



MINISTERIO
DE EDUCACIÓN
Y CIENCIA

SECRETARÍA GENERAL
DE EDUCACIÓN
Y FORMACIÓN PROFESIONAL

DIRECCIÓN GENERAL
DE EDUCACIÓN,
FORMACIÓN PROFESIONAL
E INNOVACIÓN EDUCATIVA

CENTRO NACIONAL
DE INFORMACIÓN Y
COMUNICACIÓN EDUCATIVA

Edición HTML

Estilos CSS Introducción y sintaxis



SERVICIO DE
FORMACIÓN DEL
PROFESORADO

C/ TORRELAGUNA, 58
28027 - MADRID

Índice de contenido

Cascading Style Sheets.....	3
¿Por qué se denominan hojas de estilo?.....	4
¿Por qué se denominan "en cascada"?.....	4
¿CSS1, CSS2 o CSS3?.....	4
Sintaxis básica	5
El editor CSS de N VU.....	5
Selectores de tipos.....	6
Selector universal.....	7
Agrupando selectores	7
Selectores contextuales.....	8
Selectores basados en atributos.....	11
Clases	11
Pseudoclasas.....	13
Pseudoclasas relacionadas con los vínculos.....	13
El selector id	15
Comentarios.....	16

CASCADING STYLE SHEETS

O lo que es lo mismo: hojas de estilo en cascada.

El motivo de haber puesto en primer lugar la denominación en inglés es porque vas a encontrar en muchos momentos estas siglas para referirse al concepto que trabajaremos en este epígrafe. Poco a poco iremos descifrando el significado real de esta denominación que, por ahora, puede resultarnos un tanto críptica.

Lo primero en lo que coincidiremos es en que el lenguaje HTML es francamente antipático y poco flexible cuando se trata de incidir en la apariencia de una página. El problema real está en que la concepción original del mismo no pretendía orientarlo hacia el aspecto gráfico: la misión principal de las etiquetas HTML es proporcionar estructura lógica y semántica al contenido. Pensemos que el lenguaje HTML surgió en un entorno, el universitario, y con un propósito, la publicación de contenidos científicos estructurados, eminentemente textuales. El problema es que la masiva difusión de Internet hizo que se rompiera el marco original y se hiciera necesario, especialmente desde la irrupción de un uso comercial, presentar páginas en las que el centro de gravedad se trasladaba del contenido y su estructura a la forma, ya que se trataba de crear "imagen de marca". Fue entonces cuando empezaron a aparecer los problemas, ya que el lenguaje HTML no está pensado para trabajar con los elementos formales sino con los de tipo estructural.

Si te paras a pensar un poco sobre lo que acabas de leer puedes darte cuenta por ejemplo de que, aunque hayamos podido utilizar las etiquetas `<h1>` para conseguir un texto de un determinado tamaño que nos pudiera valer de titular lo lógico habría sido emplearlas para organizar los contenidos indicando que su contenido es el título de un fragmento que tiene una importancia superior, que aquellos que van encabezados por un `<h2>`

El crecimiento exponencial de Internet hizo que la imagen que nos hemos hecho de lo que debe ser una página WEB fuera acercándose a lo que es hoy: un entorno multimedia en el que los aspectos formales han cobrado una gran importancia. Pero esta percepción sigue apoyándose en la utilización del mismo lenguaje que le había servido de base en los primeros tiempos, lo cual ha hecho que los diseñadores hayan tenido que recurrir a trucos y artimañas para paliar las deficiencias del lenguaje HTML en lo referente a la presentación visual.

Al tratarse de añadidos a la finalidad original, y en muchos casos utilizations francamente contradictorias con la misma, podemos encontrarnos con situaciones muy problemáticas cuando pretendemos introducir modificaciones globales en un sitio WEB. ¿Imaginas, por ejemplo, el trabajo que sería necesario para cambiar el tipo de letra de un sitio WEB compuesto por cincuenta páginas?

La forma de solucionarlo ha sido la utilización de las hojas de estilo en cascada. Las hojas de estilo en cascada aportan, como característica esencial, la posibilidad de separar los elementos formales de los de contenido, de forma que con una pequeña modificación de la hoja de estilo se pueda cambiar el aspecto de todas las páginas en las que se aplica esa hoja. Volviendo al ejemplo anterior tal vez nos bastaría con cambiar una vez la palabra sans-serif por serif para conseguir que las cincuenta páginas de nuestra web cambiaran automáticamente de apariencia.

Antes de empezar es conveniente considerar una cuestión: el W3C, como organismo encargado de la estandarización, elabora un conjunto de especificaciones normativas, pero... cada navegador ofrece un soporte más o menos ajustado a las especificaciones. De hecho nos encontramos en un momento en el que la especificación CSS3, que incluye una gran cantidad de posibilidades no contempladas en las versiones previas, no es soportada por ningún navegador. Incluso, la implementación de las especificaciones CSS2 presenta importantes deficiencias en Explorer 6 y anteriores, lo cual nos obligará a buscar algunos trucos para lograr efectos similares en todos los navegadores. Es muy probable que las nuevas versiones de los navegadores que tal vez hayan aparecido cuando leas esta documentación

superen estos problemas y se ajusten a la normativa común.

¿POR QUÉ SE DENOMINAN HOJAS DE ESTILO?

La denominación de hojas de estilo hace referencia a que podemos, a partir de las etiquetas básicas de HTML, realizar modificaciones en las propiedades de las mismas, indicándole al navegador unas instrucciones muy precisas sobre la forma en la que tiene que presentar un determinado elemento, esto es, el estilo de la presentación.

