

Arquitecturas Distribuidas

Tema 3. II. Hojas de estilo en cascada (CSS)

II. Hojas de estilo en cascada (CSS)

1. Introducción
2. Declaración de CSS
 - 2.1. Propiedades típicas en CSS
 - 2.2. Inserción de CSS en HTML
3. Herencia de propiedades en CSS
4. Declaración avanzada de CSS
 - 4.1. Selectores
 - 4.2. Regla de agrupación
 - 4.3. Modelo de cajas
 - 4.4. Posicionamiento
5. Conclusiones

Hojas de estilo en cascada

- *Cascading Style Sheets* (CSS)
- Lenguaje de descripción de estilos que permite especificar el formato de presentación de los datos de documentos estructurados.
- El objetivo es la separación del contenido de la presentación: los datos son independientes de plataforma, vendedor y dispositivo.
- Simplifican el desarrollo de páginas HTML (y otros documentos estructurados como XML) y facilitan su mantenimiento.

Hojas de estilo en cascada: ventajas

- Mejora la eficacia de transmisión:
 - Elimina la necesidad de imágenes invisibles o que sólo contienen letras (usadas antes de la llegada de las CSS para ajustar el contenido)
 - La información de estilo sólo se envía una vez, el navegador la almacena en caché.
- Puede definirse una hoja de estilo para cada medio destino: navegador gráfico, impresora, móvil, navegador por voz, navegador *braille*, etc.

Declaración de CSS

- Una hoja de estilo CSS es un documento de texto que contiene una serie de atributos relativos al formato de presentación a aplicar a los elementos de un documento estructurado asociado.
- Cada reglas (selector + declaraciones) determina las propiedades del elemento:

```
SELECTOR { propiedad_1: valor_1;  
          ...  
          propiedad_n: valor_n }
```

¿A qué
elementos
afecta?

¿Qué valor
tienen las
propiedades?

Declaración de CSS

- Ejemplo de CSS simple:

```
H1 {color: green}
```

Afecta a los encabezados
de primer nivel: H1

Asigna a la
propiedad
“color” el valor
“verde”

Declaración de CSS

- Otro ejemplo de CSS simple:

```
H1 {color: green; font-style: italic}
```

Afecta a los encabezados de primer nivel: H1

Asigna a la propiedad “color” el valor “verde”, y al “estilo de fuente” el valor “cursivas”

Propiedades típicas CSS

- Propiedades:
 - Fuentes
 - Colores
 - Márgenes
 - Alineados
 - Sombreados
 - Dirección del texto
 - *Pitch* de la voz
 - Azimuth
 - etc.

CSS₁ → 50 propiedades

CSS₂ → 120 propiedades!!

Inserción de CSS en HTML

Interno a un documento

```
<STYLE type="text/css">  
  Reglas  
</STYLE>
```

Indicación de la hoja de
estilo que se usará

Externo a un documento

```
<LINK rel="stylesheet" href=URL type="text/css">
```

URL contiene
fichero con las
reglas

Inserción de CSS en HTML

Como atributo de cada elemento (*inline*)

```
<P style="font-size: 20pt"> texto </P>
```

Existen diferentes especificaciones de estilo en cascada: CSS₁, CSS₂.

Herencia de propiedades en CSS

- Se denominan hojas en “cascada” ya que los elementos hijos heredan por defecto los valores de estilo de los padres.
- Las propiedades de un elemento se asignan por valores específicos, heredados o por defecto (de mayor a menor preferencia, respectivamente).

```
<html>
<title>Ejemplo CSS</title>

<body>
<h1> Compositor cl&acute;sico </h1>

<p> Johann Sebastian Bach fue un compositor
<strong> prol&acute;fico. </strong> Entre sus
obras destacan: </p>

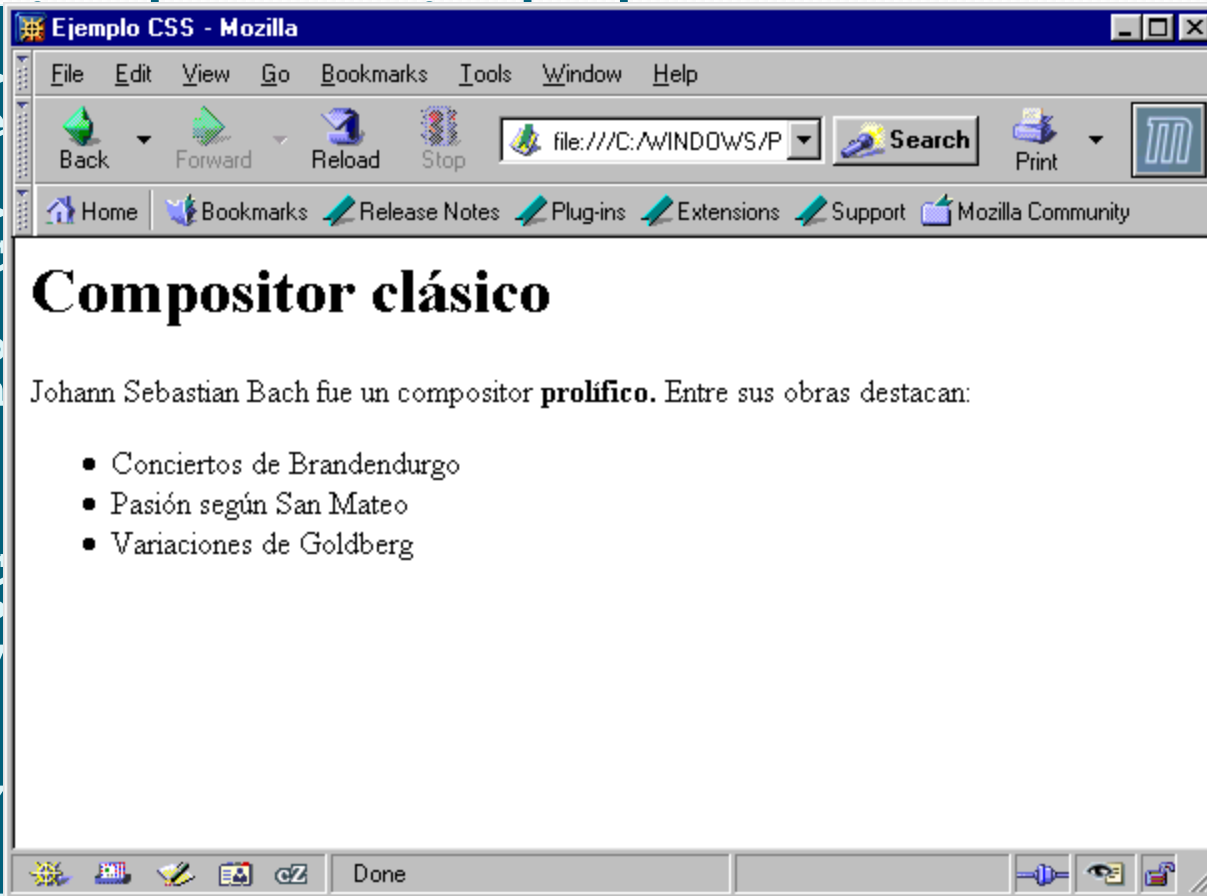
<ul>
<li> Conciertos de Brandeburgo
<li> Pasión según San Mateo
<li> Variaciones de Goldberg
</ul>

</body>

</html>
```

