

Índice: Tema 2.4 (Parte 1)

2.4 Hojas de estilo en cascada CSS

2.4.1 El lenguaje XHTML

2.4.2 Hojas de estilo CSS

2.4.3 Reglas de estilo

2.4.4 El modelo de cajas

2.4.5 Propiedades de los elementos básicos

Índice: Tema 2.4 (Parte 1)

2.4 Hojas de estilo en cascada CSS

2.4.1 El lenguaje XHTML

2.4.2 Hojas de estilo CSS

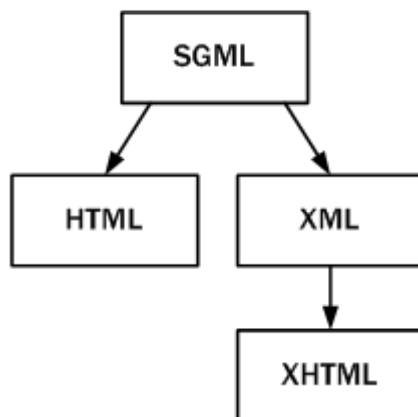
2.4.3 Reglas de estilo

2.4.4 El modelo de cajas

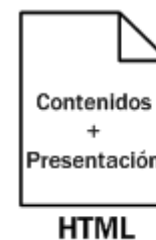
2.4.5 Propiedades de los elementos básicos

➤ Introducción a XHTML (1)

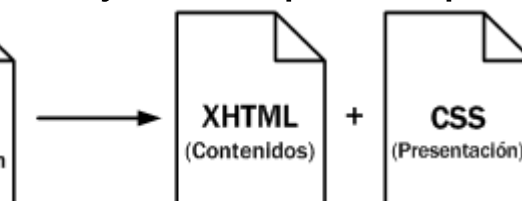
- ✓ El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML.
- ✓ Técnicamente, HTML es descendiente directo del lenguaje SGML, mientras que XHTML lo es del XML (que a su vez, también es descendiente de SGML).
- ✓ CSS es el mecanismo que permite separar los contenidos definidos mediante XHTML y el aspecto que deben presentar esos contenidos:



html compnetes <a>
CSS estilo



javascript comprtamient





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.1 EL LENGUAJE XHTML



➤ Introducción a XHTML (2)

- ✓ XHTML (EXtensible Hypertext Markup Language).
 - <http://www.w3.org/TR/xhtml1/>
- ✓ El objetivo del XHTML es reemplazar a HTML.
- ✓ HTML 4.01 es la última especificación de HTML como tal. A partir de entonces ha evolucionado mediante especificaciones de XHTML.
- ✓ XHTML 1.0 es casi idéntico de HTML 4.01. XHTML es simplemente una versión estricta y clara de HTML.
- ✓ Para que un documento sea XHTML:
 - Tiene que ser XML bien formado.
 - Tiene que ser válido frente a una DTD's proporcionadas por W3C.



➤ Introducción a XHTML (3)

- ✓ Actualmente existen muchos sitios web cuyas páginas no son válidas respecto al estándar HTML. Supóngase por ejemplo, la página siguiente:

```
<html>
<head>
  <title>This is bad HTML</title>
<body>
  <h1>Bad HTML
</body>
```

- ✓ Aunque los navegadores suelen interpretarlas de manera correcta pese a los fallos, es preferible depurarlos mediante un proceso de validación.
- ✓ Actualmente no todos los dispositivos (móviles, PDA, ...) tienen la suficiente potencia como para interpretar páginas HTML mal marcadas.



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.1 EL LENGUAJE XHTML



➤ Introducción a XHTML (4)

- ✓ Hay 3 tipos de documentos XHTML con sus correspondientes **DTDs**:

- ✓ **Strict**: Estricta, sin marcado presentacional (para usar con CSS). a única que se usa

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>
```

- ✓ **Transitional**: Permisiva con algunos aspectos presentacionales.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
```

- ✓ **Frameset**: Con soporte de marcos (ventanas con distintos marcos) .

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd>
```



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.1 EL LENGUAJE XHTML



➤ XHTML bien formado: Diferencias respecto a HTML (1)

- ✓ En el elemento raíz, `<html>`, se suele declarar el espacio de nombres:

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

- ✓ La DTD define como obligatorios la sección `<head>`, y dentro de ella el `<title>`.
- ✓ Todos los elementos XHTML deben tener la etiqueta de cierre, incluso los elementos vacíos:
 - Incorrecto: `
`
 - Correcto: `
</br>` ó `
` ó `
` (Preferiblemente esta última)
- ✓ Los elementos anidados deben tener un correcto orden de apertura/cierre (el que se abre último, debe cerrarse primero):
 - Incorrecto: `<i>Este texto está en negrita y cursiva</i>`
 - Correcto: `<i>Este texto está en negrita y cursiva</i>`



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.1 EL LENGUAJE XHTML



➤ XHTML bien formado: Diferencias respecto a HTML (2)

✓ Los valores de los atributos deben ir encerrados entre comillas (simples o dobles):

- Incorrecto: `<td rowspan=3>`
- Correcto: `<td rowspan='3'>`
- Correcto: `<td rowspan="3">`

✓ No se permiten atributos minimizados (atributos booleanos):

- Incorrecto: `<input checked>`
- Correcto: `<input checked="checked">`

✓ Los nombres de elementos y atributos deben ir en minúsculas.

- Incorrecto: `<P class="nombre">Texto com estilo</P>`
- Incorrecto: `<p CLASS="nombre">Texto com estilo</p>`
- Correcto: `<p class="nombre">Texto com estilo</p>`



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.1 EL LENGUAJE XHTML



➤ XHTML bien formado: Diferencias respecto a HTML (3)

✓ El atributo **id** sustituye a **name**:

- Incorrecto: ``
- Correcto: ``
- Correcto: ``

✓ Cuando se usan parámetros en el atributo **href** es preferible utilizar **&** a **&**:

- Incorrecto: ``
- Correcto: ``

✓ La DTD estricta define además algunas reglas adicionales:

- Desaparecen todos los atributos y elementos presentacionales, por lo que todo el estilo se debe especificar mediante CSS.
- El texto no debe ser insertado directamente en el cuerpo (dentro de la etiqueta `<body>`).



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.1 EL LENGUAJE XHTML



➤ Conversión de HTML a XHTML

✓ Una posibilidad es la conversión **manual**, para ello:

1. Declarar espacio de nombres en el elemento `<html>`:

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

2. Convertir todos los nombres de elementos y atributos a minúsculas, incluir etiquetas de cierre para cada etiqueta de inicio y poner todos los valores de los atributos entre comillas.

3. Asegurarse de que todos los elementos y atributos usados están definidos en la DTD elegida. Si se quiere validar con la DTD estricta, hay que expresar todos los atributos de estilo mediante hojas CSS.

✓ Otra posibilidad es la conversión **automática**:

- Para ello, existen herramientas como HTML Tidy, que además de “limpiar” el código HTML permite realizar una conversión a XHTML.



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.1 EL LENGUAJE XHTML



➤ Validación de un documento XHTML (1)

- ✓ La estructura básica de un documento XHTML es la siguiente:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

  <head>
    <title>Documento sin título</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  </head>

  <body>
    ...
  </body>

</html>
```



