

UNIDAD 3:

USOS DE ESTILOS

(Cascading Style Sheets - CSS)

(Hojas de estilo en cascada)

Diseño de Interfaces Web

2º Desarrollo de Aplicaciones Web

José Reyes Gil Rubio

Tabla de contenido

INTRODUCCIÓN	3
¿Qué es un estilo?	3
Reglas CSS.....	3
¿Qué es una hoja de estilos?.....	4
CREAR Y VINCULAR HOJAS DE ESTILO	5
Estilos incrustados o internos	5
Hojas de estilos externas.....	5
@import	6
Estilos en línea (estilo inline).....	7
SELECTORES.....	8
Selectores de etiqueta	8
Selectores de clase	8
Restricción de alcance.....	9
Varias clases	9
Selectores de ID.....	10
Selectores descendientes.....	11
Selectores hermanos.....	12
Pseudoclasas o pseudoelementos	12
¿Qué sucede si definimos un estilo para un párrafo en una página externa, en la propia cabecera de la página y como un estilo interno mediante style?	13
RESET CSS	13
Crear nuestras propias reglas CSS de reseteo.....	14
Importar una hoja de estilos Reset	14
HERENCIA DE ESTILOS	15
Inacción de la herencia.....	16
LA CASCADA DE ESTILOS	17
¿Qué es?	17
¿Cuándo hay conflicto de estilos?	17
Cuando entra en juego la herencia	17
Cuando se aplican varios estilos al mismo elemento.....	18

INTRODUCCIÓN

Su principal ventaja es que permiten separar en el desarrollo de un sitio web el diseño de los contenidos.

Tiene como efecto inmediato un desarrollo y mantenimiento más eficiente de sitios web.

Permite la adaptación de los sitios web a los dispositivos en los que serán visualizados.

Es un lenguaje de maquetado web independiente compatible con diferentes lenguajes.

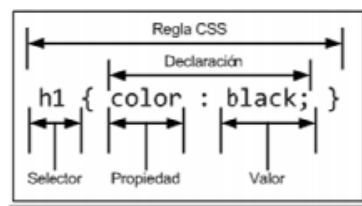
¿Qué es un estilo?

Un estilo es una regla que le indica al navegador como aplicar formato a algún elemento de nuestra web.

Reglas CSS

Una regla CSS está compuesta por:

- **Selector:** Indica los elementos HTML a los que se aplica la regla.
- **Declaración:** Puede estar compuesta por una o más propiedades.
 - **Propiedad:** permite modificar el aspecto de una característica del elemento.
 - **Valor:** indica el nuevo valor de la característica modificada en el elemento.



selector {bloque}

Zona a modificar { formato a aplicar };

En CSS sólo existen comentarios en bloque. `/* ...comentario... */`

`/* Estos son selectores de elementos básicos*/`

```
selector {  
    propiedad1:valor;  
    propiedad2:valor;  
    propiedad3:valor;  
}
```

```
selector {  
    propiedad1:valor;  
    /*  
    propiedad2:valor;  
    propiedad2:valor;  
    */  
}
```

}

Nota: Cada navegador cumple una serie de reglas CSS por defecto y no son las mismas en todos los navegadores.

```
<style>
  h1 {
    text-align: center;
    color: blue;
  }
  p {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    color: #ff0000;
    font-size: 18px;
  }
</style>
```

Hemos creado dos reglas o estilos:

- Para cabeceras h1: centradas y color azul.
- Para párrafos: Tipo de fuente Tahoma, color rojo y tamaño 18 píxeles.

Con el uso de estilos simplificamos muchísimo el código, ya que modificamos la apariencia de todos los elementos que coincidan con el selector, indicando una única vez las instrucciones que modifican su apariencia.

¿Qué es una hoja de estilos?

Es un conjunto de reglas CSS que cambiarán la apariencia de nuestra web.

CREAR Y VINCULAR HOJAS DE ESTILO

Estilos incrustados o internos

Mediante una hoja de estilos interna. Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta <style> de HTML y solamente se pueden incluir en la cabecera del documento (sólo dentro de la etiqueta <head> y por convenio, suele definirse justo antes de la etiqueta de cierre /head).

Si la web también tiene código JavaScript, primero se suele colocar la hoja de estilos CSS, luego el código JavaScript y para finalizar cerramos la etiqueta /head.

