

PROGRAMACIÓN MULTIPLATAFORMA

CSS3



CSS (I)

2

- La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos con la misma apariencia en diferentes navegadores.
- El organismo W3C propuso la creación de un lenguaje de hojas de estilos específico.
- Esa propuesta cristalizó en lo que hoy se conoce como las hojas de estilo CSS.



CSS (II)

3

- Las hojas de estilo en cascada o (Cascading Style Sheets, o sus siglas CSS) hacen referencia a un lenguaje usado para describir el aspecto y formato de un documento escrito en HTML.
- Los navegadores modernos soportan CSS y en buena medida CSS3.
- Internet Explorer desde la versión 8.



Características y ventajas (I)

4

- Permite la creación de documentos web visualmente más atractivos y mejor estructurados.
- Si utilizamos sólo atributos HTML estaremos muy limitados.
- Permite separar y distribuir mucho el trabajo y hacerlo modular, delegando el aspecto de las páginas web en una única tecnología y lenguaje.



Características y ventajas (II)

5

- Una hoja de estilos se puede reutilizar en cualquier web que queramos.
- Los cambios son mucho más fáciles de hacer porque el trabajo está centralizado y localizado.
- Si, por ejemplo, queremos cambiar de color a todas las cabeceras de una web, basta con ir al selector de cabeceras pertinente y cambiar el valor del color.



Novedades CSS3

6

- CSS3 es un estándar todavía en desarrollo.
- No se trata sólo de una especificación del lenguaje, se ha dividido en varios módulos de desarrollo cada uno con su propio equipo de desarrolladores y su propio calendario.
- Novedades
 - ▣ Usar efectos visuales sin imágenes (por ejemplo esquinas redondeadas).
 - ▣ Sombras.
 - ▣ Fondos semitransparentes, gradientes.
 - ▣ Transformación de cajas con efectos 2D y 3D mediante transformaciones.
 - ▣ Animaciones.
 - ▣ Posibilidad de adaptar trabajo a múltiples dispositivos.



Implementación de estilos (I)

7

- Las distintas formas que tenemos para incluir estilos CSS en páginas HTML son las siguientes:
 - ▣ **Estilos en línea:** Incluir el código CSS dentro de los elementos HTML (en las propias etiquetas) mediante el atributo **style**.
 - ▣ No se aconseja porque esta estrategia resulta en la dispersión de estilos.
 - ▣ Tendríamos una web muy poco mantenible.



Implementación de estilos (II)

8

- **Incrustados:** Incluir el código CSS en la sección *head* del documento HTML.
- **Enlazados:** Incluimos el código CSS en un archivo externo. Se usa la etiqueta **<link>** con los atributos:
 - **rel:** Indica el tipo de relación entre el archivo enlazado y el documento HTML. El valor para CSS es *stylesheet*.
 - **type:** Indica el tipo de recurso. Para CSS *text/css*.
 - **href:** URL del archivo.
 - **media:** Opcional. Es para indicar el tipo de medio para el que se orientan los estilos. Para pantalla, *screen* y para impresora *print*.
- **Importados:** Los estilos importados son estilos que residen en un archivo de estilos independiente que son llamados a su vez por otro archivo de estilos.



Método (I)

9

- CSS es un lenguaje que incorpora estilos de aspecto a elementos HTML.
- Pensemos en lo que hacemos en un programa de tratamiento de textos, concretamente cuando queremos cambiar de formato a una porción de texto determinada.
- Cuando queremos poner una palabra en negrita, lo primero que hacemos es seleccionar la palabra para luego acudir a la opción adecuada con el fin de aplicar el formato.