Una ventaja añadida es que, al tratarse de una colección de instrucciones escrita en formato de texto. Si recurrimos a su uso más aconsejable, dichas instrucciones estarán separadas de la propia página web y serán aplicables a múltiples páginas simultáneamente, por lo que conseguiremos que el código de nuestra página en sí resulte mucho más ligero.

Por otra parte, al quedar un código mucho más limpio gracias a la descarga de los elementos de formato, la estructura de la información quedará mucho más clara ya que el código tenderá a ser muy simple.

¿POR QUÉ SE DENOMINAN "EN CASCADA"?

A la hora de crear un estilo, las especificaciones que se realicen para un determinado elemento serán aplicables a todos aquellos elementos que se encuentren "por debajo" de él atendiendo a un criterio de herencia. Por ejemplo, si especificamos un tipo de letra para el elemento <body> todos aquellos elementos que puedan heredar las características se presentarán con el mismo tipo de fuente. Así, el elemento <p> heredarán el tipo de letra salvo que especifiquemos lo contrario y no necesitaríamos volver a indicar esta característica.

Gracias a la organización en cascada podemos crear hojas de estilo muy compactas, ya que una sola norma en el primer elemento de la cadena de herencia permite que podamos aplicarla a todos los herederos.

Además de ello se establece un orden de prioridad para la aplicación de las normas en caso de que resulten contradictorias. El orden de preferencia para aplicar una norma es el siguiente:

1. Estilo especificado dentro de la etiqueta. (estilo en línea)
2. Estilo especificado en la cabecera del documento. (hoja de estilos interna)
3. Estilo definido en un documento independiente al que se enlaza nuestra página. (hoja de estilos externa)

Teniendo en cuenta el orden mencionado, el procedimiento más cómodo es crear una hoja de estilo en un archivo independiente y vincular las páginas a este archivo. Si queremos realizar alguna modificación en un elemento concreto podríamos recurrir a uno de los dos primeros procedimientos que, al ser más específicos, resultarán preferentes.

¿CSS1, CSS2 O CSS3?

Es posible que en cuanto intentes localizar alguna información sobre hojas de estilo te aparezca la referencia a uno de los tres niveles. Obviamente son más avanzadas las especificaciones del nivel 3, pero ya hemos comentado que en el momento actual no están soportadas por la mayoría de los navegadores, por lo que haremos referencia a las de nivel 2.

Para introducirnos en el mundo de las CSS iremos presentando la información en pequeños bloques, acompañados de un ejemplo que muestre el resultado. En muchos casos les seguirá una práctica muy simple que permita aplicar lo expuesto. La recomendación es que vayas paso a paso y que, en cuanto tengas cierta seguridad, experimentes ampliando el ejercicio propuesto con elementos similares intentando prever los resultados que vas a obtener. Si lo haces así es posible que en algún momento te encuentres con que no todo sale como pretendes, pero no te preocupes porque lo más probable es que algo más adelante encuentres

la respuesta al problema que se te ha planteado.

SINTAXIS BÁSICA

La sintaxis general de cualquier declaración de estilo, va a estar compuesta por tres elementos:

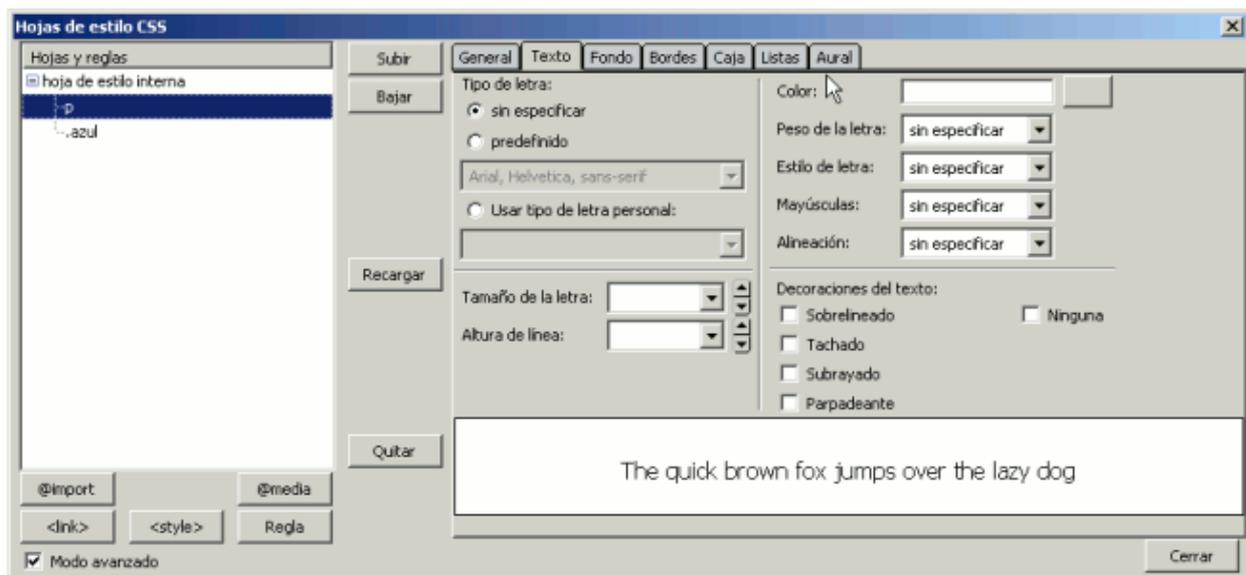
selector {propiedad:valor}

donde selector va a ser, básicamente, el elemento html que deseamos definir, la propiedad el atributo que queremos modificar, apareciendo por último el valor que deseamos asignarle al atributo. La propiedad se separa del atributo mediante dos puntos y ambos se encierran entre llaves. Aunque no es imprescindible cuando sólo hay una propiedad es conveniente añadir un punto y coma al final de cada par propiedad / valor (nos será muy útil para facilitar la lectura cuando se vayan complicando algo más las especificaciones ya que entonces sí que es necesario utilizar el punto y coma como separador)

