Engineer

HTML

Lenguaje en el que se escribe la estructura de una página

Qué es

HTML, abreviatura de HyperText Markup Language y todavía conocido ampliamente como HTML5, es el formato de texto plano en el que llega cada página web: contenido envuelto en etiquetas que dicen qué es cada pieza, para que el navegador distinga un encabezado de un párrafo, un enlace de un elemento de lista y un campo de formulario de su etiqueta.

El navegador lee ese texto de arriba abajo y construye un árbol de elementos, el modelo de objetos del documento o Document Object Model. Las hojas de estilo aplican estilos a ese árbol, los scripts lo modifican y los lectores de pantalla lo anuncian. La división del trabajo en la web es antigua y sigue vigente: HTML dice qué es algo, CSS cómo se ve y JavaScript qué hace cuando alguien interactúa con ello. Un elemento como `<nav>`, `<button>` o `<h2>` lleva un significado que todos esos lectores pueden usar. Un `<button>` real recibe foco de teclado, un rol anunciado por la tecnología de apoyo y activación con Intro o Espacio, sin una sola línea de código adicional.

El lenguaje empezó como una forma de describir semánticamente documentos científicos y creció hasta ser la capa bajo casi toda aplicación con interfaz web. Sus primeros cinco años, 1990–1995, transcurrieron en el CERN y después en la IETF; HTML 3.2 se completó en el W3C en 1997 y HTML4 llegó poco después. Durante años, W3C y WHATWG mantuvieron textos divergentes, hasta que en 2019 firmaron un acuerdo para trabajar en uno: HTML Living Standard, mantenido por WHATWG e implementado por todos los principales motores de navegador. No tiene número de versión. A la pregunta «¿esto es HTML5?», el estándar responde con un breve sí y llama al término una palabra de moda para la plataforma web moderna, no una versión publicada.

¿Dónde se sitúa junto a aquello con lo que se compara? Markdown es una notación abreviada que una herramienta de construcción convierte en HTML. PDF fija la composición en una página impresa, mientras HTML se redistribuye según la pantalla que lo recibe. XHTML es el mismo vocabulario escrito con las reglas más estrictas de XML. Las plantillas JSX, Jinja y Hugo producen HTML, y lo que llega al navegador desde todas ellas sigue siendo HTML. Un equipo que compara un sitio hecho a mano con un constructor de sitios elige quién escribe el marcado, nunca si lo habrá.

Utilice el concepto en una decisión

Aplique «¿Qué es HTML?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Interfaces que cargan rápido, funcionan en el teléfono y se comportan igual en los navegadores que utilizan sus clientes.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Dirección del desarrollo integral de mapas GIS — cartografía integral, de la base de datos al navegador

Dirección del desarrollo integral de mapas GIS Póngase en contacto →

Por qué importa

Pocos equipos eligen HTML deliberadamente. Llega como lo que emite su framework, constructor de páginas o gestor de contenidos. La elección real es si esa salida es un entregable que alguien inspecciona o un subproducto que nadie lee, y las consecuencias aparecen en tres ámbitos.

En primer lugar importa quién puede usar la página. La tecnología de apoyo lee el árbol, no el diseño. Quien usa lector de pantalla se mueve por encabezados y regiones; quien usa teclado pasa de un enlace o control a otro con el tabulador. Un `<div>` estilizado con un manejador de clic parece un botón para quien ve y usa ratón, pero es texto plano para los otros dos. Los requisitos de accesibilidad, como WCAG, se basan en gran medida en lo que expone el marcado, así que cumplirlos después del lanzamiento suele exigir reescribir plantillas, no ajustar colores.

En segundo lugar importa cuánto dura el contenido. Los navegadores aún abren páginas escritas en la década de 1990 porque la compatibilidad hacia atrás es un principio de mantenimiento del estándar. Los frameworks cambian cada pocos años. El contenido en HTML limpio y válido puede trasladarse a otro sistema,

archivarse, convertirse a otros formatos o recibir un diseño nuevo sin reescribirse. El enterrado en envoltorios anidados de un constructor suele tener que extraerse a mano.

En tercer lugar están la velocidad y el alcance. Una página que el servidor envía completa puede empezar a mostrarse conforme llegan los bytes. Una aplicación que envía una estructura casi vacía y construye el documento con JavaScript obliga a cada lector a esperar el script; todo lo que lee sin ejecutarlo, desde una vista previa de enlace hasta muchos rastreadores, solo ve esa estructura. Para un producto tras un inicio de sesión, donde la interacción es lo esencial, puede compensar. Para páginas de marketing, documentación y cualquier contenido que deba encontrarse mediante búsquedas, HTML completo es la opción predeterminada más barata.