ESTILO POR DEFECTO

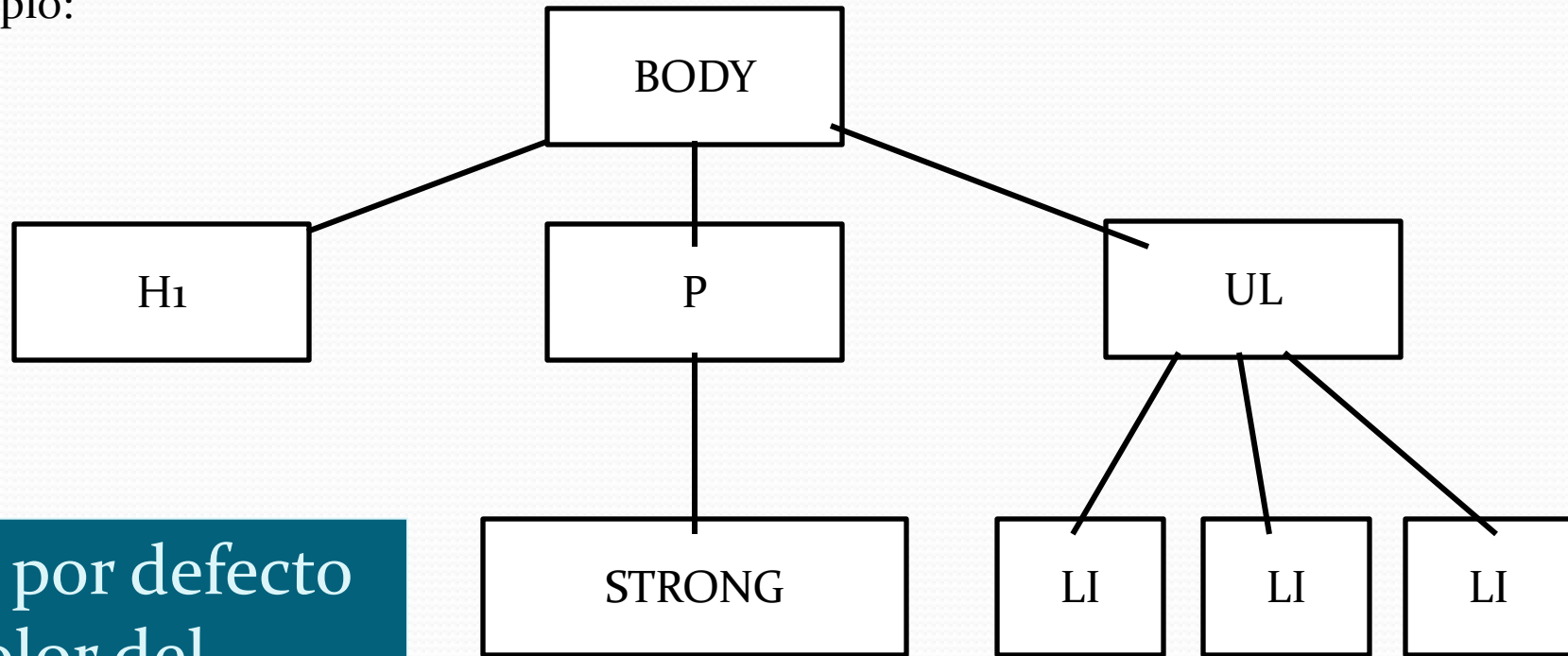
```
<html>
<title>
<body>
<h1> Compositor clásico
<p> Johann Sebastian Bach fue un compositor prolífico. Entre sus obras destacan:
<strong>
obras
<ul>
<li> Conciertos de Brandenburgo
<li> Pasión según San Mateo
<li> Variaciones de Goldberg
</ul>
</body>
</html>
```



