Engineer

CSS

lenguaje que define la presentación visual de documentos

Qué es

CSS, abreviatura de Cascading Style Sheets u hojas de estilo en cascada, y todavía llamado a menudo CSS3, es el lenguaje de reglas que un navegador aplica a un documento para decidir cómo se dibuja cada elemento: sus colores, tamaño, espaciado, posición, tipografía y movimiento, tanto si el documento termina en un teléfono como en un monitor ancho o una hoja de papel.

Una hoja de estilos es una lista de reglas. Cada regla combina un selector, que elige elementos, con declaraciones, cada una formada por una propiedad y un valor:

```
.note {
    color: #3a3a3a;
    margin-block: 1.5rem;
}
```

La primera palabra del nombre es la que determina el trabajo diario. Es habitual que varias reglas reclamen la misma propiedad del mismo elemento, y la cascada decide cuál gana. Sus pasos principales, por orden, son el origen y la importancia (los valores predeterminados del navegador, la configuración del propio lector, la hoja de estilos del sitio y si una declaración está marcada con `!important`), después la capa de cascada, luego la especificidad y, por último, el orden de aparición, donde una regla posterior vence a otra anterior con el mismo peso. La especificidad se cuenta por selector mediante tres números: primero los ID; después las clases, los atributos y las pseudoclases; y por último los nombres de elementos. El selector universal `*` no suma nada a ninguno. La herencia actúa junto a la cascada: algunos valores, como el color y la fuente, pasan a los hijos de un elemento, mientras que otros, como los márgenes y los bordes, permanecen donde se definieron.

CSS no tiene un único número de versión. Después de CSS 2.1, el grupo de trabajo adoptó un modelo modular: el nivel 3 amplía 2.1 módulo a módulo y, a partir de ahí, cada módulo tiene su propio nivel, de modo que Selectors se redacta en el nivel 4 mientras los módulos sin equivalente anterior empiezan en el nivel 1. «CSS3» es el nombre popular de esa etapa modular, y el documento anual que define el CSS actual afirma claramente que no existe un CSS de nivel 4.

La regla con mayor peso práctico es cómo trata los errores el lenguaje. Un navegador que encuentra una propiedad desconocida o un valor inválido para su propiedad debe ignorar esa declaración y aplicar el resto de la regla. Esto permite añadir propiedades y valores nuevos sin romper los navegadores antiguos y hace posible que un sitio utilice una función que solo algunos admiten. El precio es el silencio. La página se renderiza, un poco mal, y la única señal es una línea tachada en las herramientas de desarrollo que alguien tiene que buscar.

La mayoría de las cosas que se comparan con CSS resultan ser formas de escribirlo. Sass y Less son preprocesadores que compilan a CSS. Los frameworks de utilidades como Tailwind lo generan a partir de los nombres de clase del marcado. Las bibliotecas CSS-in-JS lo emiten desde componentes JavaScript, y un atributo `style` coloca declaraciones directamente en un elemento. Cada opción cambia la forma de crear CSS; todas terminan en declaraciones que la cascada ordena exactamente igual.

Utilice el concepto en una decisión

Aplique «¿Qué es CSS?» 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

La decisión que CSS impone a un equipo es dónde se permite que una regla falle en silencio. El lenguaje descarta lo que no puede aplicar y deja ganar a una regla más fuerte sin decir nada, así que toda hoja de estilos con cierta antigüedad contiene declaraciones que no hacen nada. Lo que controla el equipo es si alguien se entera y cuándo.

La accesibilidad depende de ello más de lo que suele pensarse. El contraste, un anillo de foco visible, respetar la preferencia de movimiento reducido, un modo de colores forzados utilizable y un texto que resista la ampliación: cada uno se resuelve con unas pocas declaraciones y puede quedar desactivado por una regla de otro lugar que resulte más específica. La página parece estar bien para quienes no necesitaban esa función.

El coste de mantenimiento también depende de ello. Una hoja de estilos cuyo comportamiento nadie puede predecir solo crece, porque eliminar una regla podría romper una página que nadie comprobó. El resultado habitual es un archivo cuyo peso consiste en su mayor parte en sobrescrituras de sobrescrituras anteriores, y cada cambio se hace probando valores hasta que la pantalla parece correcta.

La verificación depende de ello de la forma más directa. Un navegador que acepta lo que puede e ignora el resto no puede servir de prueba. Un linter que lee el código fuente detecta erratas y valores inválidos antes de publicarlos. No puede ver lo que un minificador u otro paso de compilación haga después con el archivo, ni cuál de dos reglas válidas gana en una página concreta. Las comprobaciones del archivo servido y la comparación visual de las páginas renderizadas cubren lo que no puede detectar el linter.

Organizar pronto la cascada, con especificidad uniforme, capas con nombre y tokens de diseño guardados en propiedades personalizadas, exige un poco de disciplina en cada cambio. Reorganizar una hoja de estilos que creció sin ella suele costar una reescritura.

