

Ministerio de Educación, Cultura y Deporte.

# HTLM5 en la educación

Módulo 9: Hojas de estilo.

GOBIERNO  
DE ESPAÑAMINISTERIO  
DE EDUCACIÓN, CULTURA  
Y DEPORTESECRETARÍA DE ESTADO  
DE EDUCACIÓN, FORMACIÓN PROFESIONAL  
Y UNIVERSIDADESINSTITUTO NACIONAL  
DE TECNOLOGÍAS EDUCATIVAS  
Y DE FORMACIÓN DEL PROFESORADO

## Hojas de estilo

# La estructura y el estilo



Atrás quedaron los años en que diseñar una página web consistía en juntar en un caldero todo tipo de etiquetas, mezclando el diseño de estructura con el formato. En la actualidad ambas partes están muy separadas.



## Programación

### Objetivos específicos

- Conocer los aspectos básicos del uso de estilos.
- Diseñar hojas de estilo.
- Utilizar hojas de estilo en una página web.

### Contenidos

- Hojas de estilo.
- Estilos internos y externos.
- Modificaciones básicas de estilos.

### Criterios de evaluación

- Crear una hoja de estilo interna y externa.
- Modificar los estilos creados.



## Requisitos mínimos

- Conocimientos sobre HTML.
- Conocimientos sobre navegadores web.
- Conocimientos de procedimientos en el ordenador: seleccionar, cortar y pegar.

## Recurso TIC: Hojas de estilo

En este punto ya conocemos las etiquetas fundamentales de HTML, incluso estamos familiarizados con algunas etiquetas del estándar que viene de camino, el HTML5. ¿Sabemos ya diseñar un sitio web? Sólo parcialmente. En la actualidad ya no basta con manejar HTML para realizar estas tareas; necesitamos conocer también el estándar que lo complementa, CSS, acrónimo de hojas de estilo en cascada.

La norma CSS nos facilita un conjunto de elementos que podremos aplicar a nuestras etiquetas HTML para proporcionarles la apariencia deseada. Encontraremos modificadores para aplicar todo tipo de variaciones, las mismas que podríamos obtener si llevamos un documento a un procesador de texto, e incluso muchas de las cosas que podríamos hacer con un programa de edición de imágenes.

El uso de hojas de estilo es obligatorio en las páginas web creadas con HTML5, ya que en ese estándar se abandonan todas las etiquetas y parámetros que en su día permitieron aplicar formato. Esto es, HTML5 se encarga de definir la estructura de la página y CSS el estilo y el diseño. El navegador, a su vez, tomará cada una de las partes y las mezclará, dando el resultado deseado: una página con sentido semántico gracias a HTML y con una apariencia estética apropiada mediante el estilo CSS.

Si a lo anterior sumamos algunas acciones creadas con *JavaScript*, conseguiremos aportar interactividad a la página web, creando así verdaderas aplicaciones que se ejecutan sin necesidad de instalar nada en el ordenador; pero éste es un aspecto que tocaremos parcialmente en este texto.

## CSS3

Las hojas de estilo llevan con nosotros varios años ya. La primera versión del estándar se empezó a emplear hacia el año 2001. En la actualidad la versión 3 de este conjunto de normas está aún en desarrollo, por lo que podemos encontrarnos con que algún navegador concreto no es capaz de entender alguna de las reglas que aplicamos; pero serán casos muy puntuales. La mayoría de los navegadores son capaces de ejecutar todas las propiedades principales de CSS3.

## Antes de seguir

Para poder ejecutar correctamente el trabajo que tenemos por delante, es imprescindible tener unas nociones sólidas sobre HTML5, por lo que es aconsejable haber seguido la primera parte de este material, donde explicamos el funcionamiento básico de HTML, la estructura general de una página web, cómo aplicar formato y maquetar esas páginas; además de aspectos tales como insertar recursos multimedia, trabajar con las listas y tablas o cómo publicar una página web en Internet.



### Pregunta de Elección Múltiple

Respecto a las hojas de estilo CSS:

- ☐ Se crean páginas con sentido semántico.
- ☐ Se crean páginas interactivas con sentido semántico.
- ☐ Se crean páginas con apariencia estética, con aplicaciones que se ejecutan sin necesidad de instalar nada en el ordenador.
- ☐ Ninguna de las anteriores es correcta.

## ¿Qué son las hojas de estilo?

Una hoja de estilo en cascada (CSS) es un conjunto de reglas en las que definimos el formato que va a tomar un determinado elemento de una página web.

Hasta el momento sabemos que una regla CSS se puede aplicar mediante el uso del parámetro **style**. Por ejemplo, la línea:

```
<h1 style="{color:blue}">Bienvenidos </h1>
```

Mostrará un título de color azul.

Ese método funciona correctamente, pero es difícil de mantener. En un documento con 50 títulos tendríamos que aplicar el formato a cada uno de ellos. ¿Y si después nos cansamos y los queremos poner todos verdes? Tendríamos que modificarlos todos. Por este motivo, las hojas de estilo se suelen aplicar de otra manera: como elementos dentro de la cabecera de la página web y mejor aún, en archivos externos. Así se recopilan todas las reglas en un mismo espacio, haciendo que sea más sencillo su mantenimiento y modificación.

De este modo, una regla que hiciese exactamente lo mismo quedaría así:

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

La línea anterior, situada en la cabecera (entre **<head>** y **</head>**) de la página, haría que se cambiasen de color todos los **<h1>** de nuestra página. Como se puede observar es más rápido y más sencillo. Incluso el mantenimiento de la página se vuelve trivial, ya que el cambio de color se reduciría a poner *green* (verde) donde ahora pone *blue* (azul).



### Pregunta de Elección Múltiple

Lo más cómodo a la hora de aplicar estilo CSS es:

- ☐ Como elementos en archivos externos.
- ☐ Como elementos dentro de la cabecera de la página web.
- ☐ Como elementos dentro de la cabecera de la página web, dentro de archivos externos.



### Actividad 1.

Utilizando un editor de páginas web como BlueGriffon y un documento base redefiniremos el estilo de la etiqueta **<p>** y observaremos los cambios que se producen. Para realizar la edición accedemos a la opción **Propiedades de estilo** del menú **Paneles**. En la ventana que se despliega utilizaremos las opciones de los paneles **General** y **Colores** para redefinir la etiqueta **<p>**. Antes de comenzar debemos seleccionar la opción **todos los elementos del mismo tipo** en el cuadro **Aplicar estilos a**.

Podemos localizar un texto largo en la web del *Proyecto Gutenberg* (<http://www.gutenberg.org>) donde se pueden descargar libros completos cuyos derechos de copyright ya han vencido. De ese libro extraeremos un capítulo concreto, ya que el tratamiento del libro completo puede resultar ligeramente lento.

## Clases e identificadores

Pensemos ahora en una página en la que queremos cambiar la apariencia de algunos **<h1>**, pero no de todos. ¿Cómo lo conseguiríamos? Para ello tenemos un parámetro general de HTML denominado **class** ("clase" en español, en el sentido de "tipo"). Este parámetro nos sirve para crear un conjunto de etiquetas que forman parte del mismo grupo. El objetivo es que luego podremos referirnos a esa clase de forma conjunta.