Características:

- Dan formato a una única web dentro de la que se han definido.
- La descarga es más lenta
- La hoja de estilos se incluye dentro de la etiqueta <head>.
- **Ejemplo:** Mostrar dos títulos con texto de color rojo sobre fondo amarillo.

```
<html>
<head>
  <title>Estilos</title>
  <style>
    h1 {color:#ff0000;background-color:#ffff00}
  </style>
</head>
<body>
  <h1>Este mensaje es de color rojo sobre fondo Amarillo</h1>
</body>
</html>
```

Hojas de estilos externas

Todos los estilos CSS se pueden incluir en un archivo de tipo CSS que los documentos HTML enlazan mediante la etiqueta <link>.

Se pueden crear todos los archivos CSS que sean necesarios y cada documento HTML puede enlazar tantos archivos CSS como necesite.

- Dan formato a todas las páginas que componen el sitio web.
- Agilizan la descarga de las páginas del sitio.
- Muchísimo más eficaz ya que escribo el código CSS una sola vez para todo el sitio y no por cada página.
- El mantenimiento es mucho más sencillo, ya que las modificaciones las hago en un solo archivo. Por este mismo motivo, es una forma de aplicar formato menos sujeta a errores.
- **Ejemplo:** Confeccionar una página HTML que incorpore una hoja de estilo desde un archivo externo. Definir reglas de estilos para las etiquetas body, h1 y p.

index.html

```
<html>
<head>
  <title>Estilos</title>
  <link rel="stylesheet" href="estilos.css">
</head>
<body>
  <h1>Definición de hojas de estilo en un archivo externo</h1>
  <p>
    Ahora veremos que la metodología más empleada es la definición de una
    hoja de estilo en un archivo separado que deberá tener la extensión css.
  </p>
</body>
</html>
```

estilos.css

```
body {
  background-color: yellow;
}
h1 {
  color: red;
}
p {
  color: blue;
}
```

Definición de hojas de estilo en un archivo externo

Ahora veremos que la metodología más empleada es la definición de una hoja de estilo en un archivo separado que deberá tener la extensión css.

En un mismo documento html podemos hacer referencia a más de una hoja de estilos externa. Se aplicarán en el orden que las coloquemos dentro del documento.

```
<link rel="stylesheet" href="css/reset.css">
<link rel="stylesheet" href="css/estilos.css">
```

@import

Usado para vincular una hoja de estilos externa con otra hoja de estilos externa.

Debe ocupar siempre el primer lugar en una hoja de estilos incrustada o al principio de una hoja de estilos externa.

En HTML se incluiría en una etiqueta <style>

```
<style>  
    @import url(styles.css);  
    ...aquí puede haber otros enunciados o estilos CSS para importarlos...  
</style>
```

En CSS:

```
@import url(styles.css);  
...aquí puede haber otros enunciados o estilos CSS para importarlos...
```

Estilos en línea (estilo inline)

No es recomendable, consiste en incluir propiedades CSS directamente en las etiquetas HTML utilizando el atributo style.

NOTA: Todo lo que veremos en este apartado se podrá realizar usando otro método y sin contar con los estilos inline (lo veremos más adelante).

- **Ventajas:**
 - Posibilidad de cambiar un único elemento HTML de la página web.
- **Desventajas:**
 - No ahorra tiempo a la hora de crear las páginas web ni tampoco ancho de banda como las hojas de estilo internas y externas.
- **¿Cuándo utilizarlo?**
 - Cuando queremos emplear algún formato especial y puntual.
 - Email o textos especiales.
- **Ejemplo:** Definir un estilo interno a la etiqueta H1 que fije el color de texto en rojo y el fondo en amarillo.

```
<html>  
<head></head>  
<body>  
    <h1 style="color:#ff0000; background-color:#ffff00">  
        Este mensaje es de color rojo sobre fondo amarillo.  
    </h1>  
</body>  
</html>
```

Este mensaje es de color rojo sobre fondo amarillo.

SELECTORES

selector {bloque de declaraciones}

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

Selectores de etiqueta

Se especifica la etiqueta HTML a la que vamos a aplicar estilos.

```
etiqueta {  
    propiedad:valor;  
    propiedad:valor;  
    ...  
}
```

```
p { color:#F00;  
    font-style: italic;  
    font-weight:bold;  
}  
h2 {color:#00F;  
    font-size:1.2em;  
}
```

Selectores de clase

Se utilizan para dar estilo a varios elementos. Te permite establecer un formato de estilo al elemento que quieras dentro de tu página web. No es tan rígido como el selector de etiqueta, sino que nos permite ir aplicando estilos diferentes a diferentes elementos que tenemos en una página web.