Método (II)

10

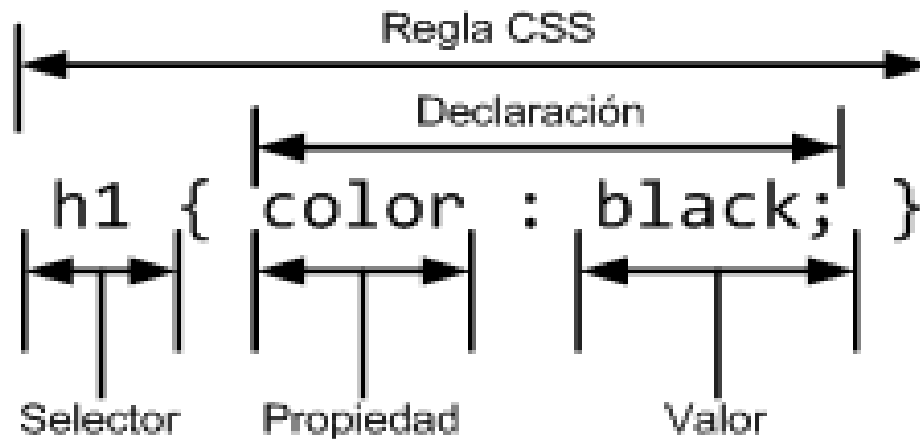
- En CSS ocurre algo muy parecido.
- Disponemos de una serie de contenidos HTML y sobre algunos de ellos queremos aplicar un estilo.
- Para especificar lo que queremos hacer, primero debemos seleccionar el o los elementos donde queremos aplicar el estilo, después especificar la propiedad a cambiar y luego darle el valor a esa propiedad.



Sintaxis básica (II)

11

- **Regla CSS:** Los estilos que se aplican en el área de influencia del selector.
- **Selector:** Donde afecta la regla CSS.
- **Propiedades y valores:** Son los estilos propiamente dichos



Selectores CSS clásicos (I)

12

- *: Selector universal. Aplica estilo a todo.
No se usa mucho.

```
* {  
  margin: 0;  
}
```



Selectores CSS clásicos (II)

13

- **Selector de tipo o etiqueta**
- Selecciona todos los elementos HTML cuya etiqueta coincida con la expuesta en la regla. El siguiente ejemplo dota de color a los párrafos de un documento y a sus cabeceras h1.

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

```
p {  
    color: black;  
}
```



Selectores CSS clásicos (III)

14

- Se pueden encadenar varios elementos separándolos por comas si queremos aplicarles el mismo estilo.
- El siguiente ejemplo aplica el mismo color a todas las cabeceras del documento.

```
h1, h2, h3, h4, h5, h6 {  
    color: blue;  
}
```



Selectores CSS clásicos (IV)

15

- ❑ **Selector descendente**
- ❑ Selecciona elementos que están dentro de otros elementos.
- ❑ El siguiente ejemplo aplica el estilo negrita a todas las etiquetas span que estén dentro de los párrafos del documento.

```
p span {  
    font-weight: bold;  
}
```



Selectores CSS clásicos (V)

16

❑ Selector de clase

- ❑ Los selectores de clase permiten hacer un filtrado en los elementos donde aplicar el estilo y para ello se sirven del atributo HTML **class**.

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

```
</p>
```

```
<span class="importante">este texto es  
importante</span>.....<span>este no</span>.....<span  
class="importante">este sí</span>
```

```
</p>
```



Selectores CSS clásicos (VI)

17

- Los selectores de clase son imprescindibles para diseñar páginas web complejas ya que permiten disponer de una precisión total al seleccionar los elementos.
- Además, estos selectores permiten combinarse con los selectores de tipo para que la aplicación del estilo sea todavía más específica.



Selectores CSS clásicos (VII)

18

- La siguiente regla aplicaría el estilo a todos los párrafos con el atributo class con el valor “importante”.

```
p.importante { ... }
```

- La siguiente aplicaría el estilo a todos los elementos con el atributo class con el valor “importante” que estén dentro de cualquier elemento de tipo párrafo.

```
p .importante { ... }
```



Selectores CSS clásicos (VIII)

19

- La siguiente regla aplicaría el estilo a todos los párrafos de la página y a todos los elementos con el atributo class con el valor “importante” de la página.
- `p, .importante { ... }`



Selectores CSS clásicos (IX)

20

- ❑ **Selector de ID**
- ❑ Habrá veces en las que necesitemos aplicar estilos CSS a un único elemento de la página.
- ❑ Podríamos usar un selector de clase para hacer esto pero hay un tipo de selector más eficaz cuando queremos hacer esto. Es el selector de id.
- ❑ El selector de id permite capturar un elemento de la página mediante el valor de su atributo **id**.
- ❑ El carácter utilizado es **#**.