En este ejemplo tenemos tres títulos diferentes:

```
<h1>Bienvenidos</h1>
```

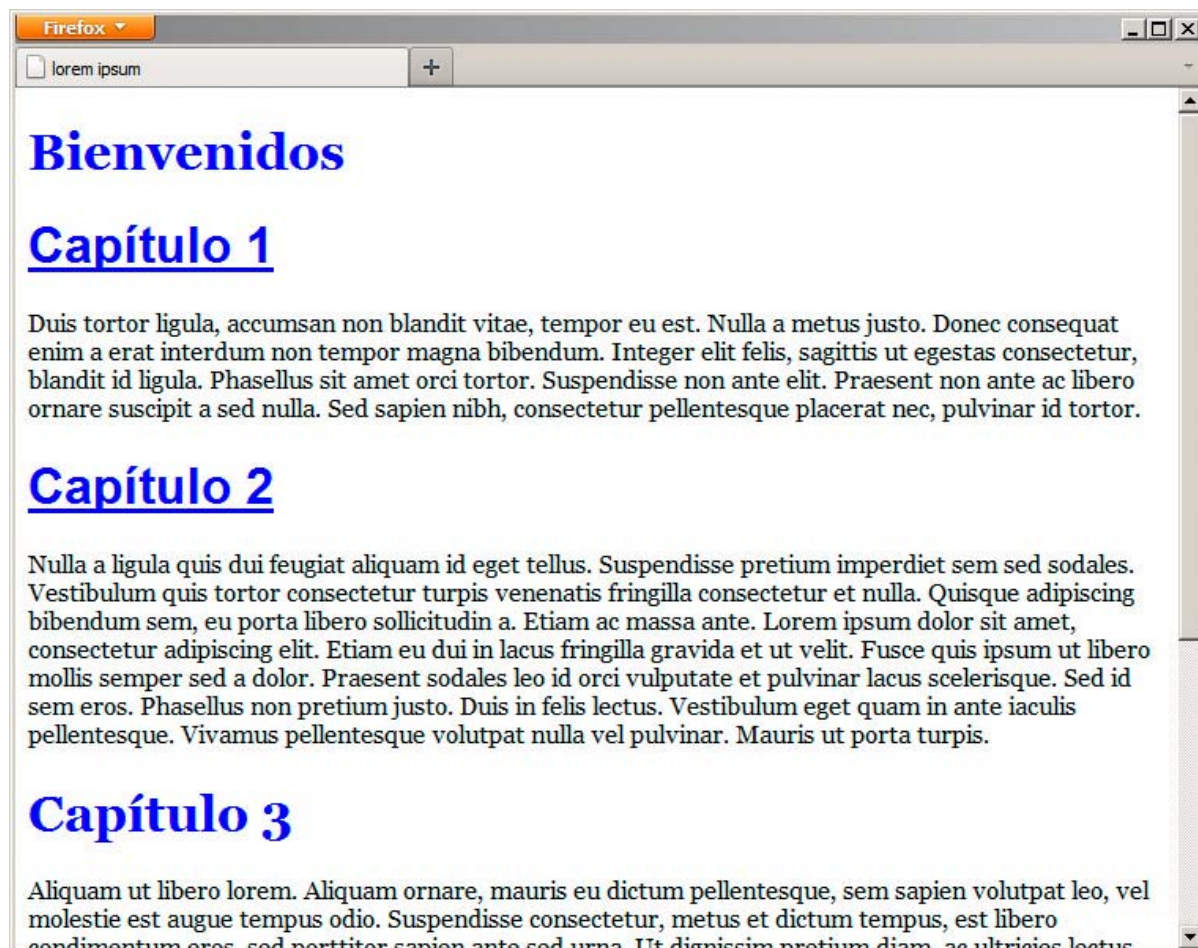
```
<h1 class="capitulos">Capítulo 1</h1>
```

```
<h1 class="capitulos">Capítulo 2</h2>
```

Dos de ellos los hemos agrupado dentro de una clase llamada "capítulos". Gracias a ello podemos modificar sólo su apariencia, diferenciándolos del resto de los títulos del documento. Para referirnos a la clase, emplearíamos el signo de punto (.).

```
.capitulos { font-family: sans-serif; text-decoration: underline;}
```

Al probarlo en un navegador, obtendríamos el resultado de la figura.



Es decir, ahora tenemos dos tipos de títulos: mientras que todos ellos serán de color azul, sólo los que forman parte de la selecta clase "capítulos" tendrán un tipo de letra concreto y estarán subrayados. Por tanto, podemos aplicar normas para todos los títulos o sólo para los de los grupos que generemos.

## Clases dependientes

Las reglas que definimos para una clase pueden ser aplicadas exclusivamente a una etiqueta. Así podríamos tener un par de reglas como éstas:

```
h1.capitulos { font-family: sans-serif; text-decoration: underline; font-size:14px}
```

```
p.capitulos { font-family: sans-serif; text-decoration: underline;}
```

De este modo podemos reutilizar el nombre de la clase y definir distintos valores, dependiendo de si se aplica a una cabecera o a un párrafo, en nuestro ejemplo.

## Identificadores

Además de las clases, podemos diferenciar un único elemento entre todos los demás de una página mediante el parámetro **id** (abreviatura de "identificador"). Con **id** asignaremos un nombre único que luego podemos emplear en nuestra sección de estilos, precediéndolo en este caso del signo #.



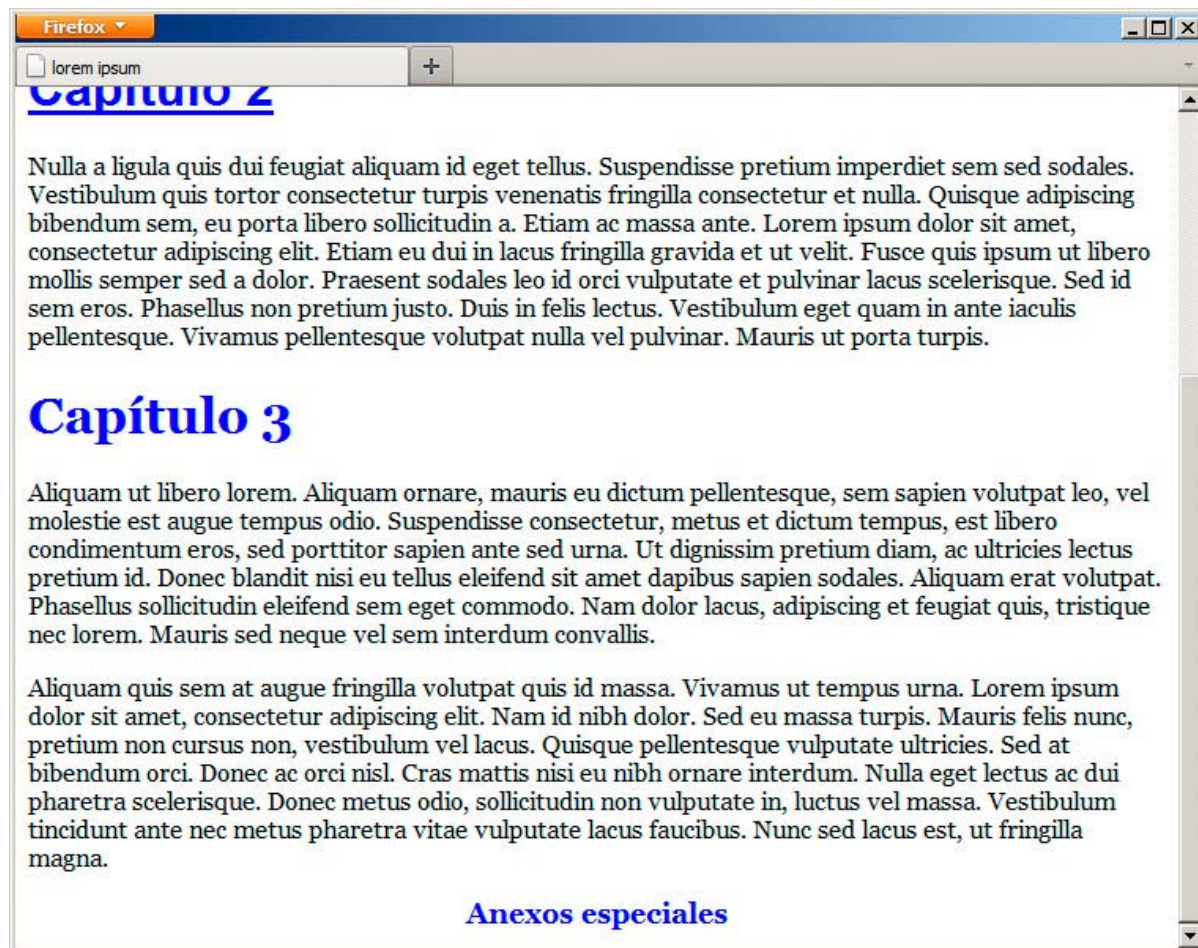
Continuando con el ejemplo anterior, supongamos que tenemos un identificador como éste:

```
<h1 id="anexos">Anexos especiales</h1>
```

