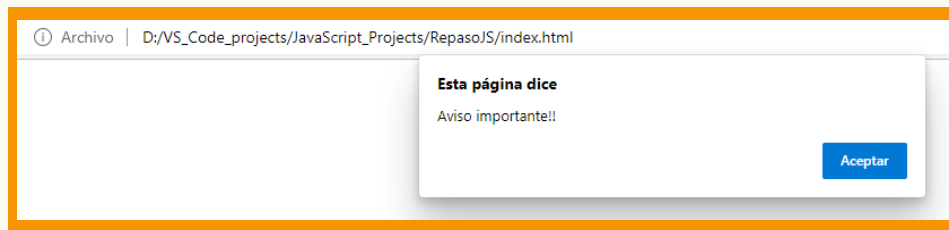
La mayoría de código JS manipula páginas HTML

- Podemos examinar el estado de los elementos
- Cambiar el estado
- Cambiar estilos



Instrucción Alert

Esta instrucción puede incluirse en el código HTML, e incluyéndola dentro de la etiqueta `<script>` es posible hacer que aparezca una ventana emergente



```
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="utf-8" />
  <title>
    Repaso JavaScript
  </title>
  <script>
    alert('Aviso importante!!');
  </script>

<body>
  <h1>
    Aprendiendo JavaScript rápido
  </h1>
  <p>Hola soy Judy Moreno WEB</p>

</body>
</head>
</html>
```



Instrucción Alert

La mejor práctica es tener un archivo exclusivo JS

```
//Archivo main.js
<script>
    alert('Aviso importante!!');
</script>
```

```
//Archivo index
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="utf-8" />
    <title>
        Repaso JavaScript
    </title>
    <script src="main.js" type="javascript"></script>

<body>
    <h1>
        Aprendiendo JavaScript rápido
    </h1>
    <p>Hola soy Judy Moreno WEB</p>

</body>
</head>
</html>
```



Variables y arreglos

- Las variables en JS se declaran usando la palabra **var**; no se especifica el tipo
- Los arreglos se declaran como variables, pero se le agrega []

```
//Archivo main.js  
var mostrar = "Tabla del 8 <br/>";  
var mult=0;  
var array = []; //Arreglo vacio  
var array2 = ["Pato", "Perro", "Gato"] //Arreglo prellenado  
array[0] = "Luna" //Almacenamiento  
array[1] = "Sol" //Almacenamiento
```



Sentencia if/else

- La estructura if/else es igual a Java
- Los operadores lógicos son iguales a Java

```
//Archivo main.js

var nombre = "Marcela Moreno"; //Creación de una variable
var edad = 20;
if(edad<18)
{
    var mensaje = nombre + " es menor de edad";
}else{
    mensaje = nombre + " es mayor de edad";
}
var datos = document.getElementById("datos");
datos.innerHTML= mensaje;
```



Ciclos

- La estructura de los ciclos **for**, **while** y **do while** es igual a Java

```
//Archivo main.js
var mostrar = "Tabla del 8 <br/>";
var mult=0;
for(var i=0; i<10;i++){
    mult=8*i;
    mostrar = mostrar + 8 +" x "+i+" = "+mult+ "<br/>" ;
}
var datos = document.getElementById("datos");
datos.innerHTML=mostrar;
```

Aprendiendo JavaScript rápido

Hola soy Judy Moreno WEB

Tabla del 8

8 x 0 = 0
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72

