CSS(Cascading Style Sheets) es un lenguaje de presentación creado para dar estilo al contenido.

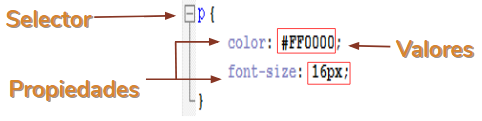
**Términos**

Selector: Determina el estilo que se va a aplicar a un elemento. Los selectores están antes de cualquier {..}

Propiedades: El estilo que se aplica puede ser una o múltiples propiedades que están contenidas dentro de {...} y el texto seguido de : (dos puntos).

Valor: Define el comportamiento de la propiedad.

Es el valor que va después de los : (dos puntos) y antes del ; (punto y coma).



**¿Cómo incluir una hoja de estilo CSS?**

Hay 3 formas de hacerlo:

**Hoja de estilos externa.**

Hoja de estilos interna .

Hoja de estilo en línea.

La mejor práctica para usar CSS es usar un archivo externo, referenciado en la sección head.

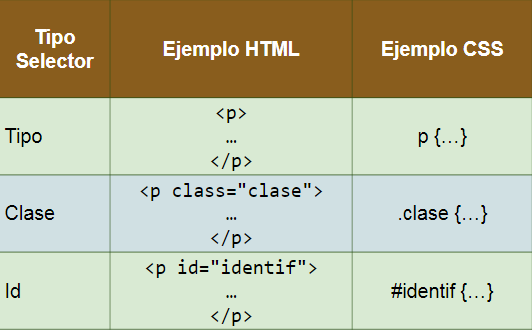
Usando un archivo CSS externo, se pueden usar los mismos estilos en todo el sitio sin replicar código.

<head>

<link rel=”stylesheet” type=”text/css” href=”estilo.css”>

</head>

**Selectores - Resumen**

****

**Selectores - Tipo**

Son los más básicos

Funciona con los elementos sin ningún atributo especial

Hay que tratar de usarlos, cuando es posible, porque son más fáciles de manejar

**Selectores - Clase**

Permite ponerle el mismo estilo a una lista de elementos, agregándole el atributo class.

**Selectores - ID**

Utilizados para un único elemento por página.

**Selector universal**

Se utiliza para seleccionar todo los elementos de la página.

El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML

\*{

margin: 0;

padding: 0;

}

**Selector - “name”**

Se utiliza agregando el atributo name a la etiqueta HTML de la siguiente forma.

<p name=”mielemento”>Hola</p>

Y en CSS de la siguiente manera:

p[name=”mielemento”]{

color: blue;

}

Este atributo contiene 3 variantes:

^: Ej. p[name^=”mi”] Si la palabra “mi” esta al comienzo del atributo name se aplica el estilo.

$: Ej. p[name$=”mi”] Si la palabra “mi” se encuentra en el final del atributo se aplica el estilo.

**Fondo - Imágenes**

Se puede usar una imágen de fondo.

La propiedad es background-image.

Por defecto la imagen se repite para cubrir todo el fondo.

body {

background-image: url (imagen.jpg);

}

Para que la imagen no se repita, hay que usar una propiedad background-repeat.

body {

background-image: url(imagen.jpg);

background-repeat: no-repeat:

}

**Propiedades de texto - Alineación**

La propiedad text-align es usada para alinear horizontalmente, esta puede ser:

Centrado (text-align: center;)

Derecha (text-align: right;)

Izquierda (text-align: left;)

Justificado (text-align:justify;)

**Propiedades de texto - Decorations**

La propiedad text-decoration es usada para eliminar o agregar algunas decoraciones al texto.

Mayormente se usa para eliminar el subrayado a los links text-decoration: none;

**Propiedades de texto - Transform**

Mayúscula (text-transform: uppercase;)

Minúscula (text-transform: lowercase;)

Capitalizado (text-transform: capitalize;)

**Familia de fuentes**

La familia de fuentes se indica con la propiedad font-family.

La font-family debe tener varias fuentes por si el browser no soporta alguna.

La primera fuente debe ser la que uno quiere, la última una genérica.

Si el nombre de la font tiene más de un nombre/palabra se escribe entre comillas. EJ “Times new Roman”

Más de una font-family se especifica por comas.

**Insertar una font en CSS**

@font-face{

font-family: “Nombre a elección”;

src: url(‘ruta del archivo’);

}

**Fuentes - Estilos**

La propiedad font-style es usada mayoritariamente para especificar la cursiva

font-style: normal – El texto se muestra normal

font-style: italic – El texto se muestra en cursiva

font-style: oblique – Es como cursiva, pero menos soportado.

**Propiedad - text-shadow**

Se utiliza de la siguiente manera:

text-shadow: color horizontal(px/%) vertical(px/%) difuminación(px/%)

Sirve para darle sombra al texto.

**Box Model**

El concepto de “Box Model” dice que cada elemento en una página se representa mediante una caja rectangular (contenedor).

CSS utiliza el modelo cajas/bloques que consta de 4 partes:

CONTENT

PADDING

BORDER

MARGIN



CONTENT: Alto (height) y Ancho (width) de un elemento.

PADDING: Usado para generar espaciado o margen interior transparente dentro de un elemento.

BORDER: Se utiliza para bordear con una línea alrededor del elemento.

MARGIN: Usado para generar margen exterior transparente fuera de un elemento (puede usarse para separar bloques)

**Unidades de medida**

CSS divide las unidades de medida en dos grupos:

**Absolutas: pixeles (px)**

Están completamente definidas, ya que su valor no depende de otro.

Ajustan tamaños fijos en los navegadores y pantallas.

Poca flexibilidad.

Sirve cuando conocemos tamaños de las salidas.

