

Hoja de estilos CSS

1. Conceptos básicos I

El CSS (Cascade Style Sheet) es un lenguaje utilizado para dar estilo y formato a los elementos HTML en una página web. Se estructura mediante reglas que especifican cómo deben ser presentados los elementos en el navegador. Estas reglas consisten en un selector y una declaración.

- **Selector:** Indica a qué elementos HTML se aplicará la regla.
- **Declaración:** Define el estilo que se aplicará a los elementos seleccionados.

Existen tres formas de aplicar CSS a un documento HTML:

- **In line:** Se define dentro del propio documento HTML usando la etiqueta `<style>` dentro de `<head>`.
- **Externo:** Se enlaza un archivo CSS externo mediante la etiqueta `<link>` dentro de `<head>`, lo que permite reutilizar el mismo estilo en múltiples archivos HTML.
- **Inyectado:** Se aplica directamente al elemento HTML usando el atributo `style`. No es recomendable porque dificulta el mantenimiento y separación de estilos y contenido.

Selectores

- **Selector universal:** Se aplica a todos los elementos del documento.
- **Selector de etiqueta:** Se aplica a los diferentes elementos o etiquetas de HTML.
- **Selector de clase:** Se aplica exclusivamente a aquellos elementos que, en el HTML, tengan un atributo `class=""`.
- **Selector de ID:** Se aplica a aquellos elementos con un id en el HTML.

- **Selector de “hijos”:** Se aplica a todo aquel elemento que cumpla la condición de ser “hijo” directo de otro elemento.
- **Selector de descendientes:** Al igual que el Selector de “hijos” permite apuntar a un elemento anidado dentro de otro elemento (en el HTML), pero en este caso no necesita ser un hijo “directo”.
- **Selector de “parientes” adyacentes:** Apunta a cualquier elemento que este a continuación de otro elementos.
- **Selector de atributos:** Permiten seleccionar elementos HTML en función de sus atributos y/o valores de esos atributos.

Otros selectores

- **Pseudoselectores (Pseudoclases):** Estos selectores no seleccionan un elemento en sí, sino un estado del elemento.
- **Pseudoelementos:** Éstos definen un set de elementos ya seleccionados, como por ejemplo `::first`, `::nth-child()` o `::after`.
- **Elementos Especiales:** Son selectores para casos muy específicos o especiales, como la barra de desplazamiento (`::-webkit-scrollbar`) o imágenes rotas (`:-moz-broken`).

El “estilo en cascada”

El estilo "en cascada" en CSS determina cómo se aplican las reglas de estilo cuando hay conflictos.

- **Última Regla:** La regla más reciente en el código tiene prioridad sobre las anteriores.
- **Especificidad:** Cuanto más específico sea un selector, más prioridad tendrá sobre los menos específicos.
- **!important:** Se puede agregar a una declaración para darle prioridad sobre otras reglas, aunque su uso se desaconseja debido a la dificultad de mantenimiento.

En caso de colisiones de estilos, se siguen varios pasos para determinar qué regla tiene prioridad, incluyendo el origen y la especificidad del selector.

CSS permite la herencia de estilos, donde un estilo aplicado a un elemento se transmite a sus elementos secundarios, aunque no todos los estilos se heredan automáticamente.

Es recomendable probar el diseño en varios navegadores y sistemas operativos para detectar incompatibilidades, siendo Internet Explorer útil como referencia debido a su falta de compatibilidad con estándares modernos.

2. Conceptos básicos II

Custom Properties

Son propiedades que se pueden definir personalmente. Se declaran en :root para ser accesibles globalmente, y se utilizan con var().

Fuentes y tipografías

Se usa font-family para declarar fuentes y @font-face para importar fuentes no instaladas.

Propiedades importantes: font-size, font-weight, line-height.

Animaciones

- **Transition:** Se usa para transiciones suaves entre estados de elementos, como :hover.
- **Keyframes:** Define pasos de animaciones específicos usando @keyframes.

Filtros de contenido

- **Filter:** Aplica efectos de filtro como blur, brightness, contrast, etc., a elementos.
- **Backdrop-filter:** Similar a filter, pero afecta a los elementos detrás del elemento actual.
- **Gradientes:** Permite crear gradientes lineales, radiales, repetidos y cónicos para aplicar a fondos u otros estilos.