us

Herencia de propiedades en CSS

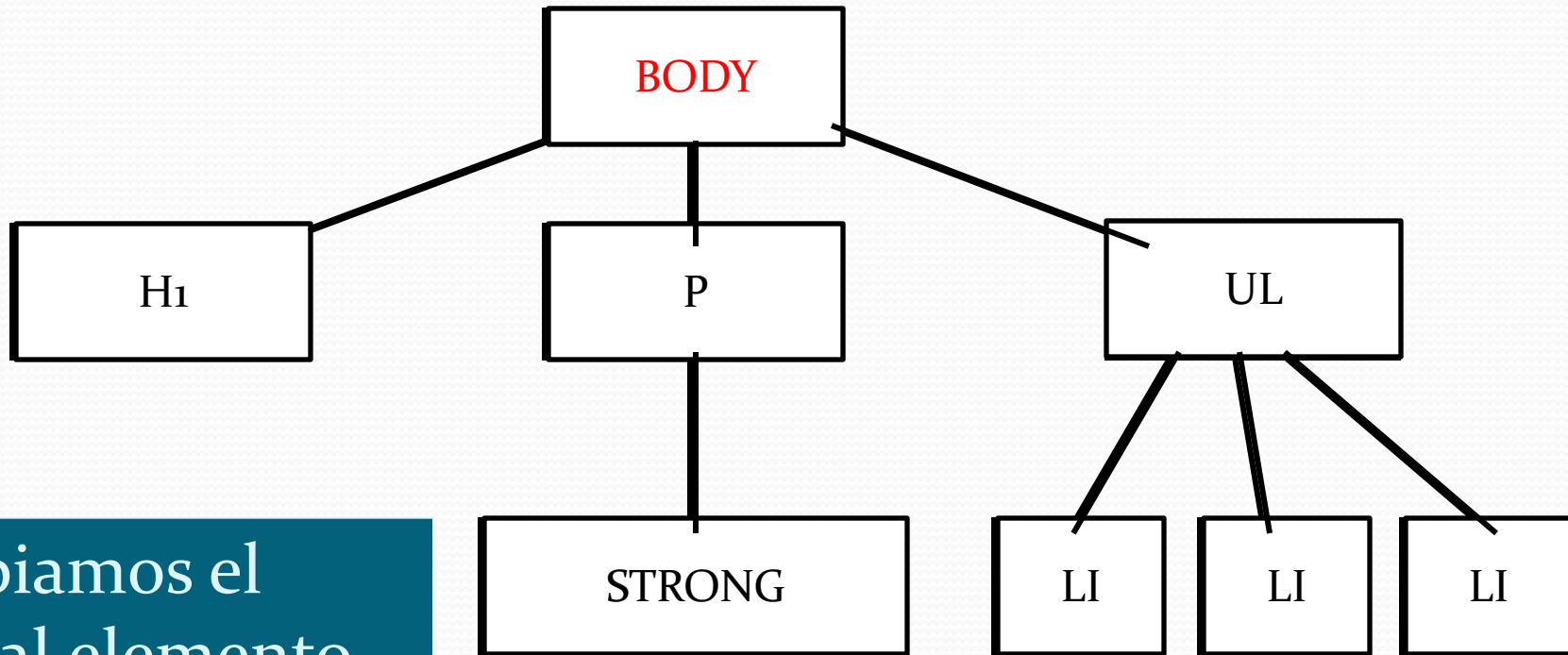
Ejemplo:



Valor por defecto
del color del
elemento: **negro.**

Herencia de propiedades en CSS

Ejemplo:



Cambiamos el
valor al elemento
BODY: **rojo**

```

<html>
<title>Ejemplo CSS</title>

<style>
    body {color: red}
</style>

<body>
<h1> Compositor cl&acute;sico </h1>

<p> Johann Sebastian Bach fue un compositor
<strong> prol&iacute;fico. </strong> Entre sus
obras destacan: </p>

<ul>
<li> Conciertos de Brandendurgo
<li> Pasión según San Mateo
<li> Variaciones de Goldberg
</ul>

</body>

</html>