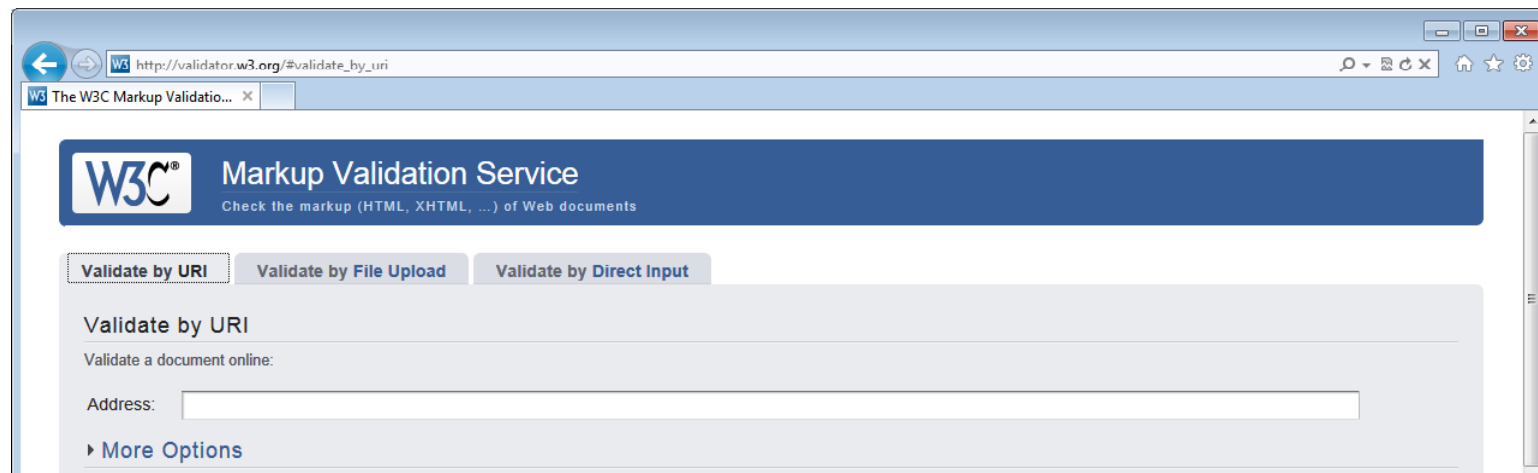
2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.1 EL LENGUAJE XHTML



➤ Validación de un documento XHTML (2)

- ✓ Existen diferentes formas de validar una página creada de acuerdo a una determinada DTD:
 - Algunos editores tienen el validador integrado en la propia herramienta.
 - Algunos navegadores permite instalar un validador como complemento.
 - El organismo W3C posee una herramienta que se puede utilizar gratuitamente a través de Internet desde: <http://validator.w3.org/>





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.1 EL LENGUAJE XHTML

EJERCICIO 32



➤ Validación de un documento XHTML (3)

- ✓ Utilizar el validador online de W3C para corregir los errores del Ejercicio 26.

W3C The W3C Markup Validation... x

W3C® Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI Validate by File Upload **Validate by Direct Input**

Validate by direct input

Enter the Markup to validate:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ejercicio 26</title>
</head>
<body>
<p>Deseo recibir información sobre novedades de:<br>
<input type="checkbox" name="Libros" value="libros" checked=""> Libros<br>
<input type="checkbox" name="Discos" value="discos"> CDs musicales<br>
<input type="checkbox" name="Videos" value="videos" checked=""> Videos
</p>
</body>
```

▶ More Options

Check

This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or [CSS stylesheets](#), [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).

The W3C validators are hosted on server technology donated by HP, and supported by community donations. [Donate](#) and help us build better tools for a better web.

3549
Flattr

100%



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.1 EL LENGUAJE XHTML

EJERCICIO 32



➤ Validación de un documento XHTML (4)

The screenshot shows the W3C Markup Validation Service interface. The browser address bar displays `http://validator.w3.org/check`. The page title is "[Invalid] Markup Validation...". The main heading is "Markup Validation Service" with the subtitle "Check the markup (HTML, XHTML, ...) of Web documents". Below this, there are tabs for "Jump To: Notes and Potential Issues" and "Validation Output". A red banner indicates "Errors found while checking this document as XHTML 1.0 Transitional!". The results section shows "Result: 8 Errors, 1 warning(s)". The "Source" tab is selected, displaying the following code snippet:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ejercicio 26</title>
</head>
<body>
<p>Deseo recibir información sobre novedades de:<br>
<input type="checkbox" name="Libros" value="libros" checked> Libros<br>
<input type="checkbox" name="Discos" value="discos"> CDs musicales<br>
<input type="checkbox" name="Videos" value="videos" checked> Videos
</p>
</body>
```

Below the code, there are fields for "Encoding: utf-8", "Doctype: XHTML 1.0 Transitional", "Root Element: html", and "Root Namespace: <http://www.w3.org/1999/xhtml>". At the bottom, there is a Mozilla logo and text stating "The W3C validators are developed with assistance from the Mozilla Foundation, and supported by community donations. [Donate](#) and help us build better tools for a better web." A "Flattr" button is also visible with the number "3549".