Cómo se aplica en la práctica

La página de créditos de este sitio describe un único paquete CSS, concatenado y con una huella digital calculada durante la compilación, con las flechas dibujadas a mano y las notas al margen trazadas solo con CSS, y Stylelint imponiendo sus reglas al código fuente en el control de calidad obligatorio. La misma página enlaza al validador CSS del W3C y avisa de lo que dirá. En agosto de 2026, su perfil aún no admitía `system-ui`, `@property`, `clamp()` en un `font-size` ni las animaciones controladas por desplazamiento, así que al comprobar /terms/ notificó 176 problemas, 175 de ellos sobre funciones que los navegadores ya incluyen y el sitio utiliza. El restante era real. Un `rotate: 0deg` en reposo perdió la unidad en el minificador; `rotate: 0` no es un ángulo y el navegador descartó toda la declaración, de modo que se publicó una animación sin la rotación con la

que se había escrito. La especificación de valores solo permite un cero sin unidad como ángulo en unos pocos casos heredados. Stylelint nunca vio el problema porque lee el código fuente y el minificador lo reescribe después; una comprobación propia de la empresa lee ahora lo que realmente se sirvió.

La especificidad lo explicaba todo en un trabajo de accesibilidad. El sitio respetaba la preferencia de movimiento reducido con una regla universal que desactivaba todas las transiciones, pero al emular esa preferencia 11 selectores seguían animándose. Un selector universal puntúa cero, así que cualquier regla con una clase lo superaba. La solución fue estructural, sin recurrir a un selector más fuerte: todo lo que se mueve se concentra ahora en una sección situada hacia el final de la hoja de estilos, y un linter falla si encuentra una transformación con transición en cualquier otro lugar. Los cambios graduales de color se permiten porque no son movimiento, y el movimiento debe escribirse como transición, no como animación por fotogramas clave, porque la transición es la que se cancela limpiamente. La lección documentada es breve: una regla que pierde por especificidad falla en silencio.

El color es donde primero se desvían las hojas de estilos, y la paleta del sitio se cerró en 28 colores, cada uno documentado con la relación de contraste que alcanza sobre la superficie en la que aparece, y 11 valores de marca declarados una sola vez como tokens. Dos colores separados por una distancia perceptual inferior a 0,02 cuentan como un mismo color escrito dos veces, y el linter encontró dos representaciones casi duplicadas en cuanto se ejecutó. La transparencia se escribe con sintaxis de color relativo en lugar de la función de mezcla por una razón que resume toda la página: la comprobación no puede ver a través de una mezcla.

Dónde surgen los problemas

Una declaración desaparece y la página sigue renderizándose

Un solo carácter incorrecto invalida una declaración y el navegador la omite sin detenerse:

```
.card {
    padding: 16; /* a length needs a unit: ignored */
    colour: #222; /* not a property name: ignored */
    margin-block: 1rem; /* applied */
}
```

La tarjeta se renderiza con margen y sin su relleno ni su color, y el navegador no da ninguna alarma. Ejecute el linter sobre el código fuente durante la compilación y haga que sus errores la detengan. Si un minificador, preprocesador o empaquetador reescribe CSS después de pasar el linter, compruebe también la salida, porque ese es el archivo que lee el navegador.

Una regla pierde frente a otra más fuerte

La regla de un componente se encuentra con un selector más específico de otro lugar, o con uno igual cargado después, y pierde. La reacción habitual es alargar el selector o añadir `!important`, que gana hoy y encarece el siguiente cambio. Mantenga la especificidad uniforme, con clases en lugar de ID y poco anidamiento, y utilice capas de cascada para ordenar explícitamente grupos completos como reset, base, componentes y utilidades, sin que la especificidad pueda alterar ese orden. Para las declaraciones `!important`, el orden de las capas se invierte y gana la primera, algo que conviene saber antes de recurrir a ellas.

La hoja de estilos solo crece

El trabajo nuevo añade reglas porque demostrar que una antigua no se usa lleva más tiempo que escribir otra. Los valores se vuelven a escribir como literales junto a los tokens que los definen y los colores casi idénticos se multiplican hasta que nadie sabe cuál se pretendía usar. Guarde los valores compartidos en propiedades personalizadas, mida qué reglas utiliza realmente un conjunto de páginas y elimine las que no se usan como parte del trabajo, con un presupuesto de tamaño en la compilación que haga visible el crecimiento.

El diseño solo se sostiene en un tamaño

Un diseño comprobado en un solo ancho con texto breve en inglés se rompe con traducciones más largas, texto ampliado y teléfonos estrechos. Las alturas fijas recortan el contenido y los anchos fijos obligan a desplazarse lateralmente. Las pautas de accesibilidad piden texto que pueda ampliarse al 200 % y contenido que se redistribuya a un ancho de 320 píxeles CSS. Dimensione las cajas a partir de su contenido, use `min()`, `max()` y `clamp()` para valores fluidos y pruebe cadenas largas y zoom antes de dar por terminado un diseño.

Lecturas adicionales

- CSS Snapshot 2026 — qué módulos componen CSS hoy y por qué el lenguaje tiene niveles en lugar de versiones.
- Cascada y herencia de CSS, nivel 5 — el orden completo de resolución, las capas de cascada y la herencia, paso a paso.

- Selectores, nivel 4 — todos los selectores y cómo se calcula su especificidad.
- Sintaxis y tipos de datos básicos — las reglas de CSS 2.1 para ignorar propiedades desconocidas y valores ilegales, con ejemplos.
- Módulo de valores y unidades de CSS, nivel 4 — longitudes, ángulos, `clamp()` y los casos en que todavía se permite un cero sin unidad.

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 CSS?» 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/css/