Cuando los valores requieren especificar un tipo de unidad, esta se indica a continuación del valor numérico, sin espacio de separación: 1cm sería una notación correcta, mientras que 1 cm no lo sería.

EL EDITOR CSS DE N|VU

Verás que constantemente se hará referencia a los códigos con los que se obtiene un determinado efecto. Es conveniente que te vayas familiarizando con ellos, pero al principio te resultará de gran ayuda la utilización del editor de estilos que integra N|VU, al que puedes acceder desde la opción de menú **HERRAMIENTAS** ➔ **EDITOR CSS** del que tienes un vídeo demostrativo en la documentación en HTML



Aunque posteriormente veremos otras formas más eficientes vamos a empezar a familiarizarnos con el editor mediante la creación de hojas de estilo internas para lo que seguiremos estos pasos

1. Pulsar el botón style
2. Crear hoja de estilo

3. Pulsar **REGLA**
4. Indicar la clase, tipo o identificador. Para que la regla afecte a un elemento html habrá que poner su nombre, sin los ángulos.
5. Pulsar crear **REGLA DE ESTILO**
6. Utilizar las pestañas para indicar los atributos y valores aplicables.

- 76** • Carga en el editor la página **quijote1.html**
- Utiliza el Editor CSS para crear una nueva hoja de estilo.
 - Pasa a la vista  **Código fuente** para comprobar cómo se han creado en la sección <head> un par de selectores para englobar las reglas de estilos que iremos creando. Guárdalo como **actividad76.html**

SELECTORES DE TIPOS

El selector más sencillo es el que denominamos de tipo. Se refiere a una etiqueta definida en el lenguaje HTML. Su gran ventaja será que con una única modificación en una hoja de estilos podremos modificar la apariencia de todas las páginas de un sitio al variar la presentación de un determinado tipo.

Ej: `body {color:white}`

Para cada selector podremos establecer más de una propiedad. Para separarlas utilizaremos el punto y coma, siendo habitual para favorecer la legibilidad, que cada propiedad se escriba en un renglón.

Ejemplo



```
p {
text-align: center;
color: blue;
font-family: Arial,
Helvetica, sans-serif;
}
```

Aplicando las características definidas en el estilo cualquier párrafo se mostraría centrado...en color azul...con una fuente de "palo seco".

Este es un nuevo párrafo

Visualizar el formato resultante

Terminado

77

- Si lo habías cerrado vuelve a abrir el archivo **actividad76.html**
- Crea una regla para que los párrafos se presenten en color azul marino.
- Cierra el editor CSS y comprueba el resultado.
- Aprovecha la visualización para hacer que el texto Primera Parte se convierta en un título de tipo **h1** y el título del Capítulo 1 en un **h2**.
- Mira en la pestaña  **Código fuente** cómo se ha definido la regla para los párrafos
- Vuelve al editor de estilos y haz que el párrafo esté justificado y con letra de la familia sans-serif (tendrás que hacer doble clic sobre la hoja de estilo para que se desplieguen los selectores y poder elegir el que hemos creado).
- Crea una regla para que los títulos **h1** se presenten en fuente sans-serif y color granate y otra para que los **h2** sean también sans-serif pero en color gris de intensidad media.
- Comprueba en la pestaña  **Código fuente** cómo han quedado definidos los estilos
- Guarda el trabajo como **actividad77.html**