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.1 EL LENGUAJE XHTML

EJERCICIO 32



➤ Validación de un documento XHTML (5)

Validation Output: 8 Errors

✖ Line 8, Column 53: end tag for "br" omitted, but OMITTAG NO was specified

```
<p>Deseo recibir información sobre novedades de:<br>
```

You may have neglected to close an element, or perhaps you meant to "self-close" an element, that is, ending it with "/" instead of ">".

⬅ Line 8, Column 49: start tag was here

```
<p>Deseo recibir información sobre novedades de:<br>
```

✖ Line 9, Column 60: the name and VI delimiter can be omitted from an attribute specification only if SHORTTAG YES is specified

```
<input type="checkbox" name="Libros" value="libros" checked> Libros<br>
```

"VI delimiter" is a technical term for the equal sign. This error message means that the name of an attribute and the equal sign cannot be omitted when specifying an attribute. A common cause for this error message is the use of "Attribute Minimization" in document types where it is not allowed, in [XHTML](#) for instance.

How to fix: For attributes such as compact, checked or selected, do not write e.g <option selected ... but rather <option selected="selected" ...

✖ Line 9, Column 61: end tag for "input" omitted, but OMITTAG NO was specified

```
<input type="checkbox" name="Libros" value="libros" checked> Libros<br>
```

You may have neglected to close an element, or perhaps you meant to "self-close" an element, that is, ending it with "/" instead of ">".

⬅ Line 9, Column 1: start tag was here

Índice: Tema 2.4 (Parte 1)

2.4 Hojas de estilo en cascada CSS

2.4.1 El lenguaje XHTML

2.4.2 Hojas de estilo CSS

2.4.3 Reglas de estilo

2.4.4 El modelo de cajas

2.4.5 Propiedades de los elementos básicos



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Motivación de las hojas de estilo (1)

- ✓ HTML se creó como un lenguaje para la descripción de documentos, sin que la presentación de la información o el diseño gráfico fuera un aspecto relevante.
- ✓ Sin embargo, con la espectacular expansión de la WWW, ésta pasó a convertirse en un medio de comunicación de masas y el aspecto de los documentos (páginas web) pasó a ser un aspecto importante.
- ✓ Aunque HTML ofrece recursos para crear presentaciones con estilo (dar un determinado formato al contenido presentado) usando elementos y atributos relativos a la presentación de la información, esto no es lo recomendable, sobre todo en webs complejas y de gran tamaño.
- ✓ Piénsese, por ejemplo, en un sitio web con 100 páginas en las que se decide cambiar el tipo de letra, su tamaño, color, etc... Habría que ir modificando cada uno de los atributos de las numerosas etiquetas `` uno a uno.



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Motivación de las hojas de estilo (2)

- ✓ Así, lo recomendable es que las instrucciones relativas al formato o estilo se pongan en un archivo aparte que se conoce como **hoja de estilo**.
- ✓ Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas: obliga a crear documentos HTML/XHTML bien definidos, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento, etc...
- ✓ Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, tabla, lista, formulario, etc...
- ✓ Una vez creados los contenidos, se utiliza una hoja de estilo para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc...



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Hojas de estilo (1)

- ✓ Las **hojas de estilo** permiten un control absoluto sobre la presentación de la información: interlineado, sangrías, colores, tamaños, estilos, posicionamiento, etc...
- ✓ Permiten separar de forma efectiva la **descripción** del contenido (**documento XHTML**) y su forma de **presentación** (**hoja de estilo CSS**).
- ✓ Formas de asociar información de estilo en XHTML:
 - Directamente en los elementos del cuerpo del documento (atributo **style**).
 - En la cabecera del propio documento (elemento **<style>**).
 - En un archivo externo vinculado al documento (hoja de estilo).
- ✓ Se pueden utilizar los tres métodos al mismo tiempo.



➤ Hojas de estilo (2)

- ✓ Cada método tiene sus implicaciones:
 - Con el atributo **style** se define el estilo de un elemento particular del documento.
 - Con el elemento **<style>** se definen estilos para los diferentes elementos elemento (**<h1>** , **<p>** , etc...) del documento.
 - El elemento **<style>** es parte del documento, mientras que la hoja de estilo externa tan solo se referencia desde el documento. Así, la misma hoja puede ser referenciada desde diferentes documentos.
 - Además, se puede disponer de varias hojas de estilo diferentes y cambiar el aspecto del documento vinculando otra hoja de estilo diferente.
- ✓ Aunque los tres métodos son válidos, la hoja de estilo externa es el método que más independencia proporciona al contenido del documento de las características de estilo a aplicar.



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Estilo predeterminado (1)

```
<html>
<head>
  <title>Estilos HTML</title>
</head>
<body>
  <p>La referencia de HTML es un archivo HLP de Windows que describe
  toda la sintaxis actual de HTML. Está disponible en los siguientes
  formatos:
  </p>
  <ul>
    <li>Windows 3.x</li>
    <li>Windows 95/NT</li>
  </ul>
  <p>Para más información sobre HTMLib, contacte con
    <a href="mailto:htmlib@htmlib.demon.co.uk">htmlib@htmlib.demon.co.uk</a>
  </p>
</body>
</html>
```



➤ Estilo predeterminado (2)

- ✓ Sin información de estilo en el documento, el texto se presenta utilizando los estilos predeterminados que están establecidos para cada tipo de elemento:

La referencia de HTML es un archivo HLP de Windows que describe toda la sintaxis actual de HTML. Está disponible en los siguientes formatos:

- Windows 3.x
- Windows 95/NT

Para más información sobre HTMLib, contacte con
htmlib@htmlib.demon.co.uk

- ✓ Los estilos predeterminados producen presentaciones bastante pobres y definen características de formato mínimas.