Podríamos modificar exclusivamente ese anexo mediante la siguiente regla:

```
#anexos { font-size: 18px; text-align: center; }
```

El resultado se muestra en la figura.



La clave del **id** es que estamos creando un grupo específico sólo con un elemento. Este modificador tiene otras muchas utilidades, ya que al designar un elemento de forma única dentro de la página web, podremos operar con él mediante *JavaScript*, aunque eso es otra cuestión.

El sentido de utilizar identificadores, al trabajar con estilos, es mayor cuando pensamos en las hojas externas. Podemos tener un elemento con un identificador que se repite a lo largo de varias páginas. Si nuestros estilos están guardados en un archivo externo, podremos modificar todos esos identificadores con facilidad.



### Pregunta Verdadero-Falso

Las siguientes afirmaciones, ¿son verdaderas o falsas?

Mediante los identificadores podemos diferenciar un único elemento entre todos los demás de una página utilizando el parámetro **id**.

Verdadero ☐ Falso ☐

Las reglas que definimos para una clase pueden ser aplicadas exclusivamente a una etiqueta, por ello no se puede reutilizar el nombre de la clase y definir distintos valores para una cabecera o a un párrafo.

Verdadero ☐ Falso ☐



## Actividad 2

Utilizando un editor de páginas web y el documento base anterior, definiremos un par de clases diferentes para distinguir los títulos de capítulos de los títulos generales o distintos tipos de párrafos, lo que mejor se ajuste a nuestro texto. Para ello estableceremos el nombre de cada clase en el cuadro **Clase** de la barra del editor y en el panel **Propiedades de estilo** seleccionaremos la opción **todos los elementos de la clase** en el cuadro **Aplicar estilos a:**

Crearemos también un identificador en algún elemento. En este caso, indicaremos un nombre en el cuadro **Identificador** de la barra del editor, situado justo antes del cuadro **Clase**. Para aplicar estilos a un identificador, seleccionaremos la opción **este elemento mediante su ID** en el cuadro **Aplicar estilos a:** del panel **Propiedades de estilo**.

Aplicaremos estilos distintos a las clases y el identificador, modificando algunos aspectos generales del texto, como su tamaño, color o alineación.

## Identificadores dependientes

Como sucede con las clases, un identificador puede estar definido de forma general o específicamente para una etiqueta concreta. Es distinto esto:

```
#anexos { font-size: 18px; text-align: center; }
```

que esto:

```
div#anexos { font-size: 18px; text-align: center; }
```

El segundo sólo se aplicaría a un elemento que se llamase así:

```
<div id="anexos">
```

pero no a uno denominado

```
<h1 id="anexos">
```

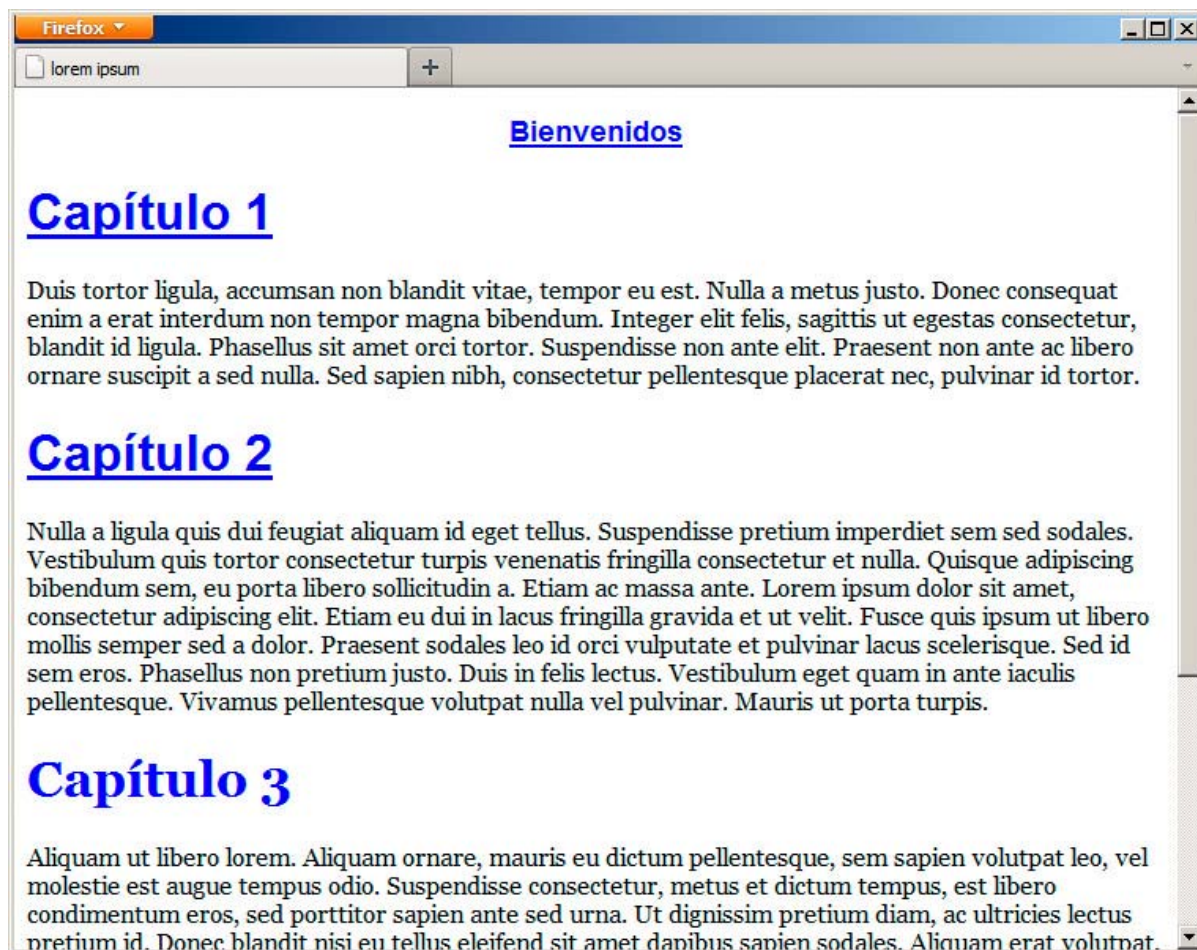
## Combinación de class e id

Se pueden combinar los parámetros **class** e **id** en un mismo elemento, con lo que conseguiremos que se le apliquen tanto las reglas definidas para la **clase** como para el **identificador**.