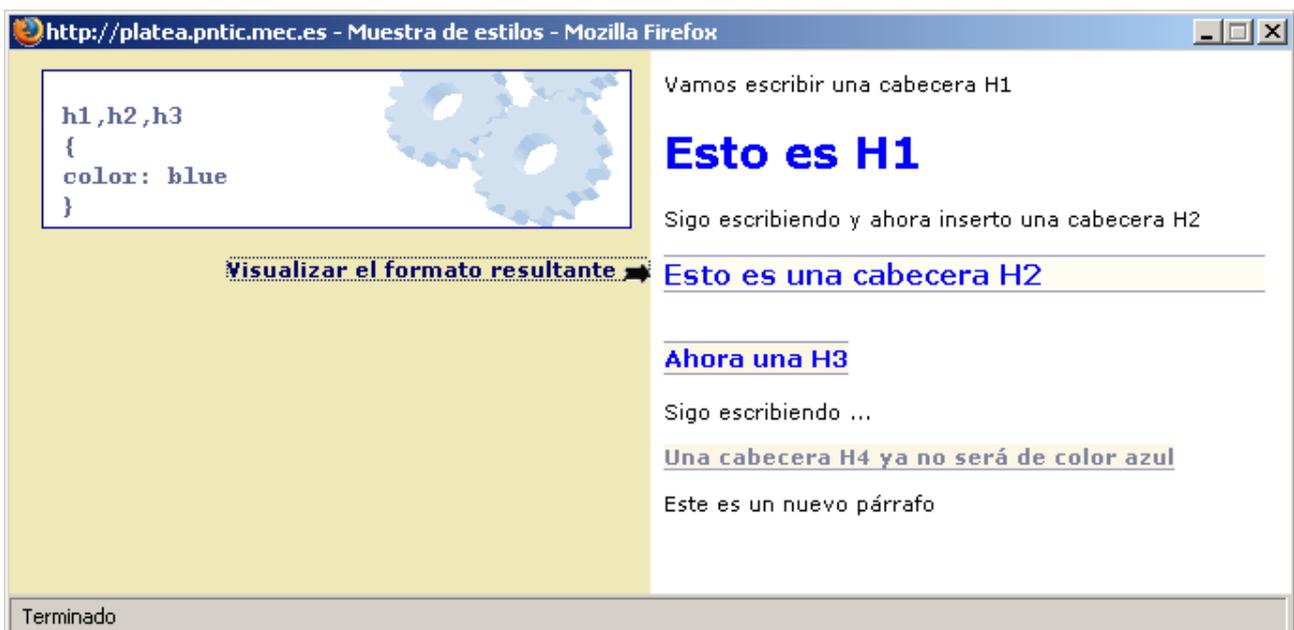
SELECTOR UNIVERSAL

Se representa mediante el asterisco (*) y se refiere a cualquier elemento del documento HTML. Su utilización está muy relacionada con las clases y selectores ID, pero suele omitirse.

AGRUPANDO SELECTORES

Si queremos aplicar una propiedad común a varios selectores sin necesidad de tener que definirla en cada caso podemos agruparlos listándolos separados por comas y haciendo una definición común para todos ellos.

Ej.: `h1,h2,h3 {color:blue}`



http://platea.pntic.mec.es - Muestra de estilos - Mozilla Firefox

```
h1, h2, h3
{
color: blue
}
```

Visualizar el formato resultante

Vamos escribir una cabecera H1

Esto es H1

Sigo escribiendo y ahora inserto una cabecera H2

Esto es una cabecera H2

Ahora una H3

Ahora una H3

Sigo escribiendo ...

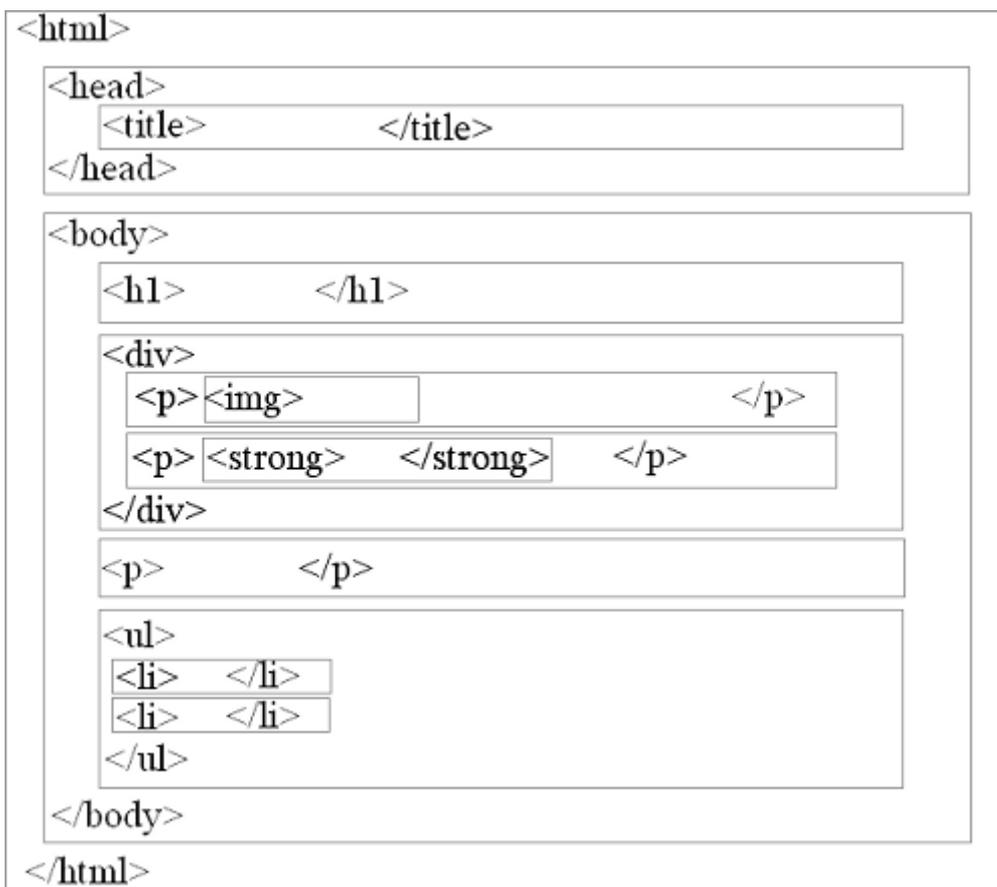
Una cabecera H4 ya no será de color azul

Este es un nuevo párrafo

Terminado

- 78**
- Vuelve al archivo **actividad77.html** que recoge los últimos pasos que hemos dado.
 - Crea una regla que agrupe **h1** y **h2** para especificar que deben llevar el tipo de letra de la familia sans-serif.
 - Borra este atributo en cada una de las reglas individuales para **h1** y **h2**
 - Comprueba que ambos títulos siguen presentándose con la fuente que hemos seleccionado.
 - Guarda este nuevo paso como **actividad77.html**

SELECTORES CONTEXTUALES



La imagen anterior corresponde a un ejemplo esquemático de la organización de los elementos en una página html.