➤ Estilo en el cuerpo (1)

- ✓ Casi todos los elementos HTML contemplan el uso del atributo **style**.
- ✓ Aunque luego hablaremos con más detalle de la sintaxis de CSS, de momento basta saber que las características de formato se expresan con pares **propiedad-valor**.
- ✓ El nombre de la propiedad va seguido de dos puntos (:) y del valor que se le asigna, y los distintos pares propiedad-valor se separan con punto y coma (;).
- ✓ La secuencia de pares propiedad-valor se encierra entre comillas y se asigna al atributo **style** del documento:

```
<p style="font-family:Arial; font-size:12pt; color:rgb(0,128,0)">
```

La referencia de HTML es un archivo HLP de Windows que describe toda la sintaxis actual de HTML. Está disponible en los siguientes formatos:

```
</p>
```



➤ Estilo en el cuerpo (2)

- ✓ Las características de formato especificadas con el atributo **style** sólo afectan al elemento en el que están.
- ✓ En nuestro ejemplo, se ha definido un estilo para el primer elemento **<p>** del cuerpo del documento, afectando sólo a ese elemento y a ninguno otro, de forma que el otro elemento **<p>** se sigue viendo igual:

La referencia de HTML es un archivo HLP de Windows que describe toda la sintaxis actual de HTML. Está disponible en los siguientes formatos:

- Windows 3.x
- Windows 95/NT

Para más información sobre HTMLib, contacte con
htmlib@htmlib.demon.co.uk



➤ Estilo en la cabecera

- ✓ En la cabecera del documento se puede colocar un elemento `<style>` para definir el formato de todos los elementos del cuerpo de determinados tipos:

```
<head>
```

```
  <title>Estilos HTML</title>
```

```
  <style type="text/css">
```

```
    p {font-family:Arial; font-size:12pt; color:rgb(0,128,0)} importante
```

```
  </style>
```

```
</head>
```

Elementos P

La referencia de HTML es un archivo HLP de Windows que describe toda la sintaxis actual de HTML. Está disponible en los siguientes formatos:

- Windows 3.x
- Windows 95/NT

Para más información sobre HTMLib, contacte con htmlib@htmlib.demon.co.uk

regla de estilo a e

selector parraf table..{ regla 1 :valor 1; regla 2 :valor 2;}



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS

EJERCICIO 35



➤ Estilo en una hoja externa (1)

- ✓ El método más versátil de aplicación de características de formato al contenido de un documento HTML se encuentra en el uso de hojas de estilo externas que se vinculan con el documento HTML.
- ✓ El contenido del archivo de hoja de estilo es el mismo conjunto de reglas que se definen dentro del elemento **<style>**.
- ✓ Creamos una hoja de estilo en un archivo llamado **ejercicio35.css** (preferiblemente dentro de una carpeta llamada "css").
- ✓ Su contenido será el mismo estilo que hemos definido en los dos casos anteriores:

ejercicio35.css

```
p {font-family:Arial; font-size:12pt; color:rgb(0,128,0) }
```



➤ Estilo en una hoja externa (2)

- ✓ Una hoja de estilo se vincula con un documento HTML por medio de un elemento

`<link>` en la cabecera del documento:

ubicacion del archiv dnd estan las reglas de esti

`<head>`

`<title>Estilos HTML</title>`

`<link href="css/ejercicio35.css" rel="stylesheet" type="text/css" media="screen"/>`

`</head>`

crear una carpeta css ejercici

- ✓ Atributos del elemento `<link>`:

- **href** Nombre y ubicación del recurso vinculado (absoluta/relativa, interno/externo).
- **rel** Tipo de relación (en el caso de hoja de estilo siempre a **stylesheet**).
- **type** Tipo del recurso que se vincula (para hojas de estilo siempre a **text/css**).
- **media** Medio en el que se van a aplicar los estilos.



➤ Estilo en una hoja externa (3)

- ✓ Una forma alternativa para incluir en la página HTML los estilos definidos en archivos CSS externos es utilizando una regla especial de tipo **@import** dentro del elemento **<style>**.
- ✓ Las reglas de tipo **@import** siempre preceden a cualquier otra regla CSS (con la única excepción de la regla **@charset** para la codificación de caracteres). Van al comienzo de la hoja de estilo, antes de definir ninguna regla con atributos.
- ✓ La URL del archivo CSS externo se indica mediante una cadena de texto encerrada con comillas simples o dobles o mediante la palabra reservada **url()**. De esta forma, las siguientes reglas **@import** son equivalentes:

@import 'css/ejercicio35.css';	<style type = "text/css">
@import "css/ejercicio35.css";	
@import url('css/ejercicio35.css');	@import...
@import url("css/ejercicio35.css");	</style>



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



- **Interpretación del estilo (1): La cascada** no entra
 - ✓ Dado que se pueden utilizar conjuntamente los tres métodos de definición de estilos debemos saber cuál se aplica en cada caso.
 - A un determinado elemento del documento se le aplicará en principio el estilo que esté definido para ese tipo de elemento en la hoja de estilo externa vinculada.
 - Si en la cabecera del documento hay características de estilo para ese tipo de elemento, el estilo que se aplicará será el indicado en la hoja de estilo externa modificado con la información de estilo del elemento **<style>**.
 - Si para un elemento individual de ese tipo se hace uso del atributo **style**, el estilo que se aplicará será el indicado en la hoja de estilo externa, modificado con la información de estilo del elemento **<style>**, y modificado con la información del atributo **style**.
 - ✓ Por tanto, todas las definiciones se combinan en **cascada** para producir una única hoja de estilo en la que las definiciones mas locales tienen prioridad.



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



- **Interpretación del estilo (2): Conflictos de estilos** no entra
- ✓ En las hojas de estilos complejas, es habitual que varias reglas CSS se apliquen a un mismo elemento HTML originando conflictos de estilo (colisiones).
 - ✓ CSS tiene un mecanismo de resolución de colisiones muy complejo y que tiene en cuenta varios aspectos:
 - El tipo de hoja de estilo que se trate (de navegador, de usuario o de diseñador).
 - La importancia de cada regla.
 - Lo específico que sea el selector.
 - ✓ Resolución de colisiones de :
 - Si tenemos varias reglas que afectan a los mismos elementos, para resolver los conflictos se asigna un peso a dichas reglas.
 - Gana la que tiene mayor peso.



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



- **Interpretación del estilo (3): Resolución de colisiones** no entra
- ✓ Determinar todas las declaraciones que se aplican sobre un determinado elemento para el medio CSS seleccionado. Se pueden dar dos casos:
 - ✓ En caso de **no existir reglas** se usa el valor heredado, y si no existe se usa el valor por defecto.
 - ✓ En caso de **existir reglas** se ordenan de acuerdo a un peso y se selecciona la de mayor peso. Los criterios para dicha ordenación son:
 - Se asigna un peso según el origen de la regla (del navegador, de usuario o de diseñador).
 - Se asigna mayor peso a las reglas marcadas con **!important**.

```
selector { atributo:valor !important; atributo:valor; }
```

 - Se asigna mayor peso a las reglas que tengan selectores sean más específicos.
 - Si dos reglas tienen el mismo peso, al última en definirse gana.