```

CAMBIO EN ESTILO DEL BODY

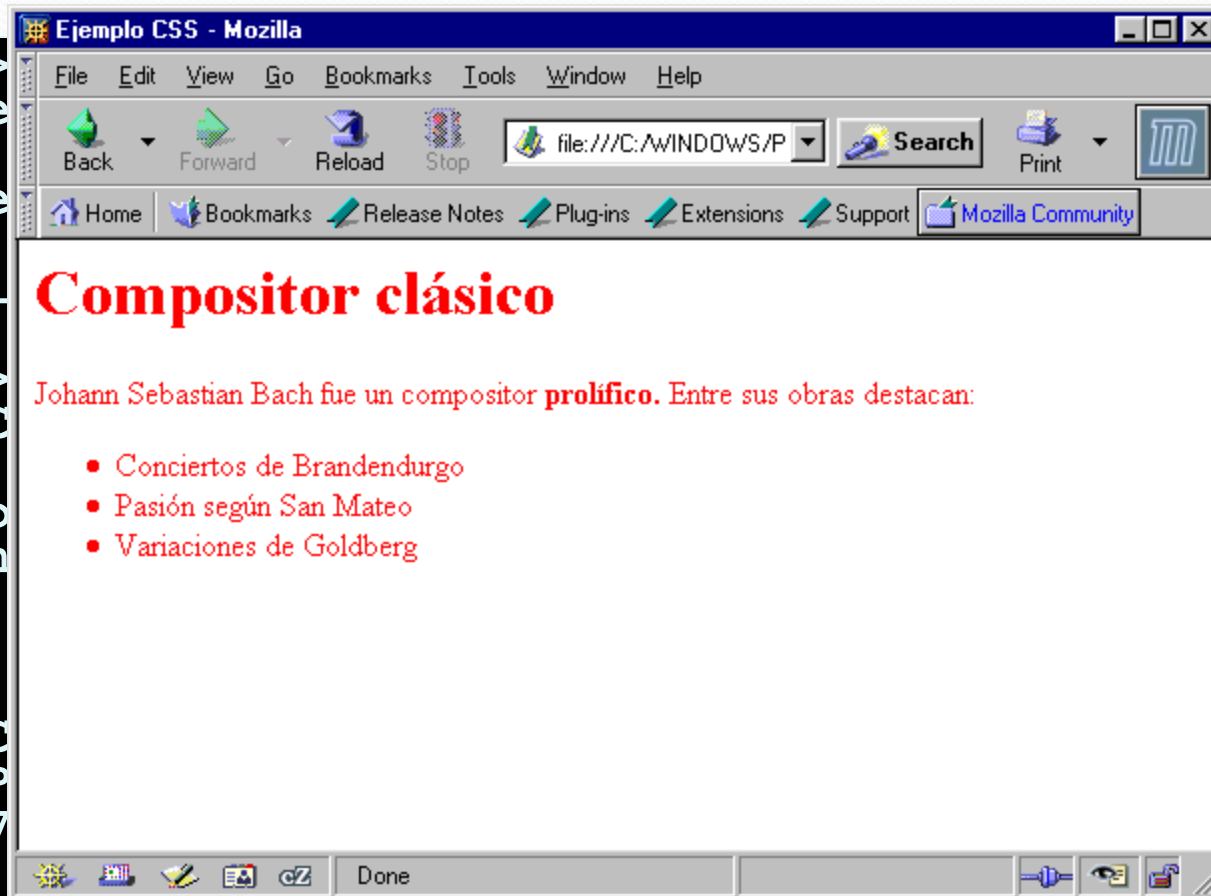

```
<html>
<title>
<style>
</style>
<body>
<h1> Compositor clásico

<p> Johann Sebastian Bach fue un compositor prolífico. Entre sus obras destacan:

<strong>
obras

<ul>
<li> Conciertos de Brandeburgo
<li> Pasión según San Mateo
<li> Variaciones de Goldberg
</ul>

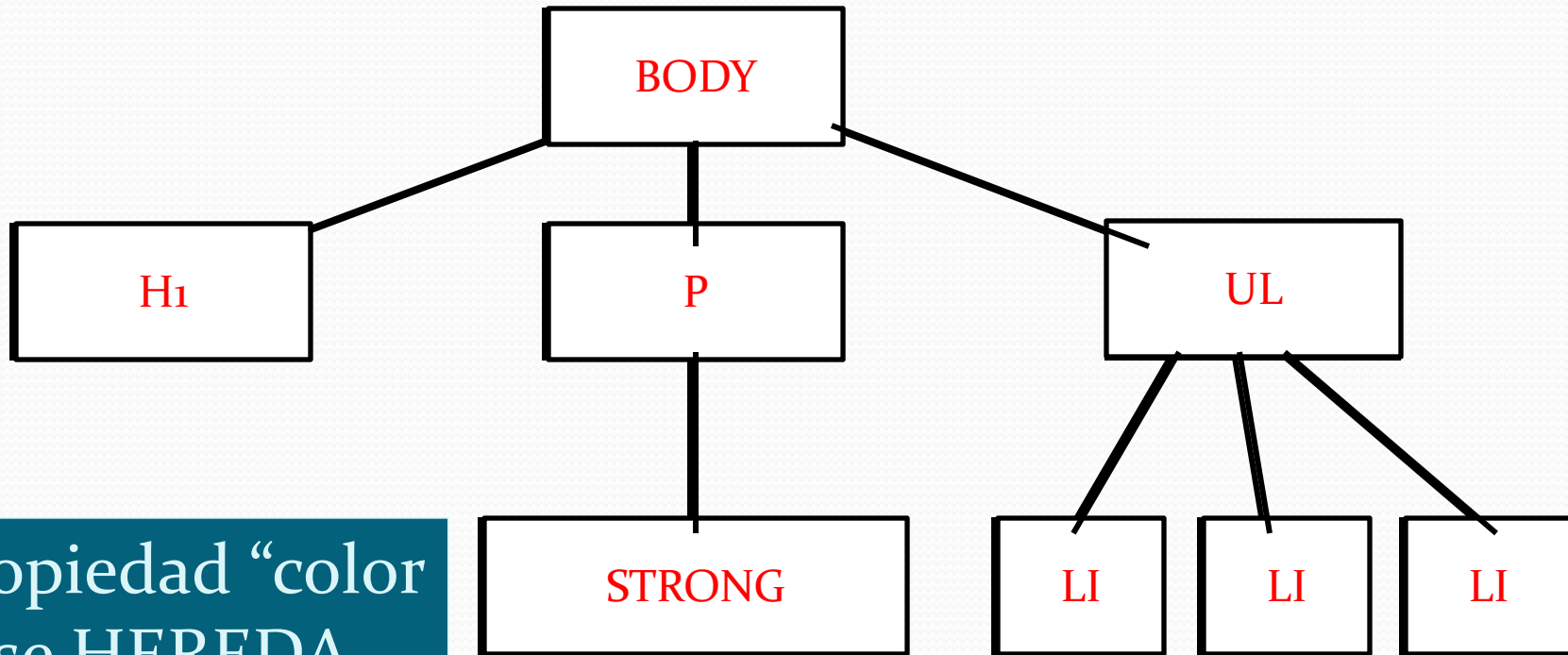
</body>
</html>
```



¿QUÉ HA SUCEDIDO?

