



Diseño de Interfaz Web



CSS - Introducción

¿Qué es CSS?

Es un lenguaje que permite asociar estilos a documentos de marcación como html.

Son conjuntos de **reglas**, que se asocian un **estilo a una parte o todo el documento**.

Uno de los principales **problemas de accesibilidad** radica en la **utilización de etiquetas y atributos HTML para determinar el formato visual** de una página Web.

La **utilización de estilos** se convierte entonces en la **mejor herramienta para separar exitosamente la información del formato visual**.

Hojas de Estilo en Cascada (Cascading Style Sheet - CSS)

Son "bibliotecas de estilos"

Su uso contribuye a tener documentos bien definidos y de significado completo, llamados **"DOCUMENTOS SEMÁNTICOS"**.

La forma en que CSS se asocian con documentos nos permite **reutilizarlos** para determinar el **formato de sitios enteros, facilitando el mantenimiento** del formato visual de los mismos.

Si no usamos CSS...

Si no separamos el contenido del diseño, o sea, incluimos los atributos en cada etiqueta html: debemos abrir y modificar todas y cada una de las etiquetas para realizar los cambios.

Beneficios del uso de CSS

Aumento de accesibilidad, ya que el contenido se mantiene "**limpio**" con etiquetas semánticas

Trabajo **ordenado**, disminución del trabajo de producción y mantenimiento

Utilidades que no se pueden lograr con HTML

Anidamiento

Podemos ver el mismo documento en una **infinidad de navegadores y dispositivos diferentes.**

Pueden existir **varias hojas de estilo para un mismo documento**, discriminando por soporte o hasta se usa para ordenar los estilos en sitios muy complejos.

Sintaxis en CSS

Selector o patrón de selección

Establece el tipo de vínculo de la regla con el documento html asociado

Una o más declaraciones

Una propiedad y un valor

*Múltiples declaraciones se separan por punto y coma.

Sintaxis en CSS

```
selector {  
    propiedad: valor ;  
    propiedad: valor ;  
}
```

Sintaxis en CSS

¿A qué elemento?

```
selector {  
    propiedad: valor ;  
    propiedad: valor ;  
}
```

¿Qué propiedad
deseo cambiar?

¿Qué valor le doy?

CSS interno

Se puede aplicar en la misma página (html), en el HEAD.

```
<style type="text/css">  
  p {  
    color: red;  
  }  
</style>
```

CSS externo - Virtudes

Una hoja sola controla todo

Se pueden hacer todos los cambios juntos, optimiza tiempos

Optimizar para nuevas tecnologías

Se tiene por separada el contenido de la apariencia

No tenemos que repetir los mismos estilos

Tenemos la **misma base** para todas las páginas, es **reutilizable**

CSS externo

Se detalla en un archivo externo (archivo .css) y se llama desde el HEAD del html.

```
<link href="/css/archivo.css" rel="stylesheet" type="text/css" >
```

```
<style type="text/css">  
@import url("/css/archivo.css");  
</style>
```

1

2

CSS externo

1

```
<link href="/css/archivo.css" rel="stylesheet" type="text/css" />
```

href - Url del vínculo, puede ser absoluto o relativo.

rel - Tipo de vínculo que se hace con el archivo externo.

type - tipo de datos externos, text/css, textos y reglas css.



Tipos de selectores

Tipos de selectores simples

Selector universal ("para todo")

```
* {  
  color: red;  
  /*comentario en css*/  
}
```

Tipos de selectores simples

Selector de tipo (elemento predefinido -etiquetas-)

```
p {  
  color: red;  
}  
h1 {  
  color: blue;  
}
```

Tipos de selectores simples

selectores de clase

(cualquier elemento particularmente denominado, pueden ser varios)

```
.nombreClase {  
  color: red;  
}
```

Tipos de selectores simples

selectores de ID

(un elemento particularmente denominado no se puede usar más de una vez por página)

```
#nombreID {  
  color: blue;  
}
```

Tipos de selectores simples

selectores de descendencia

(Cadenas de dos o más selectores separados por espacio blanco, que funcionan en cascada, no tienen porque ser "hijos directos")

```
p a{  
  color: blue;  
}
```

Tipos de selectores simples

Los diferentes selectores pueden combinarse y agruparse.