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Interpretación del estilo (4): Prioridad de los selectores no entra

✓ La prioridad de los selectores se determina de la forma siguiente:

- **a** = Número de atributos **id**.
- **b** = Número de atributos **class** en el selector.
- **c** = Número de etiquetas en el selector.
- Concatenar estos 3 valores y ésta es la prioridad: "**abc**".

```
li          {...} /* a=0 b=0 c=1 -> specificity = 1 */
ul li       {...} /* a=0 b=0 c=2 -> specificity = 2 */
ul ol li    {...} /* a=0 b=0 c=3 -> specificity = 3 */
li.red      {...} /* a=0 b=1 c=1 -> specificity = 11 */
ul ol li.red {...} /* a=0 b=1 c=3 -> specificity = 13 */
#x34y       {...} /* a=1 b=0 c=0 -> specificity = 100 */
```

- Las reglas definidas en atributo **style** tienen la misma prioridad que reglas que usan selectores **id**.



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Tipos de dispositivo (1)

- ✓ Las hojas de estilos CSS permiten definir diferentes estilos para diferentes medios o dispositivos: pantallas, impresoras, móviles, proyectores, etc...

Medio	Descripción
all	Todos los medios definidos
braille	Dispositivos táctiles que emplean el sistema braille
embosed	Impresoras braille
handheld	Dispositivos de mano: móviles, PDA, etc.
print	Impresoras y navegadores en el modo <i>"Vista Previa para Imprimir"</i>
projection	Proyectores y dispositivos para presentaciones
screen	Pantallas de ordenador
speech	Sintetizadores para navegadores de voz utilizados por personas discapacitadas
tty	Dispositivos textuales limitados como teletipos y terminales de texto
tv	Televisores y dispositivos con resolución baja



➤ Tipos de dispositivo (2)

- ✓ **Medios definidos con las reglas de tipo @media:** Son un tipo especial de regla CSS que permiten indicar de forma directa el medio o medios en los que se aplicarán los estilos incluidos en la regla:

```
@media print {  
    body { font-size:10pt }  
}  
  
@media screen {  
    body { font-size:13px }  
}  
  
@media screen, print {  
    body { line-height:1.2 }  
}
```



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Tipos de dispositivo (3)

- ✓ **Medios definidos con las reglas de tipo @import:** Cuando se utilizan reglas de tipo `@import` para enlazar archivos CSS externos, se puede especificar el medio en el que se aplican los estilos indicando el nombre del medio después de la URL del archivo CSS:

```
@import url("css/estilos_pantalla.css") screen;  
@import url("css/estilos_impresora.css") print;
```

- ✓ **Medios definidos con la etiqueta <link>:** Si se utiliza la etiqueta `<link>` para enlazar los archivos CSS externos, se puede utilizar el atributo `media` para indicar el medio o medios en los que se aplican los estilos de cada archivo:

```
<link href="css/estilos_pantalla.css" rel="stylesheet"  
      type="text/css" media="screen"/>
```



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Propiedades y valores

si

- ✓ Las propiedades definen cada uno de los aspectos de un elemento a los que se puede aplicar un determinado estilo para darle formato.
- ✓ A cada propiedad se le puede asignar un valor dentro de su conjunto de valores válidos. Así, podemos tener:

- Identificadores específicos: **blue, auto, Arial, center**
- Números (enteros y reales): **10, 12.1**
- URLs: **url(http://www.bg.com/pin.gif)**

- ✓ Las propiedades se pueden resumir. Ejemplo:

Propiedad resumido

```
h1 { font: bold 12pt/14pt Helvetica }
```



Propiedades de texto

```
h1 {  
  font-weight: bold;  
  font-size: 12pt;  
  line-height: 14pt;  
  font-family: Helvetica;  
  font-variant: normal;  
  font-style: normal;  
}
```



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Unidades de medida (1)

- ✓ El valor de muchas propiedades suele ser un número (entero o real) seguido por la unidad de medida. Ejemplos: **10px**, **12pt**, **1.5mm**
- ✓ Existen dos tipos de unidades de medidas: **Absolutas** y **relativas**.
- ✓ **Absolutas**: Son útiles sólo cuando se conocen propiedades del medio (por ejemplo el navegador, la impresora, etc...):
 - **in** Del inglés "inches", pulgadas (1 pulgada son 2.54 centímetros).
 - **cm** Centímetros.
 - **mm** Milímetros.
 - **pt** Puntos (1 punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros).
 - **pc** Picas (1 pica equivale a 12 puntos, es decir, unos 4.23 milímetros).



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Unidades de medida (2)

- ✓ **Relativas:** Representan una longitud respecto a otra medida de longitud. El valor del que se depende, es particular a cada propiedad. Valores permitidos:
 - **em:** Relativo al tamaño de fuente (**margin: 0.5em**).
 - **ex:** Relativo a la altura de la letra x minúscula (**margin: 1ex**).
 - **px:** Relativo al tamaño de pixel (**font-size: 12px**).
- ✓ Las unidades **em** y **ex** son relativas al tamaño de letra dentro del propio elemento, excepto para la propiedad **font-size** para la que se toma el tamaño de letra del padre.
- ✓ Las unidades relativas se heredan como valores calculados. Así, en el siguiente ejemplo, el valor de la propiedad **text-indent** para **<h1>** es **36px**:

```
body { font-size:12px; text-indent:3em; }  
h1 { font-size:15px }
```



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Porcentajes

- ✓ Un **porcentaje** está formado por un valor numérico seguido del símbolo % y siempre está referenciado a otra medida.
- ✓ Cada una de las propiedades de CSS que permiten indicar como valor un porcentaje, define el valor al que hace referencia ese porcentaje.
- ✓ Los porcentajes también se heredan como valores calculados.
- ✓ En el siguiente ejemplo, el tamaño de la fuente para el elemento `<h1>` es de **20px** mientras que para el elemento `<h2>` es de **15px**.

```
body{ font-size:10px }
```

```
h1 { font-size:200% }
```

```
h2 { font-size:150% }
```