3. Reglas y especificidad de CSS

Agrupación de reglas

En archivos CSS complejos, es común definir estilos para un mismo selector en diferentes reglas.

Las reglas con el mismo selector pueden agruparse para hacer las hojas de estilos más eficientes y fáciles de mantener.

Especificidad

Se refiere al peso o importancia de los selectores en una hoja de estilo.

Cuanta más especificidad tenga un selector, menos posibilidades habrá de que se aplique un estilo incorrecto.

Los navegadores utilizan la especificidad para decidir qué reglas CSS aplicar a los elementos.

Cálculo de la especificidad CSS

Se calcula sumando los puntos asignados a cada selector.

Un elemento simple (etiqueta) tiene 1 punto, una clase tiene 10 puntos, un ID tiene 100 puntos, y los atributos de estilo tienen 1000 puntos.

Por ejemplo, un selector como `p` tiene 1 punto, mientras que `.par p` tiene 11 puntos (10 de la clase y 1 del elemento), y `#parr p` tiene 101 puntos (100 del ID y 1 del elemento).

4. Modelo de caja

CSS trata cada elemento HTML como una caja, lo que permite aplicar diferentes estilos a esas cajas. Estos elementos se clasifican en dos tipos principales:

- **Block (Bloque):** Ocupa todo el espacio disponible y comienza en una nueva línea, a menos que se modifique con estilos CSS.
- **Inline (En línea):** Ocupa solo el espacio necesario para mostrar su contenido y no inicia una nueva línea.

Propiedades de la caja

- **width, height:** Especifican las dimensiones del elemento (no aplicables a elementos en línea).
- **border:** Define el borde del elemento.
- **margin:** Determina la distancia entre el elemento y los elementos circundantes (solo los márgenes horizontales se aplican a elementos en línea).
- **padding:** Establece el espacio interior alrededor del contenido del elemento.

Cada caja en HTML consta de varias partes importantes:

- **Contenido de la Caja (Content Box):** El área donde se muestra el contenido del elemento, cuyo tamaño se puede ajustar con propiedades como width y height.
- **Relleno de la Caja (Padding Box):** El espacio en blanco alrededor del contenido, controlado mediante la propiedad padding y otras propiedades relacionadas.

5. Posicionamiento de los elementos

Position

La propiedad position en CSS nos permite controlar la ubicación de los elementos en una página web. Aquí tienes una descripción de los valores que puede tomar:

- **Static (Estático):** Es el valor por defecto. Los elementos se colocan en el orden en que aparecen en el flujo del documento HTML.
- **Relative (Relativo):** Altera la posición del elemento en relación con su posición original. Aunque se mueva, dejará un espacio en su posición original.
- **Absolute (Absoluto):** Levanta el elemento y lo "saca de flujo", permitiendo que flote sobre otros elementos. Se posiciona en relación con su contenedor más cercano con una propiedad de posición diferente a static.
- **Fixed (Fijo):** Similar a absolute, pero el elemento se mantiene fijo en la ventana del navegador, incluso cuando se desplaza la página.
- **Sticky (Adhesivo):** El elemento se comporta como relative hasta que alcanza cierto punto de desplazamiento, momento en el que se convierte en fixed y se mantiene en esa posición.
- **z-index:** Controla la "profundidad" de los elementos posicionados. Es útil cuando hay varios elementos posicionados y queremos controlar cuál está encima de cuál. Un valor mayor de z-index indica que el elemento está más arriba en la pila de capas.

Flex

- Permite colocar elementos en una fila horizontal o vertical.
- La fila puede envolverse o flexionarse cuando los elementos no caben.
- Se pueden alinear los elementos a la izquierda, derecha, arriba, abajo, etc.

Grid

- Permite colocar elementos en posiciones fijas predefinidas, similar a una tabla.
- Se define con `display: grid`.
- Se pueden establecer columnas (`grid-template-columns`) y filas (`grid-template-rows`) para organizar los elementos.

6. Bootstrap

Bootstrap es un framework de CSS y HTML que permite a los desarrolladores crear diseños y prototipos de manera rápida y con una estética cuidada.

