

Clase 2 – Introducción CSS

Las hojas CSS (Hojas de estilos en cascada) aplican estilos graficos a nuestro HTML. Un elemento puede ser afectado por mas de un selector y se va a aplicar segun la especificidad (CSS->HTML(head)->Etiqueta o selector de etiquetas->clase->id o CSS de arriba hacia abajo). En algunos casos las propiedades que coincidan con los selectores se van a pisar, si en el ultimo estilo no se afecto un estilo del primer estilo aplicado se aplica este primer estilo. Más adelante veremos en profundidad la especificidad.

Existe una regla en CSS que se declara !important, eso va a romper la regla en cascada y va a priorizar ese estilo por mas que en terminos de especificidad después se vuelve a afectar.

Existen 2 tipos de modificadores para realizar estilos CSS en el mismo documento HTML

- Estilo en linea: Se utiliza la propiedad style dentro de una etiqueta en mi HTML.
- Estilos a nivel de pagina: En el head con la etiqueta style va a generar un bloque de codigo para escribir CSS. Puedo seleccionar por tag, #id y .class, se comporta como una hoja .css.

Ambas son malas practicas ya que para modularizar debemos tener nuestro CSS en un archivo aparte para el html.

En teoria los id no se puede repetir, pero si los hubiera nos va a seleccionar el primero que aparezca cuando utilicemos JS. En cambio en CSS va a seleccionar todos los id para aplicarles estilos.

Las clases se crearon para aplicar las mismas reglas de estilos a distintos tipos de etiquetas en HTML. Asi un h1 puede compartir las mismas reglas que un h2, por ejemplo.

Un selector de etiquetas/tipos lo voy a utilizar para aplicar estilos a todas las etiquetas que sean el mismo tipo.

Un selector de id lo voy a utilizar para aplicar estilos a un unico elemento.

Un selector de atributos lo voy a utilizar para seleccionar elementos de la página según sus propiedades o el valor asignado a estas propiedades.

El model box se divide en margin-border-padding-content.

CSS shorthand para el Model-Box:

- 1 valor -> Cuatro puntos cardinales
- 2 valores -> Arriba/Abajo - Izquierda/Derecha
- 3 valores -> Arriba - Izquierda/Derecha - Abajo
- 4 valores -> Arriba - Derecha - Abajo - Izquierda

Por default los elementos en bloque tienen margin arriba y abajo.

Los márgenes verticales se colapsan, no se suman los px de un margen entre dos elementos que coincidan en su margen. Básicamente se solapan los márgenes.

Los márgenes de Izquierda a Derecha en cambio no se solapan y se suman los píxeles.

- La propiedad colspan en un tr o th indica cuantos espacios en el eje de las columnas va a ocupar una sola columna. Por ejemplo colspan="2" ocupa el espacio equivalente a dos columnas.
- La propiedad rowspan funciona de manera identica al colspan solo que afecta al eje y (fila) de la columna. Por ejemplo rowspan="2" ocupa el espacio sobre el eje de las filas de dos columnas.

En ambos casos si se superpone con una columna/fila va a empujar ese elemento a la derecha en su mismo eje de fila. Va a empujar todos los elementos hacia la derecha X veces hasta que no interfiera.

Enlaces

Los enlaces son lo que dan a HTML hipervínculo, representada con la etiqueta `<a>`. Estos enlaces nos permiten saltar a referencias internas con la dirección PATH interna, externas con la URL o ir hacia un id determinado con el `#id`.

En la propiedad `href` vamos a poner el PATH de donde queremos ir. Si es una ruta externa tenemos que poner el protocolo de la página. Ejemplo `https://google.com.ar` o `https://www.google.com.ar`. Con la propiedad `target="_blank"` vamos a abrir una nueva pestaña. Con `target="_self"` se abre en la misma página, es la propiedad por default.

Propiedades interesantes: `https://comohacerunapagina.es/que-es-relnopener-noreferrer-en-wordpress/` o `https://acuere.es/2017/10/04/nopener-noreferrer/` - `https://www.humanlevel.com/diccionario-marketing-online/nofollow`

Podemos agregar un `#` después de la ruta a acceder si queremos ir a un id específico dentro del PATH indicado. Ejemplo: `Contactos`

Para agregar una imagen utilizamos la etiqueta `img`, por default nos trae la propiedad `src="PATH a la imagen"` y `alt="QUE ES LA IMAGEN"`. El `alt` lo utilizamos por si falla la carga de la imagen poner un texto placeholder.

JavaScript

latenflip loup: Para ver de manera más visual como funciona JS con su asincronía y concurrencia.

Existen los conceptos de síncronico y asíncronico. También el concepto de bloqueante y algo no bloqueante.

Normalmente lo síncronico está asociado con bloqueante y lo asíncronico con no bloqueante. También tenemos los conceptos de concurrencia y paralelismo.

JS es un lenguaje asíncronico, no bloqueante y concurrenente. C# es síncronico e implementa paralelismo (tiene hilos), con esos hilos puede implementar el asincronismo y no bloquear el hilo principal. En cambio JS no tiene hilos, tiene un event loop y aplica concurrencia.

- Concurrencia (JS): Varios procesos son atendidos de manera no-simultánea en una misma pila de ejecución. El event loop es el que los ejecuta.
- Paralelismo (C#): Más de un núcleo realizando procesos distintos en una pila de ejecución distinta.
- API: Una API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de las aplicaciones.