- **combinados**: anidación, descendencia
(separados por espacio en blanco)

≠

- **agrupados**: p, a, span, em
(separados por coma)

Tipos de selectores simples

selectores de atributo

(permite seleccionar elementos HTML en función de sus atributos y/o valores de dichos atributos)

```
a[class] {  
  color: red;  
}
```

(todos los enlaces que tengan asociados una clase, cualquiera sea, van a tomar color rojo y/o valores de dichos atributos)

Tipos de selectores simples

selectores de atributo

(permite seleccionar elementos HTML en función de sus atributos y/o valores de dichos atributos)

```
a[href="http://www.google.com"] {  
color: green;  
}
```

Tipos de selectores compuestos

Algunos son selectores dinámicos ya que exigen una acción del usuario para su cumplimiento

Pseudoclasas - refiere a estados

* es importante conservar el orden

:link links que aun no han sido visitados

:visited links previamente visitados

:hover cuando se pasa por encima sin accionarlo

:active mientras está siendo accionado

:focus cuando el enlace recibe el foco (tabulador)

Tipos de selectores compuestos

Pseudoclases

`:first-child`

Primer elemento especificado dentro de otro.

"Primer hijo de su padre"

```
p:first-child { font-size:14px; }
```

Todos aquellos casos que "p" sea el primer hijo del elemento que lo contiene.

Tipos de selectores compuestos

Otros selectores refieren a elementos generados por la estructura HTML pero no están etiquetados particularmente, sino que son abstracciones que nos permiten referenciar contenido del documento HTML... **Pseudoelementos**

Tipos de selectores compuestos

Pseudoelementos

`::first-line`

Primera línea de texto del elemento de bloque para el cual es especificado.

```
p::first-line { font-size:14px; }
```

Primer línea del párrafo.

Tipos de selectores compuestos

Pseudoelementos

`::first-letter`

Primer letra de un elemento de bloque.

```
p::first-letter { font-size:14px; }
```

Primer letra del párrafo.

Tipos de selectores compuestos

Pseudoelementos

::before

Inserta dinámicamente contenido al principio del un elemento.

```
a::before{ content:"Nuevo! " }
```

::after

Inserta dinámicamente contenido al final de un elemento.

```
a::after{ content:"*" }
```

Herencia - "estilos en Cascada"

Cuanto más específica sea la regla, más poder tiene. Si hay dos normas iguales, prevalece la última declarada.

!important - Algunos navegadores hacen caso a la importancia e interpretan dicha regla como definitiva.

Prototipo de Hoja CSS

```
/*  
versión: 1.0  
autor: Guzmán  
email: info@miSitio.com  
sitio: http://www.miSitio.com/  
*/
```

Prototipo de Hoja CSS

```
/*GENERAL*/  
/* Quita rellenos y márgenes */  
* { margin: 0; padding: 0; }  
/* Clase para eliminar flotaciones */  
.clear { clear:both; }  
/* Sacamos borde alrededor de imágenes enla-  
ce */  
img { border: 0; }
```

Prototipo de Hoja CSS

```
// * GENERAL */
```

```
/* TIPOGRAFÍA */
```

```
/* ENCABEZADOS */
```

```
/* ENLACES */
```

```
/* PUBLICIDAD */
```

```
/* NAVEGACIÓN PRINCIPAL */
```

```
/* NAVEGACIÓN SECUNDARIA */
```

```
/* CONTENIDO PRINCIPAL */
```

```
/* CONTENIDO SECUNDARIO */
```

```
/* PIE */
```

```
/* FORMULARIOS */
```

```
/* TABLAS */
```

```
/* PARTICULARES */
```