

CSS

Contenido

Introducción.....	2
Sintaxis y estructura	2
Aplicación de un CSS.....	2
CSS Externo	2
CSS Interno.....	2
Declaración in-line	3
Prioridades y orden	3
Valores y unidades:.....	4
Tipos de datos	4
Unidades de longitud absoluta	4
Unidades de longitud relativas.....	4
Colores.....	4
Imágenes	4
Tipos de letra	5

Introducción

CSS (Cascading Style Sheets) es el lenguaje que se usa para dar aspecto a las paginas HTML y XHTML

Se diseñó para separar la información del html de sus reglas de presentación, ya que previamente estaba todo junto en el mismo documento

La separación proporciona versatilidad para presentar información, puesto que se pueden poner varios CSS en un mismo documento HTML

La primera vez que CSS apareció fue en 1996. Hay distintas versiones de CSS, pero la versión CSS3 permite especificar las ventanas por módulos.

Sintaxis y estructura

CSS presenta unas reglas que se van a aplicar a un número indeterminado de elementos en el HTML al que se vinculan, por lo que necesitan dos herramientas:

- Selector: Las herramientas que seleccionan al elemento/s que van a cumplir esas reglas
- Declaraciones: Indicadores para asignar el aspecto deseado de los elementos

```
body{ //Selector
    background-color: white; //Declaración
}
```

Aplicación de un CSS

Para aplicar un CSS en un HTML, tienen que estar vinculados. Se puede realizar de tres formas **no incompatibles entre sí**, es decir, que se pueden usar a la vez.

CSS Externo

Las reglas CSS se almacenan en otro documento, al que luego se accede en el html con `<link rel="stylesheet" type="text/css" href="(url)">`

CSS Interno

Se declaran en el `<head>` de un documento HTML, más precisamente en el apartado `<style>`

```
<html>
  <head>
    <style>
      body{
        background-color:white;
      }
    </style>
  </head>
</body>
</html>
```

Declaración in-line

Se declaran en la propia marca que va a afectar, en la propiedad style:

```
<body style="background-color: white">
</body>
```

Prioridades y orden

Pueden existir tres tipos de hojas de estilo distintas: La propia del navegador, la que proporciona el usuario, y las hojas de estilo del diseñador. Se aplican en ese orden por defecto.

Hay tres conceptos fundamentales a la hora de aplicar estilos:

- **Herencia:** Los valores de algunas propiedades aplicadas a un elemento contenedor se aplican también en los elementos que contiene. No pasa con todas las propiedades, pero, por ejemplo, si a un contenedor <div> le ponemos *color: red*; todo su texto saldrá en rojo salvo que se diga lo contrario
- **Cascada y especificidad:** Los factores que participan en qué estilo aplicar son los siguientes:
 - o **Origen:** Los estilos inherentes al navegador tienen menos prioridad a los elegidos al usuario, y estos menos que los que elija el diseñador. Dentro de los de diseñador, tiene mas prioridad si esta definido en in line, en el documento y el que menos prioridad es el css externo (esto es lógica)
 - o **Especificidad:** Cuanto más se especifique, más importancia (ids, clases, etc)
 - o **Orden de aparición:** Cuanto más abajo y a la derecha esté, más prioritario

Las reglas fundamentales de aplicación también aparecen en caso de conflicto:

- El selector universal (*) se impone al resto de reglas definidas con selectores más específicos
- Si una propiedad se modifica una única vez, no hay conflicto
- Si se modifica en el atributo style, se impone al intento de cualquier modificacion
- Si se modifica en el elemento <style>, se impone a la modificación en un CSS externo
- Si dos o mas CSS externos afectan a una propiedad, aplica el último añadido
- También se tiene en cuenta la **especificidad**:
 - o Si tiene más identificadores, se pone de prioridad principal
 - o Si tiene mas numero de clases o pseudoclases, será el siguiente en la prioridad
 - o Si tiene más elementos o pseudoelementos, será el siguiente en la prioridad
 - o Si tienen la misma prioridad, se aplicará el último añadido

Resumen:

- Style > 'style' > 'link rel="stylesheet"'
- Id > clase > pseudoelementos

Aun así, estas reglas se pueden editar u omitir mediante unos valores:

- **Inherit:** Activa la herencia, coge lo del elemento padre
- **Initial:** Toma el valor de la hoja de estilos por defecto
- **Unset:** Restablece el valor al natural
- **Revert:** Restablece el valor al que habría tenido si no hubiera cambiado
- **!important:** Anula las propiedades de un elemento por las reglas de asignación

Valores y unidades:

Tipos de datos

- Entero: Numero entero positivo o negativo
- Número: decimal, usando el punto
- Dimensión:
 - o Deg – grados
 - o S – segundos
 - o Px – pixeles
- Porcentaje: Una fracción del total del referente padre

Unidades de longitud absoluta

- Px – pixel
- Cm – centímetro
- Mm – milímetro
- In – Pulgada (inch)

Unidades de longitud relativas

- Em: Tamaño de letra del elemento padre
- Ex: Altura de la fuente de un elemento
- Ch: Ancho del carácter 0 de la fuente
- Rem: Tamaño de la letra del elemento raíz

(normalmente se usan em y rem (cuando se usan (casi nunca)))

Colores

Se pueden emplear para fondos, textos y bordes. Se emplean distintas notaciones:

- Nombre en inglés
- Valor RGB (00, 00, 00)
- Valor hexadecimal (#000000)
- Valor HSL (Hue Saturation Lightness, Matriz Saturación Luminosidad), siendo el primero de 0 a 360 y los otros de 0 a 100

Imágenes

Podemos añadir imágenes a cierto contenido del css con el comando url(“url-imagen.png”)

Tipos de letra

CSS define 5 tipos de letra genéricos (serif, fantasy, sans-serif, cursive y monospace), pero se pueden añadir más haciendo referencia a fuentes no genéricas en la hoja CSS. Si el dispositivo no las tiene descargadas, se podrán añadir como stylesheet desde un servidor que guarde la tipografía deseada

```
@font-face{
  Font-family:NombredelaFuente;
  src: url(url-de la fuente")
}
```

Otra opción es:

```
<link
href="https://fonts.googleapis.com/css?family=Sofia"
rel="stylesheet" type='text/css'>

body{
  h1{
    font-family: 'Sofia', sans-serif;
  }
}
```

O también:

```
@import
url('https://fonts.googleapis.com/css2?family=Roboto');
```

```
body{
  font-family: 'Roboto', sans-serif;
}
```