Este ejemplo:

```
<h1 class="capitulos" id="anexos">Bienvenidos</h1>
```

da como resultado una mezcla de todas las reglas que hemos aplicado hasta ahora, tanto las del elemento **<h1>**, como las de la **clase** y las del **identificador**. Observa la figura:



## Más selectores

Hay otros métodos para definir reglas que se aplicarán a diferentes grupos de etiquetas.

### Selector universal

El signo \* se emplea para definir a todo el conjunto de etiquetas de una página web. Nos sirve para indicar "esto se aplica a cualquier etiqueta". Por ejemplo:

```
* {font-family:san-serif; }
```

Hace que cualquier etiqueta de la página tome ese tipo de letra.

### Agrupación de selectores

También podemos emplear la coma para agrupar varias etiquetas, clases e identificadores. Un par de ejemplos:

```
p, h1, h2 {font-family: times, word-spacing: 2px;}
img, #portada, hr, .fotografia {margin: 6px;}
```

### Sucesores y antecesores

Todavía obtenemos mayor control, gracias a las opciones que nos ofrece CSS para aplicar estilos, si un elemento descende de otro o es el padre de uno. No entraremos en demasiados detalles, pero observemos la siguiente regla:

```
h2 strong {font-style: italic; }
```

Esa línea haría que se mostrase en cursiva el contenido etiquetado dentro de **<strong>**, pero si y sólo si esta etiqueta se encuentra dentro de una cabecera **<h2>**. No es necesario que lo sea directamente; puede haber otras etiquetas conteniendo la de **<strong>**.



Por ejemplo:

```
<h1>Esta línea no se vería <strong>afectada</strong></h1>  
<h2>Esta línea <strong>sí</strong> se vería <strong>afectada</strong></h2>
```

El ejemplo anterior se mostraría como en la figura:



El espacio, por tanto, indica que la aplicación se hará dependiendo de si un elemento está dentro de otro.

Además hay otros operadores que definen el tipo de relación, como el \*, el signo de > (que indica que un elemento descende directamente de otro), el de + o el signo ~.



### Pregunta Verdadero-Falso

Las siguientes afirmaciones, ¿son verdaderas o falsas?

El signo \* se emplea para definir a todo el conjunto de etiquetas de una página web.

Verdadero ☐ Falso ☐

Se puede emplear la coma para desagrupar varias etiquetas, clases e identificadores.

Verdadero ☐ Falso ☐



### Nota

El uso de estas opciones en estos primeros compases será muy limitado, por lo que no profundizaremos más en ello, por ahora.



### Actividad 3

En el documento anterior tomaremos una etiqueta que tenga nombre de **clase** y en su interior definiremos una parte como **<strong>**. Haremos lo mismo con otra etiqueta que no tenga nombre de clase. Por último, definiremos un **selector** para que se aplique formato sólo a la etiqueta que tiene nombre de clase y no a todos los **<strong>** del documento. Es decir, recurriremos al **selector de sucesores** para conseguir el efecto.

Estos cambios no los realizaremos con *BlueGriffon*, sino que definiremos los selectores directamente en el código fuente de la página, en la cabecera del documento.

## Pseudoclases

Contamos con un último tipo de selectores denominado **pseudoclases** que permiten definir propiedades para algunos elementos html que cuentan con diferentes estados. El más conocido es el caso de la etiqueta **<a>**, empleada para definir enlaces. Esta etiqueta puede tener varios estados. Estos son los principales:

- **a:link**: el estado normal de un enlace no visitado.
- **a:visited**: el estado de un enlace, una vez que ha sido visitado.
- **a:active**: cuando se está pulsando sobre el elemento.
- **a:hover**: cuando el ratón está sobre el elemento.
- **a:focus**: cuando el elemento tiene el foco del navegador. Es el que está preseleccionado.

Pues bien, si definimos algunos de esos estados, conseguiremos, por ejemplo, que un enlace cambie su apariencia al pasar el ratón sobre él (**a:hover**) o que cambie de color cuando ha sido visitado (**a:visited**). Podríamos hacerlo así:

```
a:link {font-family: sans-serif; }  
a:hover {color:red; }  
a:visited {color:gray; }
```

La figura muestra el enlace tras ser visitado:



Aún hay muchas más pseudoclases, centradas en aspectos posicionales, dependientes de si un elemento es el primero de su

tipo (**:first-of-type**, **:last-of-type**), si es el único descendiente (**:only-child**), si está vacío (**:empty**), etc. Incluso tenemos una pseudoclase para no aplicar estilo a un elemento concreto (**:not**), pero de nuevo su uso es poco frecuente en esta etapa de aprendizaje de CSS.

Veamos un ejemplo de uso con la lista siguiente:

```
<ul>

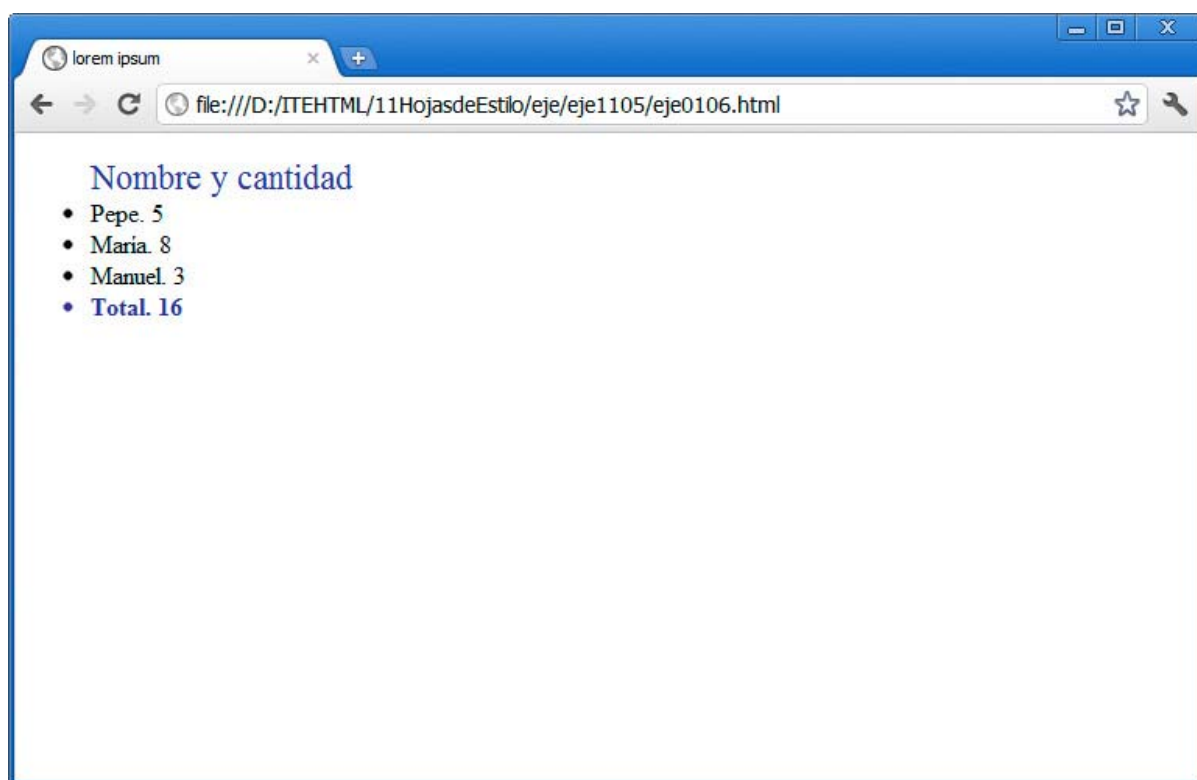
  <li>Nombre y cantidad</li>
  <li>Pepe. 5 </li>
  <li>María. 8 </li>
  <li>Manuel. 3</li>
  <li>Total. 16</li>

</ul>
```