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.2 HOJAS DE ESTILO CSS



➤ Colores

- ✓ Los **colores** en CSS se pueden indicar de cinco formas diferentes:
- ✓ Palabras clave: **acua, black, blue, ..., white, yellow**
- ✓ Colores del sistema: **ActiveBorder, Background, ..., WindowText**

- ✓ RGB (cada componente con 1 byte):

- Hexadecimal: **#rrggbb**
- Decimal: **rgb (r, g, b)**
- Porcentual: **rgb (r%, g%, b%)**

maroon #800000	red #ff0000	orange #ffa500	yellow #ffff00	olive #808000
purple #800080	fuchsia #ff00ff	white #ffffff	lime #00ff00	green #008000
navy #000080	blue #0000ff	aqua #00ffff	teal #008080	
black #000000	silver #c0c0c0	gray #808080		

Índice: Tema 2.4 (Parte 1)

2.4 Hojas de estilo en cascada CSS

2.4.1 El lenguaje XHTML

2.4.2 Hojas de estilo CSS

2.4.3 Reglas de estilo

2.4.4 El modelo de cajas

2.4.5 Propiedades de los elementos básicos



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.3 REGLAS DE ESTILO



➤ Reglas de estilo CSS (1)

- ✓ Una regla de estilo se compone de dos partes: **selector** y **declaración**.
- ✓ `h1 {color:red}`: `h1` es el selector y `color:red` es la declaración, que va encerrada entre llaves.
- ✓ El selector puede ser cualquier tipo de elemento HTML y actúa como enlace entre el documento HTML y la información de la hoja de estilo.
- ✓ La declaración tiene dos partes: **propiedad** (color) y **valor** (red), separados por dos puntos (:).
- ✓ Para simplificar se pueden agrupar los selectores en listas separadas por comas y las declaraciones separadas por punto y coma.

```
h1, h2, h3 {color:red; font-size:12pt}
```



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.3 REGLAS DE ESTILO



➤ Reglas de estilo CSS (2)

- ✓ Las propiedades se **heredan**: cuando un elemento (hijo) se encuentra dentro de otro (padre), aquel (el hijo) hereda las propiedades de formato de éste (el padre).
- ✓ Como todos los elementos del cuerpo del documento se encuentran dentro (descienden) del elemento `<body>`, éste resulta ideal para establecer las características de estilo predeterminadas.
- ✓ Supóngase la regla `h1{color:red;}` aplicada sobre el fragmento de código:

`<h1>Un documento con palabra enfaticada</h1>`

Entonces, la palabra enfatizada aparecerá enfatizada y en rojo.
- ✓ Si existe una regla del tipo `em{color:blue;}` entonces se mostrara en azul, ya que es una regla mas especifica y que prevalece sobre la herencia.
- ✓ Algunas propiedades no son heredadas, por ejemplo, **background**, etc...



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.3 REGLAS DE ESTILO



➤ Selectores básicos (1)

- ✓ **Selector universal:** Selecciona todos los elementos de la página:

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

- ✓ **Selector de etiqueta:** Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector. Los estilos comunes se pueden agrupar en una única regla:

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

```
p, h1, h2 {propiedad:valor; ...}
```

- ✓ **Selector descendente:** Selecciona los elementos que se encuentran dentro de otros elementos. En el siguiente ejemplo, el estilo indicado se aplicaría a los **em** dentro de un **p** (puede haber un mayor nivel de profundidad e1 e2 ... eN {...}):

```
p em {propiedad:valor; ...}
```



➤ Selectores básicos (2)

- ✓ **Selector de clase:** Selecciona los elementos con el atributo **class** especificado como selector (prefijado con un punto para evitar confusiones con otros):

```
<p class="nombre">Texto con estilo</p>
```

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

Permite aplicar el mismo estilo a elementos diferentes (con **class** común):

```
<h1 class="nombre">Encabezado con estilo</h1>
```

```
<p class="nombre">Texto con estilo</p>
```

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

Si se especifica una etiqueta delante del punto solo se aplica la regla a dichas etiquetas que contengan el atributo **class** correspondiente:

```
p.nombre {propiedad:valor; ...}
```



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.3 REGLAS DE ESTILO



➤ Selectores básicos (3)

- ✓ **Las etiquetas DIV y SPAN:** HTML contempla dos elementos genéricos que no tienen significado concreto que afecte a su presentación. Sirven para organizar el contenido a nivel de bloque (`<div>`) o a nivel de línea (``).
- ✓ Los elementos `<div>` y `` se usan mucho para organizar la presentación de los documentos HTML mediante el uso de selectores de clase:

`<div>`Este es un elemento de ``nivel de bloque`.</div>`

- ✓ Y en la hoja de estilo:

```
div {font-family:Comic Sans MS; font-size:14pt}
span.Cursiva{font-style:oblique}
```

Este es un elemento de *nivel de bloque*.



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.3 REGLAS DE ESTILO



➤ Selectores básicos (4)

- ✓ **Selector de ID:** Similar al selector de clase pero utilizando el atributo **id**. Dicho atributo no se puede repetir en dos elementos diferentes de una misma página. La sintaxis similar a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (**#**) en vez del punto (.) como prefijo del nombre de la regla CSS:

```
<p id="nombre">Texto com estilo</p>
```

```
#nombre {propiedad:valor; ...}
```

