

Curso de Frontend Developer

¿Qué son los motores de render?

Los motores de renderizado son programas que traducen nuestro código en un lenguaje que entienda el navegador, de esta manera el programa sabrá que es lo que tiene que mostrar por pantalla al usuario.

¿Cuáles son los motores del navegador?

Los navegadores tienen sus propios motores: Chrome - Blink, Edge - Edge HTML, Safari - Webkit y Firefox - Gecko. Todos realizan esta compilación de manera diferente, pero con el mismo resultado, es decir, convierten los archivos a píxeles.

Proceso de renderizado del motor del navegador

1. Transforma los archivos a un árbol de objetos HTML o CSS, estos se denominan DOM (Document Object Model) y CSSDOM (Cascade Style Sheet Object Model), respectivamente. Cada nodo en el árbol es una representación de los elementos que contiene el archivo HTML o CSS.
 2. Calcula el estilo correspondiente a cada nodo del DOM relacionado al CSSDOM.
 3. Calcula las dimensiones de cada nodo y dónde va en la pantalla.
 4. Pinta o renderiza los diferentes elementos como cajas o contenedores.
 5. Agrupa todas las cajas en diferentes capas para convertirlas en una imagen que se renderiza en pantalla.
-

¿Qué define la Etiqueta html?

La etiqueta `<html>` define el elemento raíz de un documento HTML. Todos los demás elementos deben estar contenidos dentro de este elemento raíz. En esta etiqueta se especifica el lenguaje de la página web mediante la propiedad `lang`.

¿Qué define la etiqueta head?

La etiqueta `<head>` define la metainformación, es decir, toda información que no es contenido como tal de la página web. Por ejemplo, los enlaces a archivos CSS y JavaScript, el título y la imagen que aparecen en la pestaña del navegador. Esto es importante para motores de búsqueda como Google.

¿Qué es la etiqueta body?

La etiqueta `<body>` define el contenido de la página web. Debe ser hijo cercano de `<html>` y padre de todas las etiquetas HTML, excepto por aquellas que definan metainformación.

¿Qué es el HTML semántico?

El HTML semántico consiste en que cada elemento tenga su propia etiqueta que lo defina correctamente. Sin utilizar etiquetas muy generales, como `<div>` o ``.

¿Cuál es el problema con la etiqueta `div`?

La etiqueta `div` define un bloque genérico de contenido, que no tiene ningún valor semántico. Se utiliza para elementos de diseño como contenedores.

¿Cuáles son las etiquetas semánticas?

- `<header>`: define el encabezado de la página (no confundir con `<head>`).
- `<nav>`: define una barra de navegación que incluye enlaces.
- `<section>`: define una sección de la página.
- `<footer>`: define un pie de página o de sección.
- `<article>`: define un artículo, el cual puede tener su propio encabezado, navegación, sección o pie de página.

¿Qué es un selector en CSS?

El selector define el elemento o conjunto de elementos HTML a los cuales se añadirán estilos.

SELECTORES BÁSICOS

| | |
|-------------|--------------------------------------|
| DE TIPO | <code>div { ... }</code> |
| DE CLASE | <code>. elemento { ... }</code> |
| DE ID | <code>#id-del-elemento</code> |
| DE ATRIBUTO | <code>a[href = "..."] { ... }</code> |
| UNIVERSAL | <code>* { ... }</code> |

¿Cuáles son los selectores combinadores?

Un selector combinador es la unión de dos o más selectores básicos.

SELECTORES COMBINADORES

| | |
|---------------------|-------------------------|
| DESCENDIENTES | <code>div p</code> |
| HIGO DIRECTO | <code>div > p</code> |
| ELEMENTO ADYACENTE | <code>div + p</code> |
| GENERAL DE HERMANOS | <code>div ~ p</code> |

- **Combinador de descendientes** Selecciona todos los elementos del selector de la derecha que son hijos del selector de la izquierda, independientemente de la profundidad. Estos selectores están separados por un espacio.

```
padre hijos {  
    /* Todos los hijos del padre */  
}  
  
div p{  
    /* Todos los hijos <p> de <div>*/  
}  
  
.container img{  
    /* Todos los hijos <img> de la clase "container"*/  
}
```

- **Combinador de hijo directo** Selecciona todos los elementos del selector de la derecha que son hijos directos del selector de la izquierda. Estos selectores están separados por >.

```
padre > hijos_directos {  
    /* Todos los hijos directos del padre */  
}  
  
div > p{  
    /* Todos los hijos directos <p> de <div>*/  
}  
  
.container > img{  
    /* Todos los hijos directos <img> de la clase "container"*/  
}
```

- **Combinador de elemento adyacente** Selecciona todos los elementos del selector de la derecha que están adyacente al selector de la izquierda. Estos selectores están separados por +. **Adyacente significa que comparten el mismo padre y está situado inmediatamente hacia abajo de otro elemento.** Básicamente selecciona a la etiqueta que esté debajo de la primera etiqueta. Ver imagen de abajo.

```

elemento + adyacente {
    /* Elementos adyacentes */
}

div + p{
    /* Todos los <p> adyacentes a <div>*/
}

.container + img{
    /* Todos los <img> adyacentes a la clase "container"*/
}

```

- **Combinador general de hermanos** Selecciona todos los elementos del selector de la derecha que son hermanos del selector de la izquierda. Estos selectores están separados por ~. **Hermanos significa que comparten el mismo padre y están situados hacia abajo de otro elemento.**

```

elemento ~ hermanos {
    /* Elementos hermanos */
}

div ~ p{
    /* Todos los <p> hermanos de <div>*/
}

.container ~ img{
    /* Todos los <img> hermanos de la clase "container"*/
}

```

EJEMPLO PRÁCTICO DE UN **Combinador general de hermanos** EN LA SIGUIENTE PAGINA

```
<div class="container">
  <p>Soy otro párrafo</p>
  <div>Soy un div</div>
  <p>Soy un párrafo hermano de div</p>
  <p>Soy un párrafo hermano de div</p>
  <p>Soy un párrafo hermano de div</p>
</div>
<hr/>
```



```
/*Agrega los selectores aquí */
div ~ p {
  color: blue;
  background-color: pink;
}
```

Soy otro párrafo

Soy un div

Soy un párrafo hermano de div

Soy un párrafo hermano de div

Soy un párrafo hermano de div