Le aplicaremos los estilos siguientes:

```
li:first-child {font-size:22px; color:#333399; list-style-type: none }
li:last-of-type {font-size:16px; color:#333399 ; font-weight:bolder; }
```

En este caso no había diferencia entre emplear **child** o **of-type**. El resultado que obtenemos es el de la figura:



## Actividad 4

Siguiendo con nuestro ejemplo, crearemos un enlace a la página desde la que tomamos el texto original y definiremos estilos para los diferentes estados posibles del enlace, mediante los **pseudoselectores**.

Definiremos en la cabecera de la página al menos los estilos para **a:link**, **a:hover** y **a:visited**.

## Estructura de una regla CSS

Ya hemos visto algunos ejemplos, pero ¿cómo se define una regla CSS? Consta de varias partes:

**elemento, clase o id** { propiedad: valor ; propiedad: valor ; propiedad: valor ; ... }

- En la parte izquierda indicaremos el elemento, la clase o el identificador a modificar.
- En los corchetes estableceremos parejas formadas por una **propiedad** y el **valor** que se le va asignar, separando cada pareja por un signo de **punto y coma**.

Los espacios y los saltos de línea no afectan a las reglas CSS. De hecho, es muy habitual escribir esas instrucciones así:

```
elemento, clase o id {  
  
    propiedad: valor;  
    propiedad: valor;  
    propiedad: valor;  
  
...}
```

Se gana en visibilidad y resulta más sencillo localizar las diferentes propiedades.



### Pregunta Verdadero-Falso

La siguiente afirmación, ¿es verdadera o falsa?

Sólo si se aplican las reglas de estilo CSS de la forma:

```
elemento, clase o id {  
  
propiedad: valor;  
propiedad: valor;  
propiedad: valor;  
  
...}
```

obtendremos el resultado web deseado.

Verdadero ☐ Falso ☐

### Inserción de reglas CSS en una página web

Ya sabemos que podemos insertar las reglas directamente dentro de un elemento de la página web mediante el parámetro **style**, pero como inconveniente obtenemos un método poco claro y que resulta difícil de mantener. Por tanto, siempre optaremos por separar los estilos del contenido de la página.

Así, los estilos los situaremos en la cabecera de la página, rodeados de una etiqueta HTML denominada **<style>** con su correspondiente **</style>**. El ejemplo anterior quedaría integrado en nuestra página de la siguiente manera:

```
<!DOCTYPE html>  
<html>  
<head>  
  
    <style type="text/css">  
        h1 { color: blue;}
```

```
.capitulos { font-family: sans-serif; text-decoration: underline; }
#anexos { font-size: 18px; text-align: center; }
</style>

</head>
<body>

    Contenido de la página

</body>
</html>
```

Este método es fácil y su mantenimiento es simple. Si tenemos que realizar cualquier modificación, sólo tenemos que acudir a la cabecera de la página y modificar allí los estilos correspondientes, para que automáticamente toda la página cambie su apariencia.



## Pregunta Verdadero-Falso

La siguiente afirmación, ¿es verdadera o falsa?

El parámetro style no se puede emplear, para insertar las reglas CSS directamente dentro de un elemento de la página web.

Verdadero ☐ Falso ☐

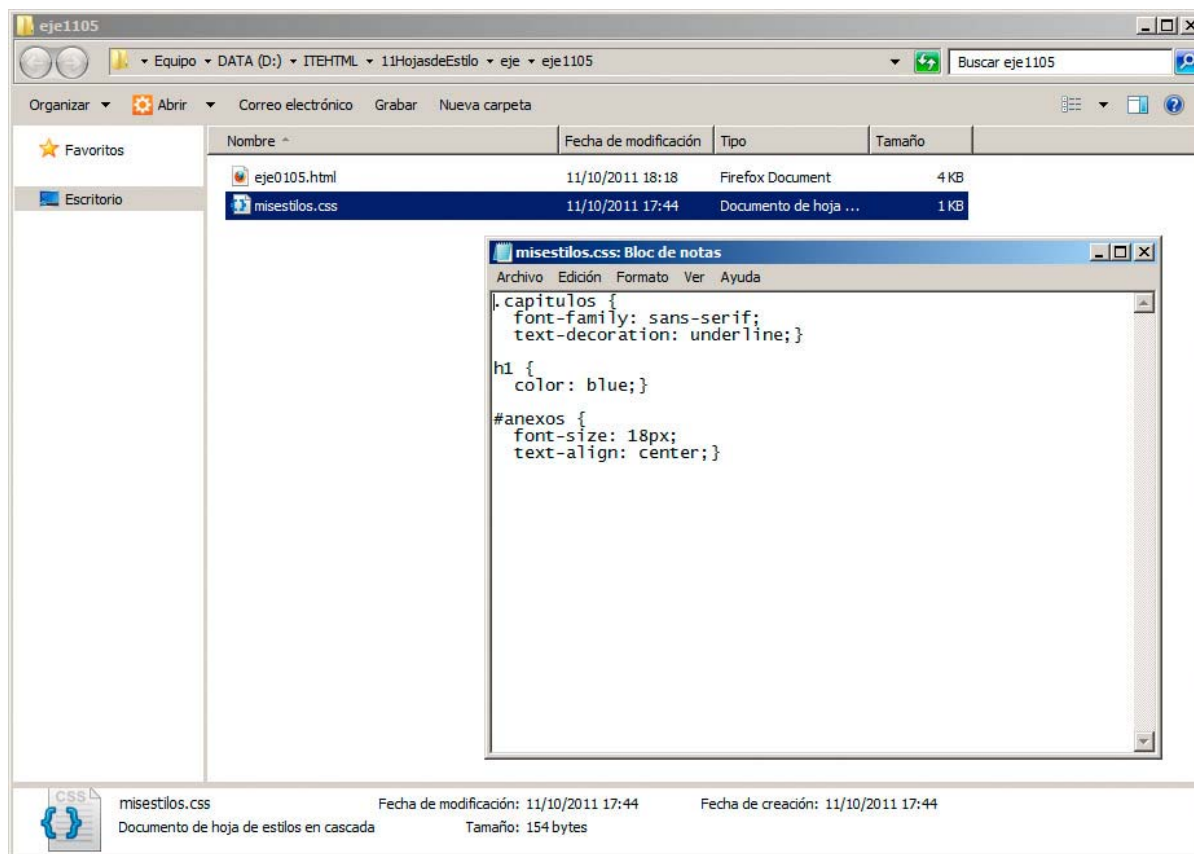
## Añadir estilos exteriores

Aún hay un método más para aplicar estilos, consistente en almacenar los estilos fuera de la página web. Pensemos en un sitio web con 20 páginas a las que le vamos a dar una apariencia común. Para usar los estilos en la cabecera de la página, tendríamos que copiar el conjunto de reglas y pegarlo en cada una de las páginas. Sería viable, pero cada vez que fuésemos a modificar algún detalle de los estilos, tendríamos que abrir las 20 páginas y aplicar los cambios. Esto es poco racional.