Se puede observar cómo determinados elementos se encuentran "anidados" dentro de otros.

Esta situación genera unas relaciones que pueden calificarse de familiares (de hecho, la nomenclatura para indicarlas utilizará los términos padre, hijo, ascendiente y descendiente entre otros).

Vamos a ver a continuación algunos ejemplos de lo que significa esta jerarquía en la organización de los elementos de una página html para lo que podremos guiarnos por el siguiente esquema de relaciones:

Relación	Descripción
Un elemento es padre de otro si...	El segundo elemento se encuentra directamente entre la apertura y el cierre de las etiquetas del elemento padre. Ej: <code><p>Los rumiantes son mamíferos</code> El elemento <code><p></code> es padre de <code></code> y de <code></code>
Un elemento es hijo de otro si...	La relación anterior es recíproca, por lo que los elementos <code></code> y <code></code> del ejemplo son hijos del elemento <code><p></code>
Un elemento es ascendiente de otro si...	Si el segundo se encuentra entre las etiquetas de apertura y cierre del primero (aunque sea de forma indirecta). Ej: <code><div><p>Los rumiantes son mamíferos</div></code> El elemento <code><div></code> es ascendiente de <code><p></code> de <code></code> y de <code></code>
Un elemento es descendiente de otro si...	Está incluido, directa o indirectamente, dentro de otro elemento. En el ejemplo anterior <code></code> es descendiente de <code><p></code> y también de <code><div></code>
Un elemento es adyacente a otro si ...	Comparte el mismo padre y, además se encuentra inmediatamente consecutivo al anterior. Ej: <code><p>Los ungulados son mamíferos con pezuñas</code> En este ejemplo sólo el primer <code></code> sería adyacente a <code></code> ya que el segundo, aunque tiene el mismo padre ya no es inmediatamente posterior.

Los patrones para indicar estas relaciones para elaborar los selectores serían los siguientes:

Patrón	Utilización
x y	Para apuntar hacia cualquier elemento del tipo y que sea descendiente de un elemento del tipo x
x > y	Para referirnos a cualquier elemento del tipo y que sea hijo de un elemento del tipo x
x + y	Para identificar cualquier elemento del tipo y que sea adyacente a un elemento del tipo x

Teniendo en cuenta estas relaciones podemos realizar una selección muy específica de elementos del documento para aplicarles un determinado estilo. Podríamos, por ejemplo querer que los fragmentos que estuvieran enfatizados mediante la etiqueta `` aparecieran en color verde. Pero podríamos ser incluso más específicos indicando que sólo queremos que aparezcan en color verde los fragmentos enfatizados que formen parte de una lista. Esto lo haríamos colocando primero el selector `li` y, a continuación separado por un espacio, el selector `em` para luego especificar las propiedades

Ej.: `li em {color: green}`

http://platea.pntic.mec.es - Muestra de estilos - Mozilla Firefox

```

<head>
...
<style type="text/css">
li em {color: green}
</style>
</head>
<body>
<p>La palabra
<em>"atención"</em> está
enfaticada pero se presenta en
su color normal porque:</p>
<ol>
<li>Está enfaticada pero...
</li>
<li><em>No pertenece a una
lista</em></li>
</ol>
<p>La segunda condición también
está enfaticada, pero en este
caso pertenece a una lista y se
ve en verde </p></body>

```

La palabra "atención" está enfaticada pero se presenta en su color normal porque:

1. Está enfaticada pero...
2. *No pertenece a una lista*

La segunda condición también está enfaticada, pero en este caso pertenece a una lista y se ve en verde.

Visualizar el formato resultante

Terminado

Advertencia importante: La versión 6 de Internet Explorer sólo es capaz de interpretar correctamente el selector contextual que utilizamos para apuntar hacia un descendiente. Aunque posteriormente veremos alguna forma de corregir este error tenlo en cuenta si intentas crear algún selector para elementos hijos o adyacentes, que sí funcionarán correctamente en los navegadores Mozilla, Firefox u Opera.

- 79**
- Abre en NIVU archivo **licenciacc.html**
 - Crea una regla para que todas los términos enfaticados con aparezcan en color azul
 - Crea otra regla para que los elementos que se encuentren dentro de un item de una lista se muestren en negrita.
 - ¿Te animas a crear una regla para que los elementos adyacentes a otro elemento se muestren de color rojo
 - Guarda el trabajo como **actividad79.html**

SELECTORES BASADOS EN ATRIBUTOS

Si la posibilidad de utilizar selectores contextuales aporta una gran flexibilidad, todavía es mucho mayor la que nos brinda la posibilidad de decidir que un estilo se aplicará a aquellos elementos que cumplan determinados atributos.

Patrón	Utilización
x[atributo]	Para apuntar hacia cualquier elemento del tipo x que cuente con un determinado atributo (independientemente del valor que tenga el atributo)
x[atributo="valor"]	Para referirse a cualquier elemento del tipo x que cuente con un determinado atributo cuyo valor sea el especificado
x[atributo~="valor"]	Algunos elementos pueden admitir varios valores separados por espacios para un atributo. Este patrón nos sirve para identificar a cualquier elemento del tipo x que cuente con un determinado atributo alguno de cuyos valores sea el especificado.