Se define la clase dentro de la etiqueta html con el argumento “class”

<h1 class="miclase">...

Se define el estilo en la hoja de estilos con un punto “.” delante del nombre y no debe de empezar por número o símbolos extraños y es “case sensitive”:

```
.miclase {  
    Bloque;  
}
```

```
.miestilo {  
    color:#0f0;  
    font-size:16px;  
}
```


Este estilo de clase “miestilo” se puede aplicar a cualquier elemento de nuestra web, simplemente asignándole a dicho elemento el atributo `class=“miestilo”`.

Ej. `<small class=“miestilo”>...`

Restricción de alcance

.miestilo = *.miestilo

Para restringir el alcance de estilo de clase basta con sustituir el asterisco por la etiqueta a la que queremos limitar el alcance.

Por ej. **p.miestilo**

Aplica estilos solamente al párrafo cuyo atributo `class` sea igual a “miestilo”

En resumen, NO CONFUNDIR:

`p.avisos {...}` → Todos los elementos `p` con atributo `class=“avisos”`

`p .avisos {...}` → Todos los elementos con atributo `class=“avisos”` que estén dentro de cualquier elemento `p`.

`p, .avisos {...}` → Todos los elementos `p` y todos los elementos con atributo `class=“avisos”`.

Varias clases

Para evitar repetición de código podemos asignar varias clases a un mismo elemento. Por ejemplo:

```
.cabecera1 {
    font-size: 2.5em;
    margin-left: 25px;
    border-bottom: 2px solid rgb(241, 40, 164);
    color:green;
}

.cabecera2 {
    font-size: 2.5em;
    margin-left: 25px;
    border-bottom: 2px solid rgb(241, 40, 164);
    color: blue;
}

.cabecera3 {
    font-size: 2.5em;
    margin-left: 25px;
    border-bottom: 2px solid rgb(241, 40, 164);
    color:red;
}
```

```
<h1 class="cabecera1">Título 1a</h1>  
<h2 class="cabecera2">Título 1b</h2>  
<h3 class="cabecera3">Título 1c</h3>
```

O lo que es lo mismo:

```
.cabecera{  
    font-size: 2.5em;  
    margin-left: 25px;  
    border-bottom: 2px solid rgb(241, 40, 164);  
}  
.cabecera1 {  
    color:green;  
}  
  
.cabecera2 {  
    color: blue;  
}  
  
.cabecera3 {  
    color:red;  
}
```

```
<h1 class="cabecera cabecera1">Título 1a</h1>  
<h2 class="cabecera cabecera2">Título 1b</h2>  
<h3 class="cabecera cabecera3">Título 1c</h3>
```

Selectores de ID

Se utiliza para especificar el formato de un elemento en concreto de la página web. Sólo se puede aplicar un estilo id por elemento.

Se define la clase dentro de la etiqueta html con el atributo “id”

<h1 id="miclase">...

Se define el estilo en la hoja de estilos con una almohadilla “#” delante del nombre y no debe de empezar por número o símbolos extraños y es “case sensitive”:

```
#miclase {  
    Bloque;  
}
```

```
#tabla{  
    background-color: blue;  
    border-color: red;  
}
```

En el HTML:

Ej. <table id="tabla">

Selector de grupo

Tiene como finalidad aplicar un conjunto de estilos a varias etiquetas diferentes.

```
selector1, selector 2, ..., selectorn {  
    bloque;  
}
```

```
h1,h2,small:{  
    color:blue;  
}
```

Selector universal

Aplica las características de formato a todos los elementos de la página web.

```
* {  
    color:blue;  
}
```

Selectores descendientes

Tenemos que estar muy pendientes de la jerarquía de etiquetas.

Por ejemplo:

```
li a {  
    color:red;  
}
```

Con esta regla nos referimos a todos los elementos con etiqueta **a** que se encuentren dentro de un elemento **li**.

- li → etiqueta padre
- a → etiqueta hija

Este estilo no afectará a las etiquetas **a** que no se encuentren dentro del elemento **li**.