Para trabajar cómodamente con estilos en sitios grandes, el método ideal se basa en crear un archivo con extensión .css, dentro del que meteremos todos los estilos de nuestro sitio web.

En concreto, un archivo de estilos tendría una apariencia similar al de la figura.





## Enlazar un archivo .css

Para que el archivo con los estilos tenga efecto en la página, debemos indicar en la cabecera del archivo .html que se va a tomar ese estilo, mediante la etiqueta **<link>**, así:

```
<link rel="stylesheet" href="misestilos.css" type="text/css" media="all">
```

Cada parámetro tiene su propio significado:

- **rel**: indica la relación entre el documento y el archivo que se carga; en este caso será su hoja de estilos.
- **href**: como en los enlaces normales, indica la URL del archivo. Al encontrarse en la misma carpeta, sólo indicaremos el nombre; pero si estuviese en otro lugar o incluso en otro servidor, crearíamos las rutas siguiendo las normas de los enlaces.
- **type**: indica el tipo de información. Tanto este valor como el de **rel**, serán siempre iguales.
- **media**: especifica a qué medio se aplicará la hoja de estilo. En este caso se aplica a todos los medios, pero podríamos tener hojas de estilo diferentes para impresión (**print**), móviles (**handheld**), televisiones (**tv**), etc.

Por tanto, esa línea será la que repetiremos a través de nuestras diferentes páginas, para que tomen la apariencia deseada.



## Pregunta Verdadero-Falso

Las siguientes afirmaciones, ¿son verdaderas o falsas?

Para trabajar cómodamente con estilos en sitios grandes, el método ideal se basa en crear un archivo con extensión .css, dentro del que meteremos todos los estilos de nuestro sitio web.

Verdadero ☐ Falso ☐

Para que el archivo con los estilos tenga efecto en la página, debemos indicar en la cabecera del archivo .html que se va a tomar ese estilo, mediante la etiqueta **<type>**.

Verdadero ☐ Falso ☐



### Nota

Otra forma de importar una hoja de estilos es mediante la secuencia **@import url(misestilos.css)**. Esta línea se insertaría entre las etiquetas **<style>** y **</style>**. Este método se puede usar también para importar una hoja de estilos dentro de otra hoja de estilos.

## Múltiples hojas de estilo

Un detalle a tener en cuenta es que una misma página web puede emplear varias hojas de estilo diferentes. De hecho es una práctica muy frecuente para separar, por ejemplo, los aspectos relativos a maquetación de los puramente gráficos.

Cuando llegue el momento de preparar la página web, el navegador juntará todos los estilos y los ordenará, para aplicarlos correctamente.

### Prioridad de los estilos

Al crear estilos en diferentes lugares, nos surge una duda, ¿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>**? ¿Cuál se aplicará?

Las hojas de estilo se denominan "en cascada" precisamente por la forma que tienen de definir su prioridad. Cuando un mismo estilo está definido en varios lugares, se sigue un orden de preferencia muy claro:

- Primero se aplican los estilos de las hojas externas.
- Si en la cabecera de la página se repite alguna propiedad de un estilo, se aplica el de la cabecera, obviando la misma propiedad que estuviese definida en la hoja externa.
- Si en un elemento concreto se ha definido una propiedad específica, se aplica esta última, saltándose las anteriores.

Es decir, se va de mayor a menor, aplicando cada propiedad. Insistimos en lo de la propiedad, ya que podemos encontrarnos con que en la hoja exterior se define el tipo de letra de los párrafos y en el documento interior se define el tamaño. En ese caso se aplicarían las dos propiedades, ya que no hay colisión entre ellas.



### Pregunta de Elección Múltiple

Respecto al orden de preferencia en la aplicación de reglas de hojas de estilos CSS, podemos decir:

- ☐ Primero se aplican los estilos de las hojas internas.
- ☐ Si en la cabecera de la página se repite alguna propiedad de un estilo, se aplica el de la cabecera, obviando la misma propiedad que estuviese definida en la hoja externa.
- ☐ Si en un elemento concreto se ha definido una propiedad específica, se salta esta última, se aplican las anteriores.



### Nota

Deberíamos intentar evitar siempre el uso del parámetro **style** en cualquier elemento de una página. Nuestras páginas se vuelven más complejas con su uso y más difíciles de modificar. Es más apropiado mantener los estilos en archivos externos.

El uso del modificador **!important** en una determinada regla hace que ésta se salte la cadena de prioridades y que se aplique de forma prioritaria. Se emplea así:

```
p {margin: 6px !important; }
```

Su uso se suele hacer más en la fase de diseño de la página web que una vez terminada, ya que pocas veces se justifica el saltarse el orden predefinido de los estilos.

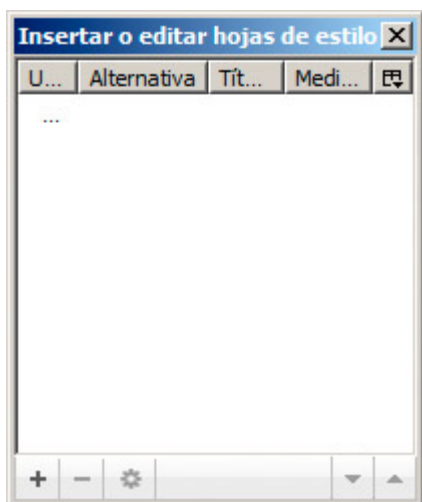
## Aplicar estilos con un editor de páginas web

Antes de comenzar a aplicar estilos a un documento, crearemos el espacio necesario para acoger las reglas. Como venimos explicando, lo más apropiado es contar con una hoja de estilos externa; así que eso es lo que haremos.