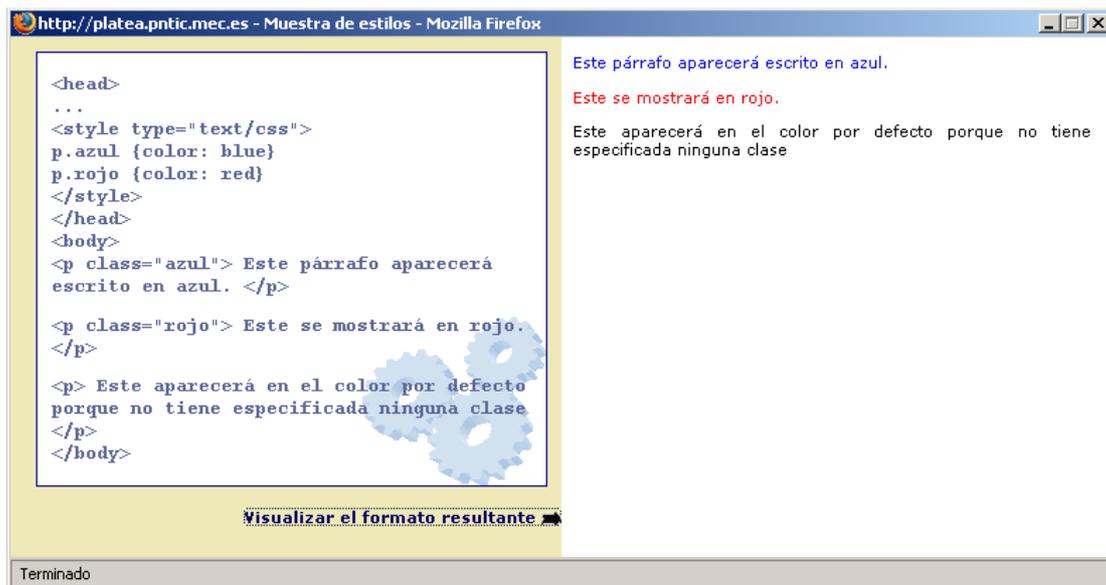
Al igual que ocurría en el caso de los selectores contextuales no existe soporte para este tipo de selectores en las versiones iguales o inferiores a la 6 de Internet Explorer.

- 80**
- Abre el archivo **selectoresporatributo.html**
 - Crear una hoja de estilo con una regla para que todos los elementos que incluyan el atributo **face** en la etiqueta **font** se muestren en negrita.
 - Añade una regla para que aquellos elementos que tengan el valor **Courier** como atributo único de **face** se muestren en itálica.
 - Por fin una tercera regla que haga que los que tengan, entre otros, el valor **serif** en el atributo **face** se muestren en color verde.
 - Comprueba visualmente los resultados. Recuerda que, mientras no apliquemos un parche, Internet Explorer será incapaz de aplicar estos selectores.

CLASES

El selector class nos permite definir diferentes estilos para el mismo elemento HTML. El ejemplo siguiente supondría la posibilidad de tener dos tipos de párrafo en el documento: uno en azul y otro en rojo, en función de la clase que cada uno lleve indicada. Para establecer la clase añadimos un punto y el nombre que le adjudicamos a la clase a continuación del selector y antes de iniciar las propiedades.

Ej.: `p.azul {color: blue} p.rojo {color: red}`



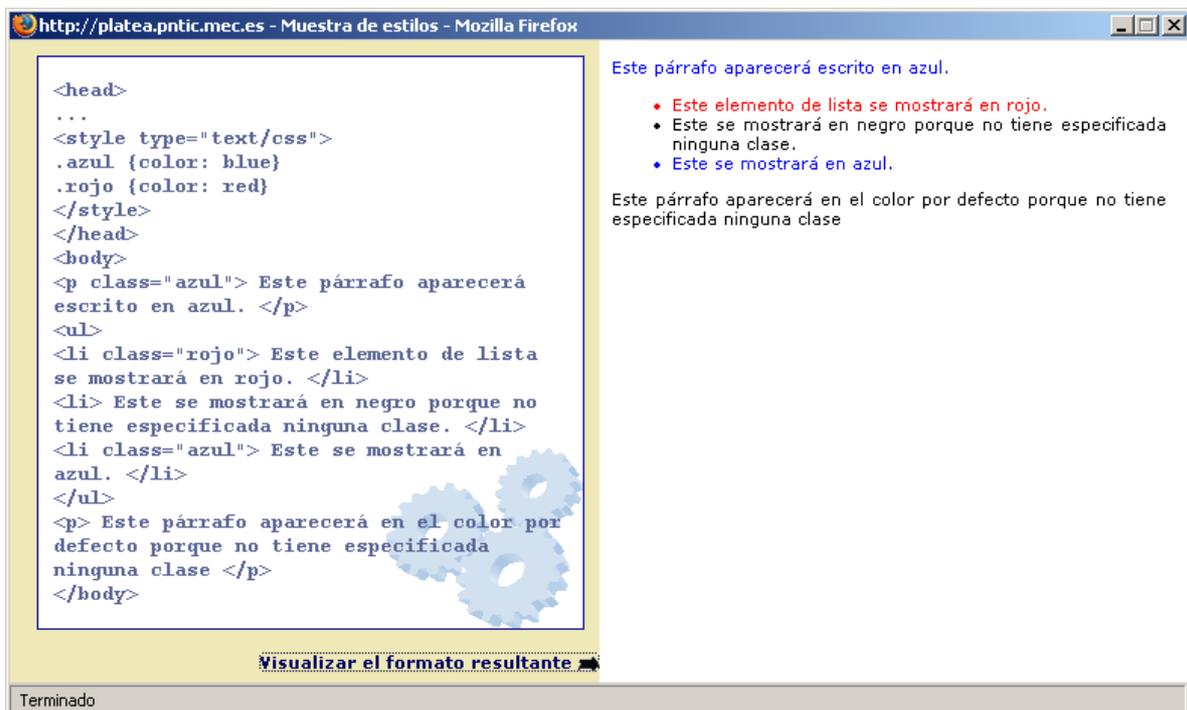
Se puede aplicar más de una clase a un elemento html. Al aplicar las clases lo haremos separándolas por un espacio.