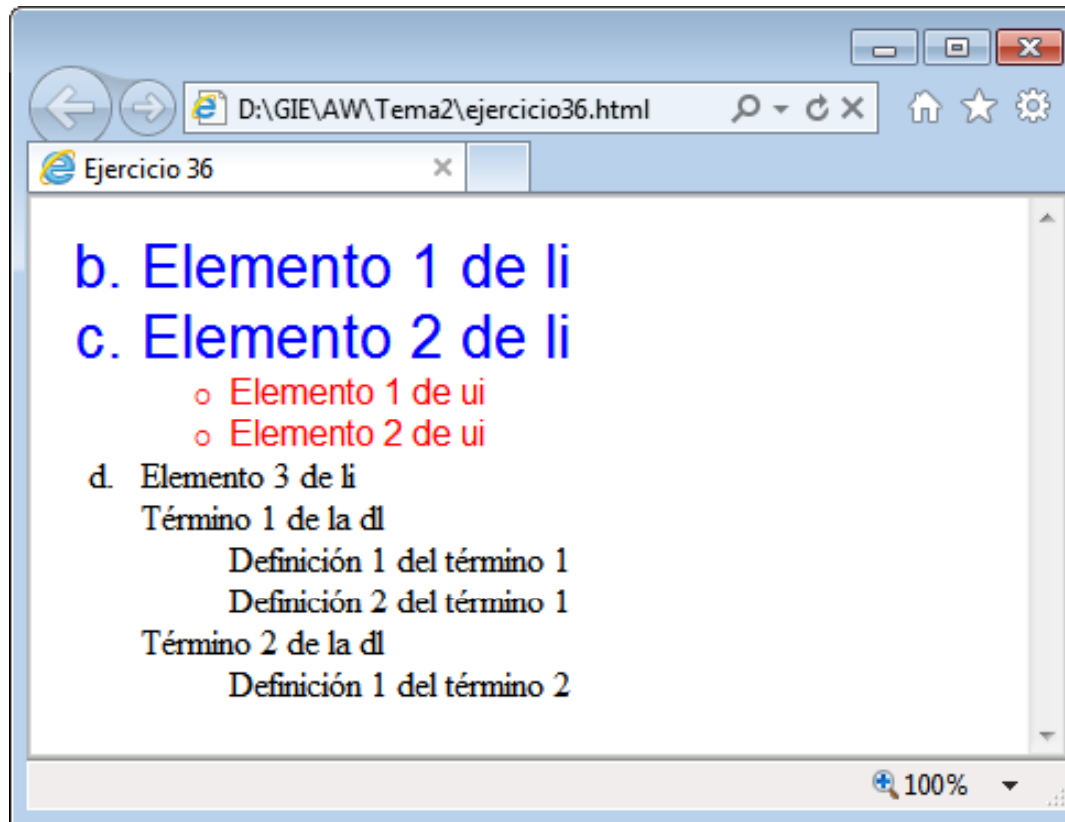
- ✓ Al igual que los selectores de clase, en este caso también se puede restringir el alcance del selector mediante la combinación con otros selectores (**¿absurdo?**):

```
p#nombre {propiedad:valor; ...}
```

- ✓ La recomendación general es la de utilizar el selector de **id** cuando se quiere aplicar un estilo a un solo elemento específico de la página y utilizar el selector de clase cuando se quiere aplicar un estilo a varios elementos diferentes de la página.

➤ Selectores básicos (5)

- ✓ Utilizar selectores básicos para generar las reglas con las que dar el siguiente formato a las listas del Ejercicio 15:





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.3 REGLAS DE ESTILO



➤ Selectores avanzados (1)

- ✓ **Selector de hijos:** Selecciona un elemento que es hijo directo de otro elemento y se indica mediante el signo `>`. En el siguiente ejemplo, el estilo se aplicaría al texto del primer `` pero no al del segundo:

```
p > span {color:blue;}
```

```
<p><span>Texto1</span></p>
```

```
<p><a href="#"><span>Texto2</span></a></p>
```

- ✓ **Selector adyacente:** Selecciona todos los elementos de tipo `elemento2` que cumplan las dos condiciones siguientes : a) `elemento1` y `elemento2` deben ser hermanos (su elemento padre debe ser el mismo); y b) `elemento2` debe aparecer inmediatamente después de `elemento1` en el código de la página:

```
elemento1 + elemento2 {propiedad:valor; ...}
```

Utilidad (primer párrafo diferente al resto): `p + p {propiedad:valor; ...}`



➤ Selectores avanzados (2)

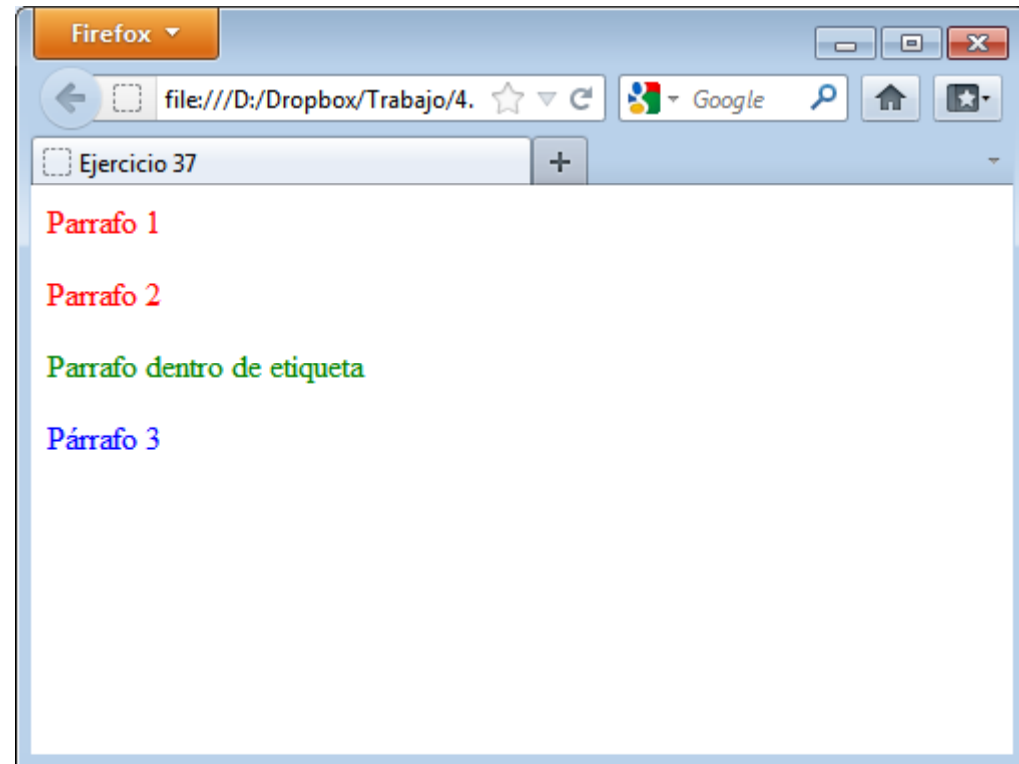
✓ **Selector de atributos:** Selecciona elementos en función de sus atributos y/o valores de éstos. Los cuatro tipos de selectores de atributos son:

- `elemento[atributo] {propiedad:valor; ...}`: Selecciona los elementos que tienen establecido el atributo especificado, independientemente de su valor.
- `elemento[atributo=valor] {propiedad:valor; ...}`: Selecciona los elementos que tienen establecido el atributo especificado con un determinado valor.
- `elemento[atributo~=valor] {propiedad:valor; ...}`: Selecciona los elementos que tienen establecido el atributo especificado con al menos uno de sus valores igual al valor especificado en la regla. Por ejemplo en el título de una imagen.
- `elemento[atributo|=valor] {propiedad:valor; ...}`: Selecciona los elementos que tienen establecido el atributo especificado y cuyo valor es una serie de palabras separadas con guiones, pero que comienza con el valor especificado en la regla. Normalmente útil para los atributos de tipo lang.