Herencia de propiedades en CSS

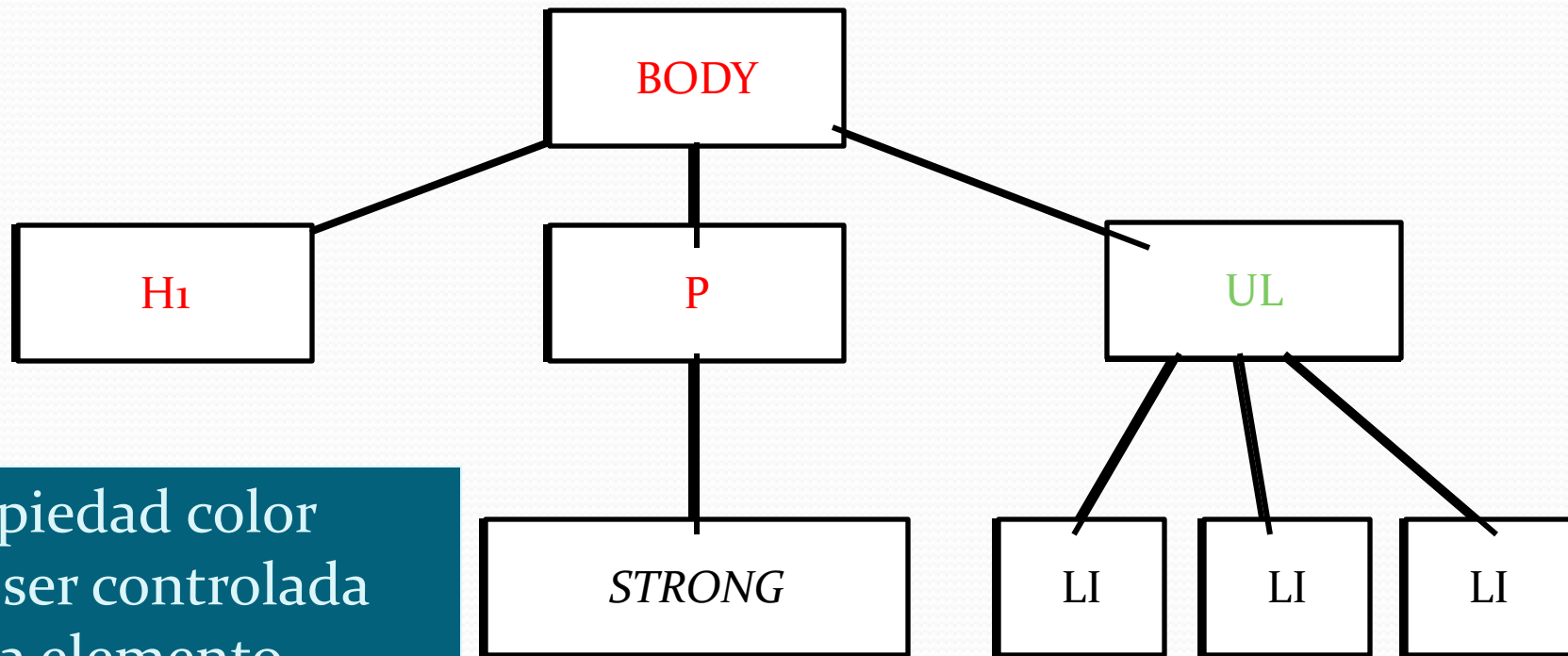
Ejemplo:



La propiedad “color rojo” se HEREDA por todos los hijos.

Herencia de propiedades en CSS

Ejemplo:



La propiedad color puede ser controlada en cada elemento específico: UL → verde
STRONG → *Itálica y negro*

```
H <html>
<title>Ejemplo CSS</title>

<style>
  body {color: red}
  strong {color: black; font-style: italic}
  ul {color: green}
</style>

<body>
<h1> Compositor cl&acute;sico </h1>

<p> Johann Sebastian Bach fue un compositor
<strong> prol&iacute;fico. </strong> Entre sus
obras destacan: </p>

<ul>
<li> Conciertos de Brandendurgo
<li> Pasión según San Mateo
<li> Variaciones de Goldberg
</ul>

</body>

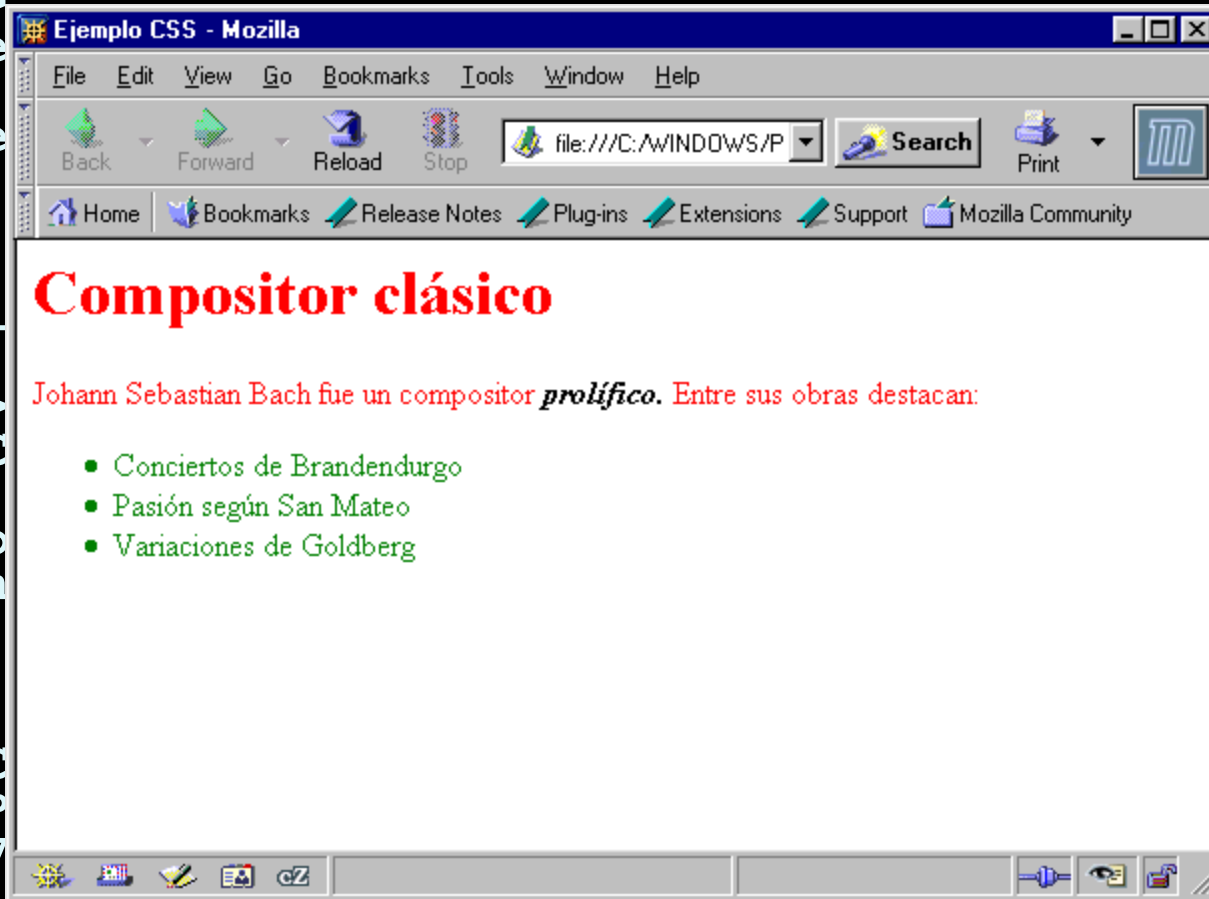
</html>
```