Ej.: `<p class="azul centrado">Texto</p>`

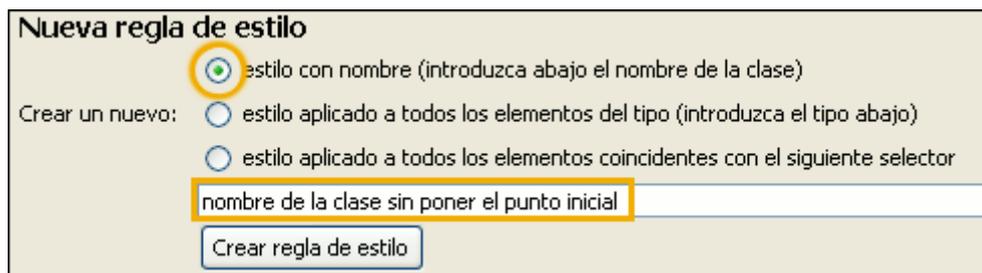
Es muy frecuente que la definición de las clases se haga sin ligarlas a una etiqueta concreta. En el ejemplo anterior podría omitirse la referencia al elemento p y eso haría que las características de la clase se aplicaran a cualquier elemento al que se le aplicara. En este caso el selector empieza con un punto seguido del nombre de la clase.

Ej.: `.azul {color: blue} .rojo {color: red}`

El efecto será equivalente si se utiliza el selector universal: `*.azul {color: blue} *.rojo {color: red}`



Para crear una regla de tipo clase en el editor de CSS utilizaremos la opción estilo con nombre y escribiremos el nombre sin precederlo con el punto (el programa se encargará de ello).



Nueva regla de estilo

estilo con nombre (introduzca abajo el nombre de la clase)

Crear un nuevo: estilo aplicado a todos los elementos del tipo (introduzca el tipo abajo)

estilo aplicado a todos los elementos coincidentes con el siguiente selector

nombre de la clase sin poner el punto inicial

Crear regla de estilo

Una vez definidas las clases, la aplicación de las mismas en N|VU se limita a marcar el fragmento que deseamos modificar arrastrando con el ratón y elegir la clase correspondiente en el listado que aparece en la segunda barra de formato. Con el procedimiento de pinchar arrastrar definiremos zonas del documento que no tienen que coincidir con un elemento completo, como podía ser un párrafo, lo cual hace que el programa genere una etiqueta `span` que englobe los fragmentos seleccionados.

Si lo que realmente queremos es aplicar una clase a un elemento, por ejemplo un párrafo, una lista ordenada completa, un contenedor genérico, etc, es preferible que lo seleccionemos situándonos sobre cualquier punto del mismo y marcando la etiqueta que lo identifica en la barra de estado (la que aparezca en último lugar corresponderá al punto en que nos encontramos). Una vez hecha esta marca podemos aplicar la clase:

- eligiéndola en el desplegable de la 2ª barra de formato
- pulsando con el botón derecho sobre el elemento señalado en la barra de estado y desplegando clases para elegir la adecuada

PSEUDOCLASES

Las pseudoclases se utilizan para particularizar algunos efectos de determinados selectores en algunas situaciones especiales. Uno de los grupos que se aplica más frecuentemente es el que tiene que ver con la etiqueta `<a>` para marcar diferentes presentaciones de los enlaces.

La sintaxis es ligeramente diferente de la general: **selector:pseudoclase {propiedad : valor}**

PSEUDOCLASES RELACIONADAS CON LOS VÍNCULOS

Su función es cambiar la apariencia según el estado del vínculo. Existen cuatro tipos:

- `a:link` sirve para establecer las propiedades de un enlace no visitado
- `a:visited` indica las propiedades que adoptará un enlace visitado
- `a:hover` especifica la apariencia que adoptará un enlace en el momento en que el botón se sitúa sobre él.
- `a:active` fija las propiedades que tomará un enlace en el momento en que se pulsa sobre él para activarlo.

Ejemplo:

http://platea.pntic.mec.es - Muestra de estilos - Mozilla Firefox

Desactivar CSS Formularios Imágenes Información Varios Resaltar Tamaño Herramientas Código fuente

```

<style type="text/css">
<!--
a:link {decoration:none;
color:blue}
a:visited {color:#9900FF;}
a:hover {font-style:italic}
a:active {color:#FF9900;}
-->
</style>

```

En este texto aparece un enlace que cambia su aspecto según su estado. Pasa por [aquí](#) con el ratón y pulsa para comprobar los cambios.

⚠ Para comprobar el cambio de color del enlace visitado en Mozilla, Galeón o Konqueror tendrás que recargar la página.