Selectores CSS clásicos (X)

21

```
#elmasimportante { color: red; }
```

...

```
</p>
```

```
<span id="elmasimportante">este  
texto es el más importante</span>
```

```
</p>
```



Selectores CSS clásicos (XI)

22

- ❑ **Selector de hijos**
- ❑ Se usa para aplicar un estilo a los hijos directos de un elemento. El carácter que establece la relación padre hijo es el símbolo `>`.
- ❑ El siguiente ejemplo ilustra su funcionamiento:



Selectores CSS clásicos (XII)

23

```
p > a {color: red;}
```

```
<p><a href="#">Link1</a></p>
```

```
<p><span><a href="#">Link2</a></span></p>
```

- El estilo pondría de color rojo los enlaces hijos directos de todos los párrafos del documento.
- Al primer enlace le aplicaría el estilo pero al segundo no porque es hijo directo de una etiqueta span y no de la etiqueta de párrafo.



Selectores CSS clásicos (XIII)

24

□ Selector de atributos

- Permiten identificar elementos HTML según atributos que tengan declarados, pudiendo incluso apuntar a un valor concreto de un atributo. A continuación se ven dos ejemplos:

```
img[alt]{width: 100px;}
```

- Este ejemplo aplicaría un ancho de 100 píxeles a aquellas imágenes que tengan un atributo alt declarado.

```
img[alt="foto"]{width: 100px;}
```

- Este ejemplo aplicaría un ancho de 100 píxeles a las imágenes cuyo atributo alt declarado con valor sea *foto*.



Jerarquía de las reglas CSS (I)

25

- Además de las declaraciones individuales de cada elemento, en el aspecto final de un elemento intervienen dos mecanismos:
- **La herencia:** un elemento puede heredar alguna de las propiedades de su padre, como ocurre con las propiedades *color* o *font*.



Herencia (I)

26

- La herencia de estilos en CSS implica que cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad.



Herencia (II)

27

- El orden de herencia sería así:
 - ▣ Si un valor está definido en una regla se utiliza ese valor.
 - ▣ Si no lo está y la propiedad puede ser heredada se usa el valor del padre del elemento.
 - ▣ De otro modo, se utilizará el valor por defecto de la propiedad.



Jerarquía de las reglas CSS (II)

28

- ❑ **La cascada:** cuando muchas propiedades iguales con valores diferentes afectan a un mismo elemento, los navegadores establecen una serie de factores para interpretar la que prevalece: importancia, especificidad y orden.
- ❑ Cuando no se han definido estilos para algún elemento, cada navegador aplicará su estilo por defecto.



Cascada (II)

29

- Se clasifican por **importancia**. Cuanta mayor importancia, mayor será su impacto. Si dos reglas se encuentran en igual importancia, se clasificarán por **especificidad**.
- Si dos reglas tienen igualdad de especificidad, se comprueba el **orden de aparición** en la hoja de estilos: las últimas reglas declaradas ganan en prioridad.



Cascada (III). Especificad (I)

30

- La especificidad va en función de cuánto de específico sea el selector de una regla. Su grado de importancia es el siguiente:
 - ▣ Los más importantes son los **estilos en línea** definidos mediante la etiqueta `<style>`.
 - ▣ Seguidamente van los **selectores de id.**



Cascada (III). Especificad (II)

31

- Seguidamente van los **selectores de clase**.
- Seguidamente van los **selectores de elemento o pseudoelemento**.