**Relativas: porcentajes (%)**

No están completamente definidas, ya que su valor siempre es dependiente respecto a otro valor de referencia padre.

Permiten ajustes con cambios de tamaño de pantalla.

Mayor flexibilidad.

**Medida “Auto”**

La opción auto significa “expandir” o ajustar automáticamente un tamaño.

Para centrar un bloque

.bloque{

width: 100px;

margin: 0 auto;

}

Para hacer que ocupe el máximo espacio

.bloque{

width: auto;

}

**Tipos de cajas (Type of boxes)**

CSS puede definir la manera en la que los elementos de una página “encajan” uno con otros. Estas maneras se pueden categorizar en:

Block

Inline

Se controlan con la propiedad display

Elemento {

display: block | inline;

}

**Tipos de cajas - Block vs Inline**

Block

Elemento-1{

display: block;

}

Elemento-2{

display: block;

}

Ocupan el ancho total del elemento contenedor.

Las dimensiones son controlables.

Insertan un salto de línea en el flujo al terminar.

Inline

Elemento-1{

display: inline;

}

Elemento-2{

display: inline;

}

**Otras propiedades de posicionamiento**

display: inline-block; Es una combinación de los dos tipos anteriores.

display: none; Oculta el elemento de la página.

**Flotar elementos**

La propiedad CSS float permite a los elementos colocarse del lado izquierdo o derecho de su contenedor.

Cuando se flotan elementos, van a ir hasta el borde de su contenedor padre, si no tiene padre va a ir hasta el borde de página.

Cuando un elemento es flotado rompe con el flujo normal de la página, para volver al flujo normal de la página siempre usar la propiedad “clear: both”.

Cuando un elemento tiene la propiedad clear, a partir de ese punto en adelante la página vuelve al flujo normal.

**Posicionamiento**

La propiedad position sirve para posicionar un elemento dentro de la página.

Muy útil cuando queremos posicionar elementos fuera del flujo normal de la página.

Dependiendo de cual sea la propiedad que usemos, el elemento tomará una referencia u otra para posicionarse.

**Herencia**

La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un elemento padre se transmiten a sus hijos.

**¿Para que nos sirve entonces la herencia?**

Escribir menos código.

Si ponemos la font-family al elemento HTML

* todos los elementos lo heredan
* no tengo que re escribirlo para que otros lo tengan
* más mantenible
* menos redundante (menos código repetido)

la redundancia lleva a inconsistencia

**Selectores combinados**

Se pueden combinar selecciones para hacerlas más específicas.

p.destacado{ color: red; }

Elige los párrafos que tengan la clase destacado.

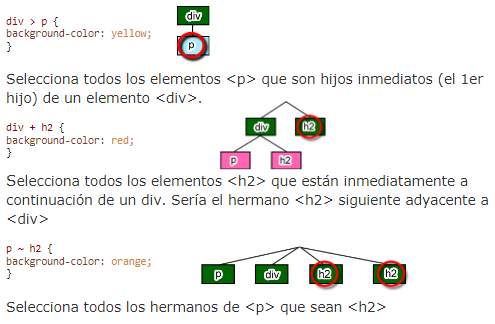
**Selectores anidados**

Permite seleccionar elementos contenidos dentro de otros elementos. Así se puede aumentar el nivel de detalle.

Selecciona los span que estén dentro de algún párrafo (incluye si está contenido indirectamente).

p span{ color: blue;}

**Combinaciones de selectores**



**Cascada**

La cascada es el mecanismo que controla el resultado final cuando se aplican varias declaraciones CSS contrapuestas al mismo elemento.

Hay tres conceptos principales que controlan el orden en el que se aplican las declaraciones de CSS:

Importancia.

Especificidad.

Orden en el código fuente.

***Importancia:*** La palabra reservada !important, sobreescribe cualquier cascada.

***Especificidad:*** Más específico es el selector, entonces ese es el estilo que se aplica.

Lo que tiene clases es más específico que lo que no.

Si tiene la misma cantidad de clases, gana el que tiene más elementos para cumplir.

***Orden en el código fuente:*** Si dos declaraciones afectan al mismo elemento, tienen la misma importancia y la misma especificidad, la selección final es el orden de las fuentes.

La declaración que se ve después en la hoja estilo “ganara” a las anteriores.

**Pseudo clases**

Las pseudo clases se utilizan para definir un estado o comportamiento especial de un elemento.

Por ejemplo: Se puede asignar un estilo cuando se pasa con el mouse encima del texto.

Sintaxis:

selector:pseudo-clase{

propiedad: valor;

}

**Pseudo elementos**

Un pseudo elemento de CSS es usado para dar un estilo a una **parte** de un elemento. Pueden usarse combinados.

Por ejemplo: Se puede asignar un estilo a la primera letra de un elemento, o a la primera línea.

Sintaxis:

selector::pseudo-elemento{

propiedad: valor;

}

**CSS3 - Box-shadow**

Se utiliza de la siguiente manera:

box-shadow: color horizontal(px/%) vertical(px/%) difuminación(px/%)

**CSS3 - Degradado**

degradado lineal [mozila firefox]

background: -moz-linear-gradient(posicion, color1, color2);

degradado lineal [Google Chrome]

background: -webkit-linear-gradient(posicion, color1, color2);

degradado lineal [Internet Explorer]

background: -ms-lineal-gradient(posicion, color1, color2);

**CSS3 - RGBA**

La A es de Alpha, le otorga transparencia al texto su utilización es:

color: rgba(red, green, blue, alpha)

El alpha va de 0 a 1.

**CSS3 - Transform: rotate**

(especificar qué navegador) transform: rotate(Xdeg);

X cantidad de grados a rotar.