➤ Selectores avanzados (3)

- ✓ Utilizar selectores avanzados para generar las reglas con las que dar formato a los siguientes párrafos:
 - Los párrafos hijos de un **body** deben ir en rojo.
 - Pero si están dentro de una etiqueta **div** deben ir en verde.
 - Y si están a continuación de un **div** como hermano en azul.





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.3 REGLAS DE ESTILO



➤ Pseudo-selectores (1)

✓ **Pseudo-clases:** Son clases "virtuales" que nosotros no insertamos de manera explícita en nuestras páginas. Por ejemplo, para enlaces, tenemos:

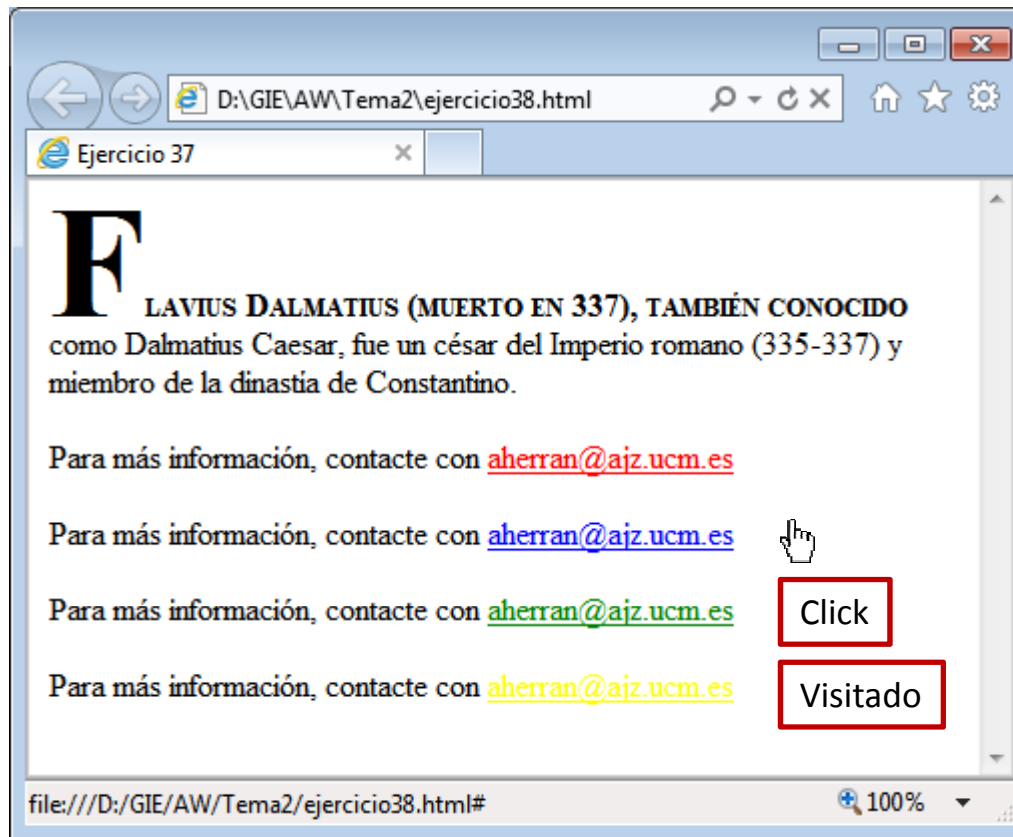
- **a:link:** Estado del enlace cuando aún no ha sido visitado.
- **a:hover:** Estado del enlace cuando se coloca el ratón sobre el mismo.
- **a:active:** Estado del enlace cuando se hace click sobre el mismo.
- **a:visited:** Estado del enlace tras ser visitado por el usuario.

✓ **Pseudo-elementos:** Permiten aplicar estilos a ciertas partes de un texto. Por ejemplo, la primera letra de una palabra o la primera línea de un párrafo.

- **e:first-letter:** Primera letra en un elemento de bloque **e** como `<p>` o `<h1>`.
- **e:first-line:** Primera línea en un elemento de bloque **e** como `<p>` o `<h1>`.

➤ Pseudo-selectores (2)

- ✓ Utilizar pseudo-selectores generar las reglas con las que dar formato a los siguientes párrafos y enlaces:





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.3 REGLAS DE ESTILO



➤ Resumen de selectores

Patrón	Significado	Ejemplo
*	Encaja con cualquier elemento	* { }
E	Encaja cualquier etiqueta E	p { }
E F	Encaja cualquier etiqueta F descendiente de E	div p { }
E > F	Encaja cualquier etiqueta F hija de E	body > h1 { }
E:first-child	Encaja la etiqueta E que sea el primer hijo de su padre	div:first-child { }
E:link E:visited	Encaja cualquier enlace que aún no ha sido visitado, o que ya lo ha sido	a:link{ }
E:active E:hover E:focus	Encaja cuando la etiqueta E está siendo activada, el cursos pasar encima de ella o ha adquirido el foco	div:active{ }
E + F	Encaja cualquier etiqueta F que tiene como hermana a una etiqueta E que aparece justo antes	div + h1 { }
E[foo]	Encaja cualquier elemento E que se ha establecido el atributo foo	div[name]
E[foo="warning"]	Encaja cualquier etiqueta E que se ha establecido el atributo foo a warning	span[class="warning"]
DIV.warning	Cualquier elemento con valor warning en el atributo class	.titulo{ }
E#mi_id	Encaja con cualquier elemento con id mi_id	#float{ }