[Visualizar el formato resultante](#)

http://platea.pntic.mec.es/rluna/nvu2005/estilos/muestra21.htm#aquimismo

El orden en el que se deben especificar estas pseudoclases en las hojas de estilos para que tengan efecto es el que se ha utilizado para describirlas. Esto tiene su lógica debido a que, en caso de haber alguna contradicción entre dos selectores siempre se aplica el enumerado en último lugar: por ello el que se refiere al enlace activo debe ser el último para que se eviten variaciones en la presentación mostrándose las propiedades del enlace visitado (visited) o con el ratón encima (hover). Por el mismo motivo la pseudoclase hover debe aparecer detrás de visited para que la apariencia del enlace pueda modificarse al pasar el ratón sobre el mismo, independientemente de que se haya visitado o no el enlace. Por último, visited tendrá que ir después de link puesto que de lo contrario se aplicaría la apariencia del enlace no visitado aunque ya se hubiera estado en la página de destino.

Existen también los llamados pseudo-elementos que verás en la sección dedicada al texto pero todavía podemos considerar la pseudoclase que se refiere al primer elemento hijo `first-child` que nos permitiría obtener, por ejemplo un efecto similar al que presentan algunas ediciones impresas cuando el primer párrafo de un capítulo se muestra en un tamaño más grande o en itálica.

Ejemplo

http://platea.pntic.mec.es - Muestra de estilos - Mozilla Firefox

Desactivar CSS Formularios Imágenes Información Varios Resaltar Tamaño Herramientas Código fuente

```

<style type="text/css">
<!--
div.zonadcha > p:first-child
{font-size : large}
-->
</style>

```

Este párrafo es el primer hijo de un elemento div (es la primera vez que aparece la etiqueta `<p>` entre las etiquetas `<div>` y `</div>` que definen esta zona de la pantalla). Por eso aparece más grande.

Este segundo párrafo aparece ya en su tamaño normal.

⚠ Internet Explorer 6 y versiones previas no soportan esta pseudoclase

[Visualizar el formato resultante](#)

Terminado

- 81**
- Crea un documento que contenga un enlace a una página en la red (o bien hazlo con un enlace a un documento local y crea también este último)
 - Crea una hoja de estilo en la que incluyas características que te apetezcan para definir las pseudoclases en el orden especificado.
 - Comprueba las variaciones que se producen al pasar el ratón sobre el enlace o al visitarlo y volver nuevamente a la página.
 - Modifica el orden subiendo la pseudoclase `a:visited` al primer lugar y comprueba lo que ocurre.
 - Introduce otras variaciones en el orden para comprobar los efectos.
 - Guarda el trabajo como **actividad81.html**

EL SELECTOR ID

Mientras que el selector clase puede aplicarse a varios elementos del mismo o diferente tipo en una página podemos crear también un selector id para diferenciar elementos específicos que añadirán alguna característica diferente de las que tienen los elementos de su clase. El nombre del selector se inicia con el signo # seguido del nombre del identificador.

Ej: `#rojo {color: red}`

Ejemplo

The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://platea.pntic.mec.es - Muestra de estilos - Mozilla Firefox`. The main content area is split into two parts:

- Left side (Code Editor):** Contains the following HTML and CSS code:


```
<head>
...
<style type="text/css">
.importante {font-weight : bold}
#muy-importante {color: red}
</style>
</head>
<body>
<p>Escribimos una lista de condiciones que consideramos importantes:</p>
<ul class="importante">
<li>Primera condición importante</li>
<li>Otra que también lo es.</li>
<li id="muy-importante">Ésta es muy importante y por eso la diferenciamos del resto.</li>
<li>Ésta vuelve a ser importante pero no tanto como la anterior. </li>
</ul>
<p>Ahora seguimos con el texto normal </p>
</body>
```
- Right side (Rendered Output):** Shows the visual result of the code. It starts with the text "Escribimos una lista de condiciones que consideramos importantes:". Below it is a bulleted list:
 - Primera condición importante
 - Otra que también lo es.
 - Ésta es muy importante y por eso la diferenciamos del resto.
 - Ésta vuelve a ser importante pero no tanto como la anterior.
 The text "Ahora seguimos con el texto normal" follows. A button labeled "Visualizar el formato resultante" is visible at the bottom of the code editor area.

The status bar at the bottom of the browser window shows "Terminado".

Para crear una regla de tipo id en el editor de CSS utilizaremos la opción estilo aplicado a todos los elementos coincidentes con el siguiente selector y escribiremos el nombre empezando por el signo #

Nueva regla de estilo

estilo con nombre (introduzca abajo el nombre de la clase)

Crear un nuevo: estilo aplicado a todos los elementos del tipo (introduzca el tipo abajo)

estilo aplicado a todos los elementos coincidentes con el siguiente selector

nombre del identificador precediéndolo con #

Una vez definidos los selectores **ID** , la aplicación de los mismos habrá que realizarla seleccionando la etiqueta correspondiente al elemento en la barra de estado y pulsando el botón derecho para elegir entre las posibilidades que se muestran en la opción id del menú desplegable.

COMENTARIOS

Puede venir muy bien incluir comentarios que nos sirvan para aclarar o recordar algún aspecto de las especificaciones que escribimos en una hoja de estilo. Para hacerlo lo iniciaremos con `/*` y lo finalizaremos con `*/`

Ej:

```
/* pongo la primera línea del párrafo con ID=muestra en versales de color azul*/  
p#muestra:first-line {color: #0000FF; font-variant: small-caps}
```