El coste de las tres cosas es el mismo y pequeño: alguien en el equipo que lea HTML con tanto cuidado como el código. El marcado semántico cuesta casi nada si se escribe primero y muchísimo si se incorpora después.

Cómo se aplica en la práctica

Este sitio es el ejemplo más directo. Su página de créditos enumera lo que llega al navegador del visitante: HTML, una hoja de estilos, fuentes web locales e imágenes, sin JavaScript ejecutable. Cada página se escribe completa como archivo en disco antes de que alguien la pida, en vez de ensamblarse por solicitud desde una plantilla. El control de calidad obligatorio valida ese HTML con vnu y trata cada diagnóstico como error; la misma página enlaza al W3C Markup Validator, que ejecuta ese motor contra el estándar vivo, de modo que el lector puede comprobar la afirmación en vez de confiar en ella.

El trabajo anterior para clientes abordó HTML desde el otro lado. En treinta sitios construidos a partir de diseños visuales terminados, los archivos de Photoshop se convirtieron a HTML a mano. El criterio tenía dos caras: respetar espaciado, tipografía y detalles previstos por el diseñador, y ser lo bastante flexible para que el sitio siguiera funcionando la primera vez que alguien editara su contenido. Reproducir una maqueta es la mitad del trabajo; un marcado que sobreviva a la próxima edición es la otra.

Mantener HTML como forma canónica también deja espacio para publicar otros formatos a su lado. Cuando el sitio se replicó en Gemini y Gopher, la misma construcción produjo 777 documentos Gemini y 777 menús Gopher desde la

misma fuente en la etapa de implementación documentada. En una comparación aparte entre construcciones se midieron cero bytes HTML modificados al añadir esos formatos; el resultado no describe cambios editoriales ni visuales posteriores.

Dónde surgen los problemas

La composición cambia de formas que ninguna hoja de estilo explica. Una página sin `<!DOCTYPE html>`, o con uno antiguo y mal formado, lleva al navegador al modo quirks, que imita navegadores heredados. Alturas porcentuales, celdas de tabla e interlineados siguen entonces comportamientos antiguos que se apartan de las especificaciones CSS, y el resultado difiere del de la misma página en modo estándar. La solución es una línea al principio de cada plantilla.

El marcado roto se muestra y el árbol resulta distinto. El analizador HTML se recupera de cada error de una forma definida, idéntica en todos los navegadores, así que los errores rara vez se ven. Surgen después, cuando un selector o script busca un elemento donde el analizador no lo puso. El caso clásico es un elemento de bloque dentro de un párrafo:

```
<p>Opening line
  <div>A block placed inside it</div>
</p>
```

El analizador cierra el párrafo antes del `<div>` y convierte el `</p>` suelto en un segundo párrafo vacío. Ejecute un validador durante la construcción y haga que sus errores la detengan; la tolerancia del navegador es un mecanismo de recuperación, no una revisión.

Cada control se convierte en un `div`. Elementos `div` y `span` pulsables, encabezados elegidos por su tamaño, un subíndice para hacer texto pequeño: todo parece correcto y transmite lo equivocado a la tecnología de apoyo. Quienes usan teclado pasan de largo controles que nunca reciben foco y quienes usan lector de pantalla pierden los encabezados y regiones por los que navegan. Una página sin `<main>` o `<h1>` les obliga a leer desde arriba cada vez. Use el elemento que nombra la función, como `button`, `a`, `h2`, `main` o `nav`, y deje que CSS cambie su aspecto.

La página queda en blanco hasta ejecutar su script. Un sitio renderizado en el cliente envía poco más que `<div id="app"></div>` y construye el documento en el navegador. En un teléfono lento el lector espera; si el script falla o un filtro lo bloquea, la página queda en blanco definitivamente. Renderice el documento en el servidor o durante la construcción y añada interacción sobre un HTML que ya tenga sentido por sí mismo.

Lecturas adicionales

- HTML Living Standard — La especificación implementada por los navegadores, mantenida por WHATWG y actualizada continuamente.
- Introducción al estándar HTML — La historia del CERN al acuerdo de 2019 y la propia respuesta del estándar sobre HTML5.
- Análisis de documentos HTML — La recuperación de errores que comparten todos los navegadores, regla por regla.
- La sintaxis HTML — Doctype, elementos, atributos y qué hace que un documento sea conforme.
- Nu Html Checker — El motor de vnu y del validador de marcado del W3C y su uso por línea de comandos.
- Estándar DOM — El árbol que construye el analizador y las interfaces con las que los scripts lo modifican.

Glosario

Elija según sus restricciones reales

Describa el flujo de trabajo, las interfaces existentes y los usuarios que necesitan el cambio. Podemos definir una entrega comprobable y un traspaso que permita a su equipo mantener lo entregado.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es HTML?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/html/