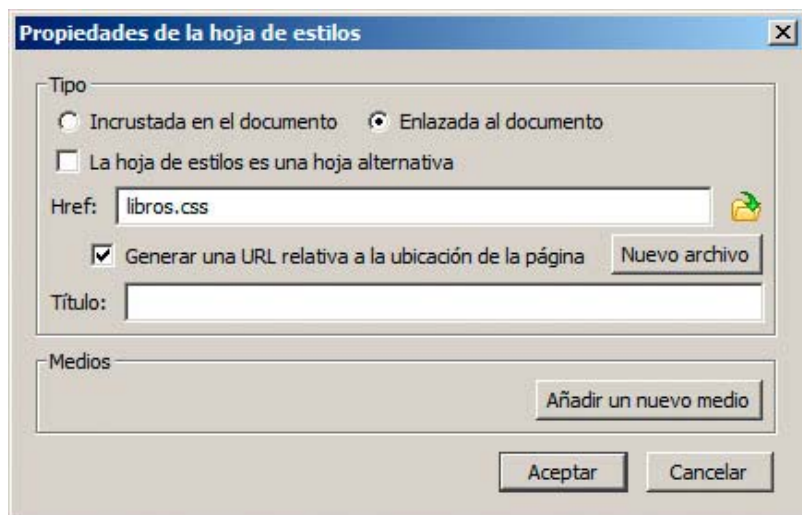
### Crear una hoja de estilos externa

Una hoja de estilos es un documento de texto que guardamos con extensión .css; no obstante, programas como *BlueGriffon* nos permiten crear la hoja desde sus diferentes opciones, siguiendo estos pasos:

1. En *BlueGriffon* abriremos el documento al que le vamos a aplicar formato.
2. En el menú **Insertar** seleccionaremos la opción **Hojas de estilo**. Aparece el cuadro de la figura:



3. Haciendo clic en el botón con el signo de suma, en la esquina inferior derecha aparece el cuadro de diálogo de la figura, para añadir una hoja de estilo al documento.



4. Como vamos a crear una hoja externa, seleccionaremos la opción **Enlazada al documento** y haremos clic en **Nuevo archivo**. Indicaremos el nombre que le queramos dar a nuestra hoja.

5. Para que no tengamos problemas al subir la página a Internet, activaremos la casilla **Generar una URL relativa a la ubicación de la página**.

6. Haremos clic en el botón **Aceptar**.

7. Cerraremos la ventana de estilos.

Con esto hemos terminado. Si miramos el código fuente de la página mediante el botón **source**, podremos ver el enlace dentro de la cabecera.

```

22
23 <link href="libros.css" rel="stylesheet" type="text/css">
24 </head>
25 <body>
26

```

La ventana de inserción de estilos merece un análisis más detenido. Con ella podremos hacer todas las operaciones que necesitemos respecto a los estilos:

Es posible, por ejemplo, incorporar varias hojas de estilo diferentes e incluso añadir estilos en el encabezado.

Haciendo clic en el botón **Añadir un nuevo medio**, podemos establecer hojas de estilos diferentes para impresión, para dispositivos móviles, etc.

En el vídeo siguiente realizamos el proceso habitual de creación de una plantilla externa.



### Pregunta Verdadero-Falso

La siguiente afirmación, ¿es verdadera o falsa?

Para que no tener problemas al subir la web a Internet, es necesario activar la casilla Generar una URL relativa a la ubicación de la página, dentro del programa *BlueGriffon*.

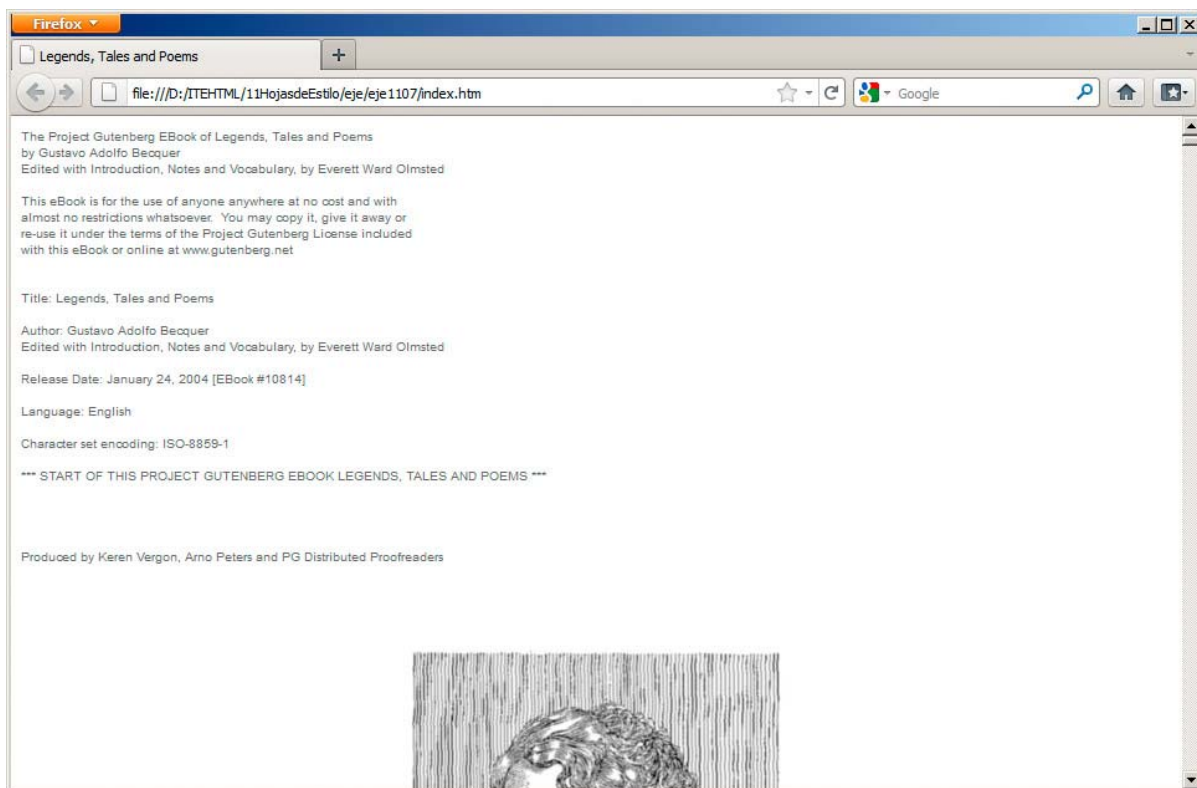
Verdadero ☐ Falso ☐



### Vídeo

Crear una plantilla externa.

El resultado del trabajo realizado con el vídeo se muestra en la figura:



Con sólo un par de líneas hemos variado la apariencia de la parte inicial del documento y, lo que es más importante, podemos proporcionar ese mismo aspecto a cuantos documentos queramos y modificarlos con sencillez.



## Actividad 5

Crearemos una hoja de estilos externa siguiendo los métodos analizados y llevaremos a ella todos los estilos que hemos definido hasta el momento. Si encontramos alguna dificultad al hacerlo siempre podremos recurrir a cortar los estilos con un editor de textos desde el archivo .html para pegarlos posteriormente con el mismo editor en el archivo .css

Una vez que los estilos han sido insertados en la página web debemos eliminarlos de la cabecera de nuestras páginas, para evitar estilos duplicados. Los estilos deben permanecer únicamente en el archivo externo y en las páginas sólo tendremos un enlace a ese archivo.

## Resumen

# Resumen



A lo largo de este módulo hemos revisado los conceptos teóricos fundamentales para emplear las hojas de estilo y comenzar a diseñar nuestras propias reglas.





Contamos con tres formas fundamentales para definir la apariencia de un elemento de una página web:

- Redefinir la **etiqueta** de forma genérica, con lo que cambiaríamos todas las apariciones de esa etiqueta en la página (indicando simplemente el nombre de la etiqueta).
- Modificar una **clase**: precediendo el nombre de la clase del signo ".", cambiaremos todas las apariciones de los elementos que pertenezcan a esa clase.
- Modificar un **identificador** específico, que se identifica mediante el signo "#". En este caso, sólo se aplicará el estilo al elemento que pertenezca a ese identificador.