CAMBIO EN ESTILO DEL UL y
STRONG

```

<html>
<title>
<style>
</style>
<body>
<h1> C
<p> Jo
<strong>
obras
<ul>
<li> C
<li> P
<li> V
</ul>
</body>
</html>

```

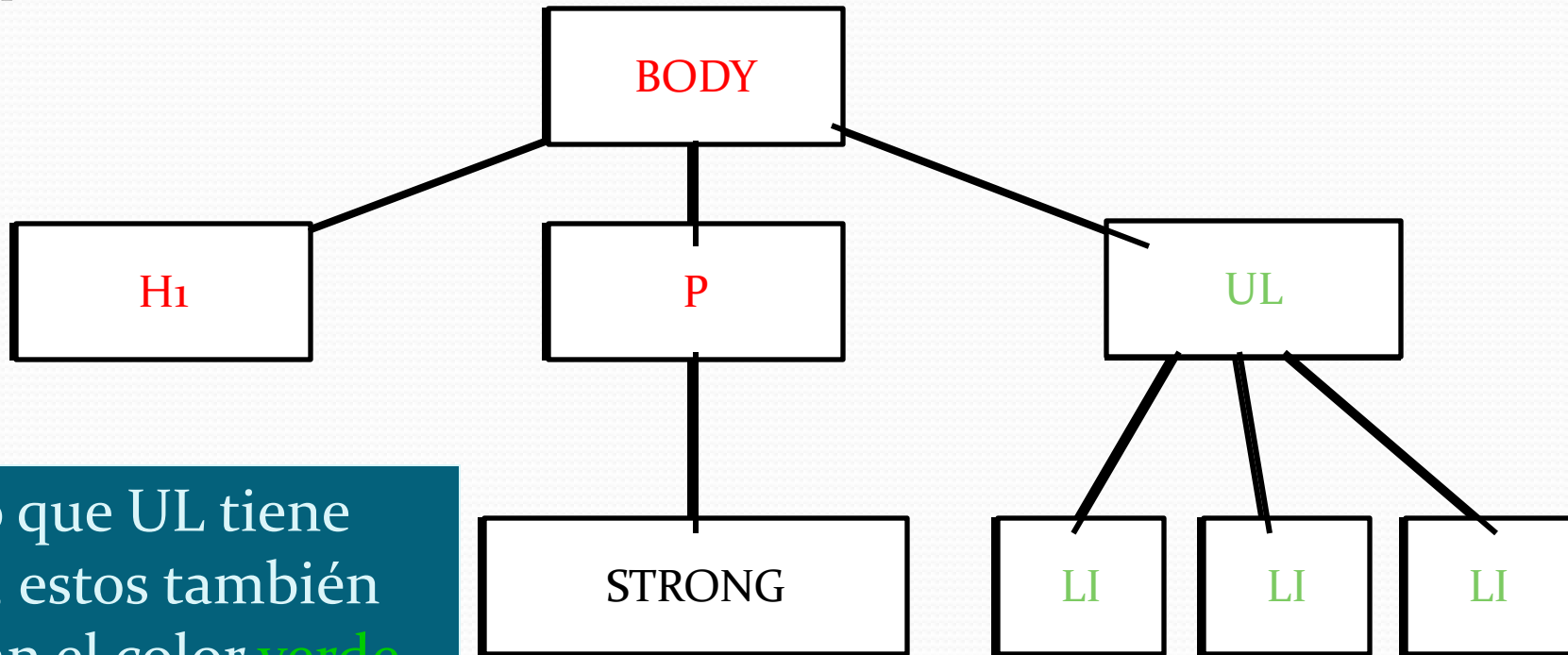


ics}

us

Herencia de propiedades en CSS

Ejemplo:



Puesto que UL tiene “hijos”, estos también heredan el color verde

H

```
<html>
<title>Ejemplo CSS</title>

<style>
    ul {color: green}
    strong {color: black; font-style: italic}
    body {color: red}
</style>

<body>
<h1> Compositor cl&acute;sico;

<p> Johann Sebastian Bach fu
<strong> prol&iacute;fico.
obras destacan: </p>

<ul>
<li> Conciertos de Brandendurgo
<li> Pasión según San Mateo
<li> Variaciones de Goldberg
</ul>

</body>

</html>
```