Otro ejemplo:

```
li a strong {  
    color:green;  
}
```

En estilos descendientes se pueden combinar estilos de etiqueta con estilos de clase o de ID.

Por ejemplo:

```
strong.destacado{  
    text-decoration: underline;  
}
```

NOTA: El estilo de clase .destacado desciende de la etiqueta strong.

Si a una palabra de un párrafo no le queremos dar ningún formato en especial, solamente el definido por un estilo de clase solemos usar la **etiqueta ** por ejemplo:

```
<strong><span class="destacado">voluptatem</span></strong>
```

Podemos usar otra opción eliminando el espacio en blanco entre la etiqueta y la clase:

```
strong.destacado{  
    text-decoration: underline;  
}
```

NOTA: Se aplica a un estilo de clase que se llama destacado y que esté aplicado como atributo en la etiqueta strong.

```
<strong class="destacado">natus</strong>
```

Selectores hermanos

Los selectores hermanos consiguen modificar el estilo de un elemento que esté inmediatamente después de otro elemento concreto. Por ejemplo: Modifica los párrafos que estén inmediatamente después de una etiqueta H2.

```
h2+p {  
    color: #444;  
    font-size: 1.2em;  
    line-height: 120%;  
}
```

Pseudoclases o pseudoelementos

Por ejemplo en vínculos...

Los estados en los que se puede encontrar un vínculo son:

- Link: estilo que tiene el vínculo en reposo
- Visited: estilo que tiene el vínculo cuando ha sido visitado con anterioridad
- Hover: estilo que tienen los vínculos cuando se pasa el ratón por encima
- Active: estilo cuando se está haciendo clic.

Para darle estilos a unos estados o a otros haremos uso de pseudoclases o pseudoelementos que consiste en agregar el estado correspondiente al selector de la siguiente forma:
a:hover{...}

```
a:link{
    color:#f00;
}
a:visited{
    color:#0f0;
}
a:hover{
    color:#ff0;
}
a:active{
    color:#f0f;
}
```

NOTA: Si vamos a modificar los cuatro estados a la vez este sería el orden por el que deberíamos definir los estilos (link, visited, hover, active = LOVE HATE)

Las pseudoclasas no son sólo para vínculos, también pueden ser empleadas por ejemplo en párrafos.

```
p:hover{
    background-color:grey;
}
```

¿Qué sucede si definimos un estilo para un párrafo en una página externa, en la propia cabecera de la página y como un estilo interno mediante style?

- Primero se aplican las hojas externas
- Luego se aplica el estilo definido en la cabecera
- Si en un elemento concreto se ha definido una propiedad específica, se aplica esta última.
 - Podemos hacer saltar esta cadena de prioridades usando el modificador **!important**.
 - Ej. p {margin: 6px !important;}

RESET CSS

La finalidad del Reset CSS es que el punto de partida sea el mismo para todos los navegadores, ya que existen pequeñas diferencias de tamaños de letra, márgenes, distancias a bordes, etc. entre un navegador y otro.

Para resetear CSS podemos:

Crear nuestras propias reglas CSS de reseteo

```
/*Aquí comienza nuestro propio RESET CSS*/
html, body, h1, h2, h3, h4, h5, h6, p, ol, ul, li, pre, code, address, form, fieldset
{
    padding:0;
    margin:0;
    font-size:100%;
    font-weight:normal;
}

ol{
    margin-left: 1.4em;
    list-style: decimal;
}

ul{
    margin-left: 1.4em;
    list-style: circle;
}

img {
    border:0;
}

/*Aquí termina nuestro propio RESET CSS*/
```

Importar una hoja de estilos Reset

```
@import url(reset.css);
```

HERENCIA DE ESTILOS

Nos permite ahorrar tiempo y consiste en que las propiedades CSS pasan de las etiquetas padre a las etiquetas que puedan estar anidadas dentro de ellas.

Por ejemplo: una etiqueta <p> dentro de <body> o con más generaciones, por ejemplo, una etiqueta dentro de un <p> que a su vez está dentro de un <body>.

Es decir, si hemos establecido unas reglas de estilo para el <body>, esas reglas pasan por herencia a la etiqueta <p> y al que hay dentro del <p>. Aunque hay que tener en cuenta que NO todas las propiedades se heredan.

```
p{  
  color: #555;  
}
```

Si tenemos esta regla, el color de letra de las etiquetas anidadas dentro de la etiqueta <p> será también #555.