Instalación mediante CDN

- Enlaza las librerías necesarias desde el CDN de Bootstrap a tu documento HTML.
- Coloca el enlace a la hoja de estilo de Bootstrap dentro de la etiqueta `<head>` antes que cualquier otra hoja de estilo.
- Importa el script JavaScript necesario al final de la página, justo antes de la etiqueta de cierre `</body>`.
- El starter template de Bootstrap proporciona una estructura básica para comenzar un proyecto.
- Descarga los archivos CSS y JS de Bootstrap y guárdalos en las carpetas de tu proyecto.

Archivos bootstrap CSS

Archivos CSS	Diseño	Contenido	Componentes	Utilidades
bootstrap.css bootstrap.rtl.css bootstrap.min.css bootstrap.rtl.min.css	Incluido	Incluido	Incluido	Incluido
bootstrap-grid.css bootstrap-grid.rtl.css bootstrap grid.min.css bootstrap-grid.rtl.min.css	Único sistema de cuadrícula	–	–	Solo utilidades flexibles
bootstrap-utilities.css bootstrap-utilities.rtl.css bootstrap-utilities.min.css bootstrap-utilities.rtl.min.css	–	–	–	Incluido
bootstrap-reboot.css bootstrap-reboot.rtl.css bootstrap-reboot.min.css bootstrap-reboot.rtl.min.css	–	Solo reiniciar	–	–

Breakpoints Responsive Bootstrap

Archivos JS	Class infix	Dimensions
X-Small	None	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥922px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

Contenedores

Los contenedores son un bloque de construcción fundamental de Bootstrap que contienen, rellenan y alinean su contenido dentro de un dispositivo o ventana determinada.

Clases	Extra pequeño <576px	Small≥ 576px	Medium≥ 768px	Large≥ 992px	XLarge≥ 1200px	XXLarge≥ 1400px
.container	100%	540px	720px	920px	1140px	1320px
.container sm	100%	540px	720px	960px	1140px	1320px
.container md	100%	100%	720px	960px	1140px	1320px
.container lg	100%	100%	100%	960px	1140px	1320px
.container xl	100%	100%	100%	100%	1140px	1320px
.container xxl	100%	100%	100%	100%	100%	1320px
.container fluid	100%	100%	100%	100%	100%	100%

Sistema de GRID

El sistema de cuadrícula de Bootstrap utiliza contenedores, filas y columnas para diseñar y alinear el contenido.

- Los contenedores centran y rellenan horizontalmente su contenido. Puedes usar `.container` para un ancho máximo del contenedor o `.container-fluid` para ocupar el ancho completo de la ventana del navegador.
- Está basado en un sistema de 12 columnas realizadas con flexbox. Cada fila debe estar envuelta en un contenedor con la clase `.row`.
- Las columnas suman un total de 12 unidades. Por ejemplo, puedes tener una fila con una columna de ancho 8 y otra columna de ancho 4, sumando un total de 12 unidades.
- Bootstrap utiliza breakpoints para definir el comportamiento de las columnas en diferentes tamaños de pantalla, desde `xs` hasta `xxl`.

Gutters

Los gutters son el espacio entre las columnas, que se utiliza para alinear el contenido.

Puedes ajustar el espacio intercolumna con las clases `g-*`, donde `*` va desde 0 hasta 5 según el tamaño del gutter que desees.

Puedes controlar los gutters horizontales y verticales utilizando `gx-*` y `gy-*` respectivamente.

Utilidades de layout

Bootstrap ofrece clases de utilidad para mostrar, ocultar, alinear y espaciar el contenido. Puedes cambiar rápidamente el valor de visualización de los componentes con las clases `d-*`. También puedes utilizar clases de utilidad para aplicar espaciado y relleno a los elementos con las clases `m-*` y `p-*`.

Las clases de espaciado se pueden aplicar a todos los breakpoints, y puedes controlar el tamaño del margen y el relleno con diferentes valores.

Iconos

Bootstrap ofrece su propia librería de iconos con licencia MIT. Además, ofrece alternativas como Font Awesome, Feather y Octicons, que proporcionan una amplia variedad de iconos para utilizar en tus proyectos.

Componentes

Bootstrap incluye una gran variedad de componentes prefabricados, como carruseles, alertas, acordeones y ventanas modales. Estos componentes te permiten implementar funcionalidades comunes de forma rápida y sencilla en tus proyectos.

Puedes personalizar los componentes modificando las clases de Bootstrap o utilizando tu propia hoja de estilos.