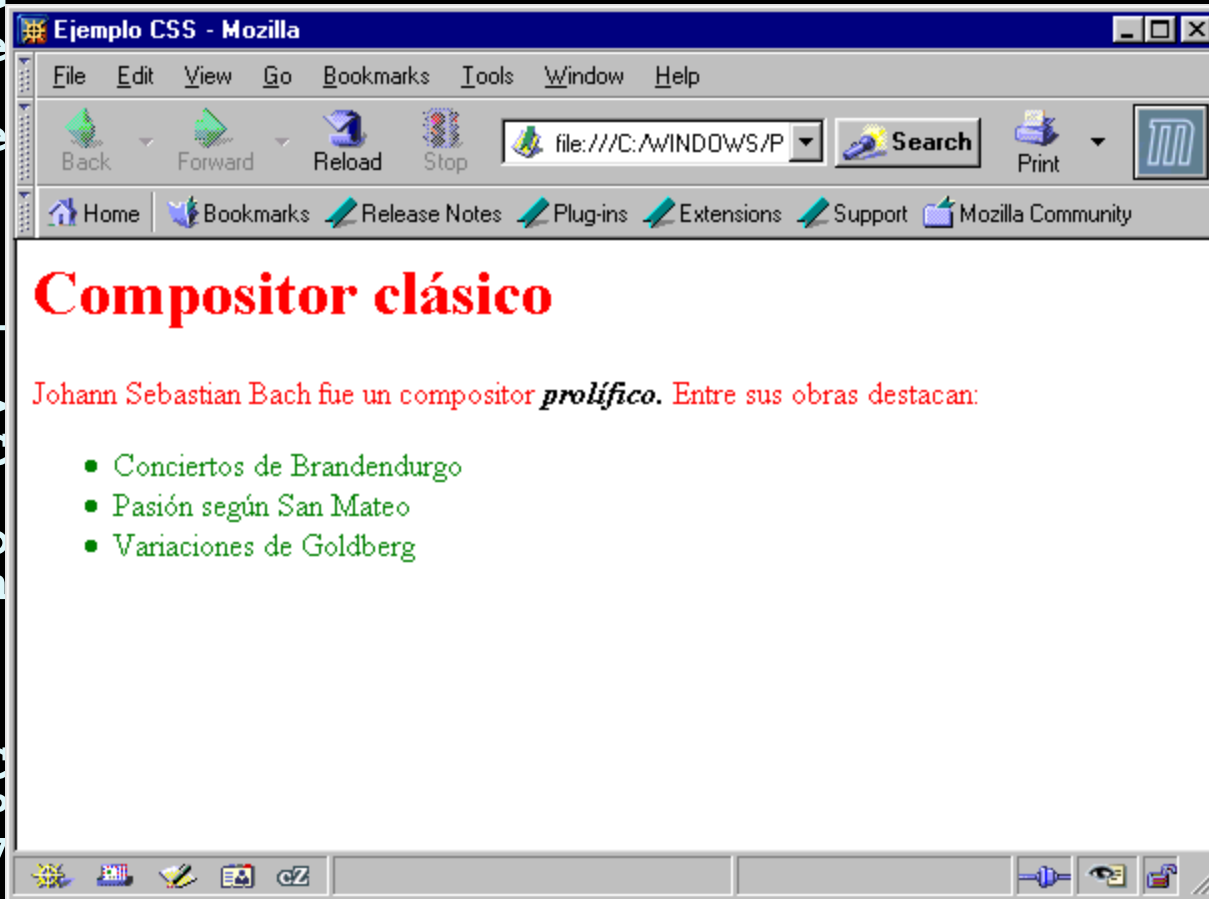


PREFERENCIA ESTÁ
DETERMINADA POR LA
ESPECIFICIDAD, NO POR
EL ORDEN DE
DECLARACIÓN.

```

<html>
<title>
<style>
</style>
<body>
<h1> C
<p> Jo
<strong>
obras
<ul>
<li> C
<li> P
<li> V
</ul>
</body>
</html>

```



ics}

us

Declaración avanzada de CSS

- Selectores:

- * Selector universal (cualquier elemento):

- * `{font-family: Arial}`

Declaración avanzada de CSS

- Selectores:

E Identifica cualquier elemento E:

```
LI {font-family: Arial}
```

```
P {color: black}
```

Declaración avanzada de CSS

- Selectores:

E>F Cualquier elemento F que sea hijo de E

UL>LI {font-size: 70%}

OL>LI {font-size: 115%}

```
H <html>
<title>Ejemplo CSS</title>

<style>
  ol>li{font-size: 70%}
  ul>li{font-size: 115%}
</style>

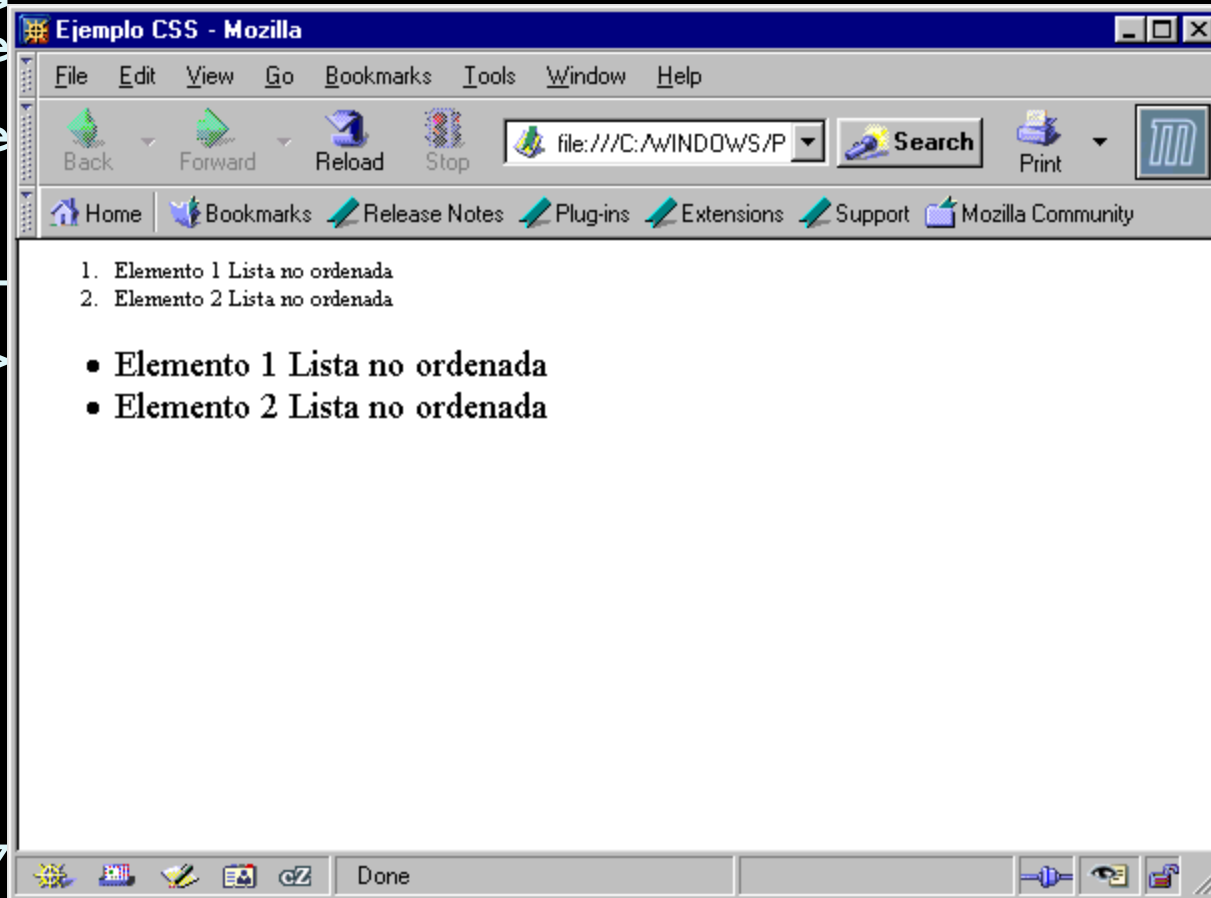
<body>
  <ol>
    <li> Elemento 1 Lista no ordenada
    <li> Elemento 2 Lista no ordenada
  </ol>

  <ul>
    <li> Elemento 1 Lista no ordenada
    <li> Elemento 2 Lista no ordenada
  </ul>

</body>
</html>
```