Junto a estos tres métodos contamos con diferentes combinaciones y alternativas. Éstas son las más empleadas:

- Podemos mezclar clases e identificadores.
- Podemos modificar sólo las clases e identificadores de una etiqueta específica.
- Podemos hacer que sólo se modifiquen aquellas clases o identificadores que descendan de un elemento concreto.
- Podemos modificar algunos estados específicos de una etiqueta o los elementos que están en una determinada posición.

Para terminar, las reglas que definimos se pueden establecer en tres lugares diferentes, ordenados de menor prioridad a mayor:

- En un **archivo externo**, enlazado con la página con la etiqueta `<link>`.
- En la **cabecera de la página** (mediante la etiqueta `<style>`).
- En un **elemento concreto** de la página (con el parámetro **style**).

De los tres, el primer método es el más aconsejable.

## Actividades y ejemplos



### Actividad 1. ¿Qué son las hojas de estilo?

Utilizando un editor de páginas web como *BlueGriffon* y un documento base, redefiniremos el estilo de la etiqueta `<p>` y observaremos los cambios que se producen. Para realizar la edición, accedemos a la opción **Propiedades de estilo** del menú **Paneles**. En la ventana que se despliega utilizaremos las opciones de los paneles **General** y **Colores** para redefinir la etiqueta `<p>`. Antes de comenzar, debemos seleccionar la opción **todos los elementos del mismo tipo** en el cuadro **Aplicar estilos a**.

Podemos localizar un texto largo en la web del *Proyecto Gutenberg* (<http://www.gutenberg.org>), donde se pueden descargar libros completos cuyos derechos de *copyright* ya han vencido. De ese libro extraeremos un capítulo concreto, ya que el tratamiento del libro completo puede resultar ligeramente lento.



### Actividad 2. Clases e identificadores

Utilizando un editor de páginas web y el documento base anterior, definiremos un par de clases diferentes para distinguir los títulos de capítulos de los títulos generales o distintos tipos de párrafos, lo que mejor se ajuste a nuestro texto. Para ello estableceremos el nombre de cada clase en el cuadro **Clase** de la barra del editor y en el panel **Propiedades de estilo** seleccionaremos la opción **todos los elementos de la clase** en el cuadro **Aplicar estilos**

a:

Crearemos también un identificador en algún elemento. En este caso indicaremos un nombre en el cuadro **Identificador** de la barra del editor, situado justo antes del cuadro **Clase**. Para aplicar estilos a un identificador, seleccionaremos la opción **este elemento mediante su ID** en el cuadro **Aplicar estilos a:** del panel **Propiedades de estilo**.

Aplicaremos estilos distintos a las clases y el identificador, modificando algunos aspectos generales del texto, como su tamaño, color o su alineación.



### Actividad 3. Clases e identificadores

En el documento anterior tomaremos una etiqueta que tenga nombre de clase y en su interior definiremos una parte como **<strong>**. Haremos lo mismo con otra etiqueta que no tenga nombre de clase. Por último, definiremos un selector para que se aplique formato sólo a la etiqueta que tiene nombre de clase y no a todos los **<strong>** del documento. Es decir, recurriremos al **selector de sucesores** para conseguir el efecto.

Estos cambios no los realizaremos con *BlueGriffon*, sino que definiremos los selectores directamente en el código fuente de la página, en la cabecera del documento.



### Actividad 4. Clases e identificadores

Siguiendo con nuestro ejemplo, crearemos un enlace a la página desde la que tomamos el texto original y definiremos estilos para los diferentes estados posibles del enlace mediante los **pseudoselectores**.

Definiremos en la cabecera de la página al menos los estilos para **a:link**, **a:hover** y **a:visited**.



### Actividad 5. Aplicar estilos con un editor de páginas web

Crearemos una hoja de estilos externa, siguiendo los métodos analizados, y llevaremos a ella todos los estilos que hemos definido hasta el momento. Si encontramos alguna dificultad al hacerlo, siempre podremos recurrir a cortar los estilos con un editor de textos desde el archivo .html, para pegarlos posteriormente con el mismo editor en el archivo .css.

Una vez que los estilos han sido insertados en la página web, debemos eliminarlos de la cabecera de nuestras páginas, para evitar estilos duplicados. Los estilos deben permanecer únicamente en el archivo externo y en las páginas sólo tendremos un enlace a ese archivo.



### Ejemplos

Las diferentes prácticas, recursos y ejemplos realizados en este módulo están disponibles para realizar pruebas.

[Ejemplos del módulo](#)

## Aplicación al aula

# Hojas de estilo

Trabajaremos con el alumnado en la creación de hojas de estilos y la comprensión de los conceptos asociados a ellas.



## Programación dirigida al alumnado

### Objetivos

- Conocer los conceptos básicos de las hojas de estilo en cascada.
- Crear los espacios necesarios para aplicar estilos.
- Añadir estilos manualmente.
- Crear hojas de estilo con una aplicación.

### Contenidos

- Hojas de estilo.
- Archivos externos de hojas de estilo.
- Creación de estilos con un editor web.

### Materiales y recursos

- Ordenador con acceso a Internet.

### Temporalización

- Dos sesiones

## Planificación



### Planteamiento de la actividad

El alumnado aprende los conceptos básicos de la creación de estilos y comienza su aplicación progresiva sobre un documento modelo.

### Organización del aula

Trabajaremos en un aula con ordenadores con un agrupamiento individual o por parejas.

## Desarrollo de la actividad

- Se analizan los conceptos básicos de creación de hojas de estilos en cascada.
- Se localiza un texto apropiado.
- Se añaden algunos estilos al documento.
- Se guardan los estilos de diferentes maneras.

## Presentación y evaluación de los resultados

La evaluación se realizaría mediante la revisión del resultado y la observación del proceso. Se pueden evaluar varios aspectos a lo largo de todo el proceso:

- Capacidad para localizar el documento externo.
- Inserción apropiada de los estilos en los lugares indicados.
- Apariencia de los cambios aplicados en el documento.

## Sugerencias metodológicas



La metodología empleada es la de **proyecto**.

Para su aplicación proponemos:

### Primera sesión

Explicamos el objetivo de la actividad y describimos los conceptos necesarios.

Localizamos un documento largo en Internet, en formato html o lo convertimos a HTML. Puede ser el mismo texto para todos los alumnos. En la web del *Proyecto Gutenberg* ([www.gutenberg.org](http://www.gutenberg.org)) podemos encontrar algunos ejemplos apropiados.

Aplicamos algunos estilos básicos en la cabecera del documento.

### Segunda sesión

Retomando el ejemplo anterior creamos una hoja de estilos externa y aplicamos algunos estilos básicos.

Probamos a definir reglas similares en el documento externo y en la cabecera, para que los alumnos comprendan los conceptos de prioridad de las reglas.

Modificamos los estilos hasta obtener el resultado apropiado.

Desplazamos todos los estilos al archivo externo, dejando limpio el documento HTML.

## Atención a la diversidad



### Actividad de refuerzo

---

Para aquellos alumnos/as que puedan tener más dificultad, se les facilita el documento previamente e incluso se les entrega la hoja de estilos vacía.

---



### Actividad de ampliación

---

La profundización en esta actividad se basaría en que el alumnado probase algunos de los selectores más específicos, de descendencia, hijos, etc.

---