La herencia también tiene efecto, además de con los selectores de etiqueta, con los selectores de clase y con los selectores de id.

```
<body class="estilopagina">
```

```
.estilopagina {  
  font-family: 'Trebuchet MS', 'Lucida Sans Unicode', sans-serif;  
  font-size: 18px;  
  color: #ef3e67;  
  width: 90%;  
  margin: 0 auto;  
}
```

Si aplicamos esta regla a un selector de clase asignada al <body> todos los elementos incluidos en el body tomarán por herencia este aspecto CSS.

Cuando un elemento recibe por herencia un estilo y otro específico, siempre tendrá preferencia el específico.

Con las reglas anteriores, el color de letra de <p> será #555 (específico) aunque reciba por herencia del selector de clase .estilopagina el color #ef3e67.

El resto de propiedades (tamaño, familia, etc) si las recibe por herencia.

Inacción de la herencia

Esto quiere decir que hay ciertas propiedades de CSS que no se aplican a las etiquetas que hay por debajo en la jerarquía del documento web.

Por ejemplo:

Con la regla .estilopagina que definía para el body un tamaño de letra font-size: 18px; aunque existan cabeceras h1, h2,..., dentro del body, estas no heredarán esta propiedad, es decir, no tomarán ese tamaño de letra por inacción de la herencia.

Otro ejemplo:

```
p{  
  color: #555;  
  margin-left: 50px;  
  padding-left: 20px;  
  border-left: solid 12px #3bc1cd;  
}
```

Si tenemos un strong dentro de <p> este no recibirá por herencia las propiedades de bordes, márgenes, padding,...

En general, todas las propiedades CSS que hacen referencia a bordes y a colocación de elementos no se heredan. Mientras que las que hacen referencia al aspecto del texto, color, tamaño,... si se heredan.

LA CASCADA DE ESTILOS

¿Qué es?

Reglas que determinan que propiedades de estilo se aplican a un elemento concreto cuando existe sobre él un conflicto de estilos.

¿Cuándo hay conflicto de estilos?

- Cuando entra en juego la herencia
- Cuando se aplican varios estilos al mismo elemento

Cuando entra en juego la herencia

```
<style>
  body {
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    color:blue;
  }

  p{
    color:red;
  }

  strong{
    font-size: 32px;
    color:green;
  }
</style>
```

¿Qué color se le aplican a las palabras que aparecen dentro del párrafo? ¿Y a las del strong?

Sería el color azul porque, cuando dos propiedades entran en conflicto, siempre se aplica la regla de estilo más específica.

Esto explica el por qué algunas propiedades heredadas no tienen efecto sobre algunas etiquetas. Por ejemplo, si añadimos una etiqueta <a> dentro del párrafo, ésta no tomará el color rojo del párrafo, ya que la propia etiqueta <a> tiene un estilo predefinido y entre las propiedades de dicha etiqueta está que el color es azul.

Lorem ipsum **dolor**, sit amet [adipiscing](#) elit. Maiores, nihil possimus temporibus id totam accusamus! Quidem, saepe eum voluptas molestiae at recusandae neque vitae, quas, id autem cupiditate incidunt quam!

Cuando se aplican varios estilos al mismo elemento

```
<style>
  strong{
    color: red; /*rojo*/
  }

  .color_propio{
    color: blue; /*azul*/
  }

  #otro_color{
    color: green; /*verde*/
  }

</style>
</head>
<body>
  <p>Lorem ipsum <strong class="color_propio" id="otro_color" style="color:yellow;">dolor</strong>, sit ametadipisicing elit! Quidem, saepe eum voluptas molestiae at recusandae neque vitae, quas, id autem cupiditate incidunt quam!</p>
</body>
```

¿Qué color tomará el texto de la etiqueta ?

- En un principio tomará el color de la regla de estilo inline (amarillo).
- Si eliminamos la regla de estilo inline (amarillo) prevalece el color de la regla de estilo de id.
- Si eliminamos la regla de estilo de id, prevalece el color de la regla de estilo de clase.
- Y por último, tendríamos el color de la regla de estilo de etiqueta.

Podemos establecer un sistema de puntuación para resolver estos conflictos:

- Un selector de etiqueta = 1 punto
- Un selector de clase = 10 puntos
- Un selector de id = 100 puntos
- Un selector inline = 1000 puntos