LI SE VISUALIZA DIFERENTE
SEGÚN QUIEN SEA EL PADRE

```
<html>  
<title>  
<style>  
</style>  
<body>  
  
</body>  
</html>
```



a
a
a
a

Declaración avanzada de CSS

- Selectores:

E F Cualquier elemento F descendiente de E

```
H1 EM{color: blue}
```

Declaración avanzada de CSS

```
<html>
<title>Ejemplo CSS</title>

<style>
    H1 { color: red }
    EM { color: red }
    H1 EM { color: blue }
</style>

<body>
    <h1>Esta <span class="myclass">cabecera
    es <em>muy</em> importante</span></h1>

    Esta <em>palabra</em> es tambi&eacute;n
    importante.
</body>
</html>
```

Declar

```
<html>  
<title>
```

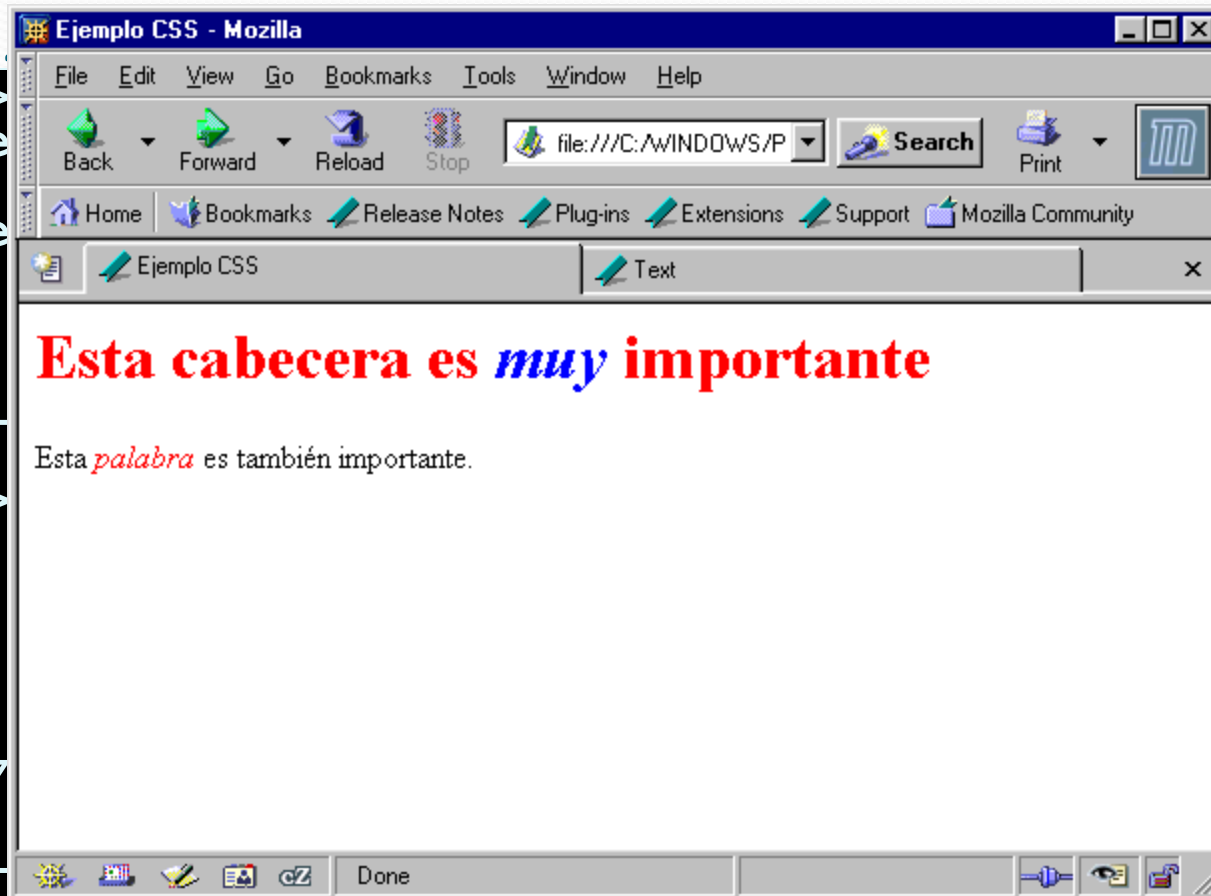
```
<style>
```

```
</styl
```

```
<body>
```

```
</body>
```

```
</html>
```



a
n

Declaración avanzada de CSS

- Selectores:

E+F F inmediatamente después de E (en el mismo nivel del árbol)

`H1+H2{color: red}`

Declaración avanzada de CSS

```
<html>
<title>Ejemplo CSS</title>

<style>
    h1 + h2 { color: red }
</style>

<body>
    <h1> Cabecera principal </h1>
    <h2> Cabecera secundaria </h2>

    <h1> Otra cabecera principal </h1>
    <p> Parrafo intermedio
    <h2> Cabecera secundaria </h2>
</body>
</html>
```

Declar

```
<html>  
<title>
```

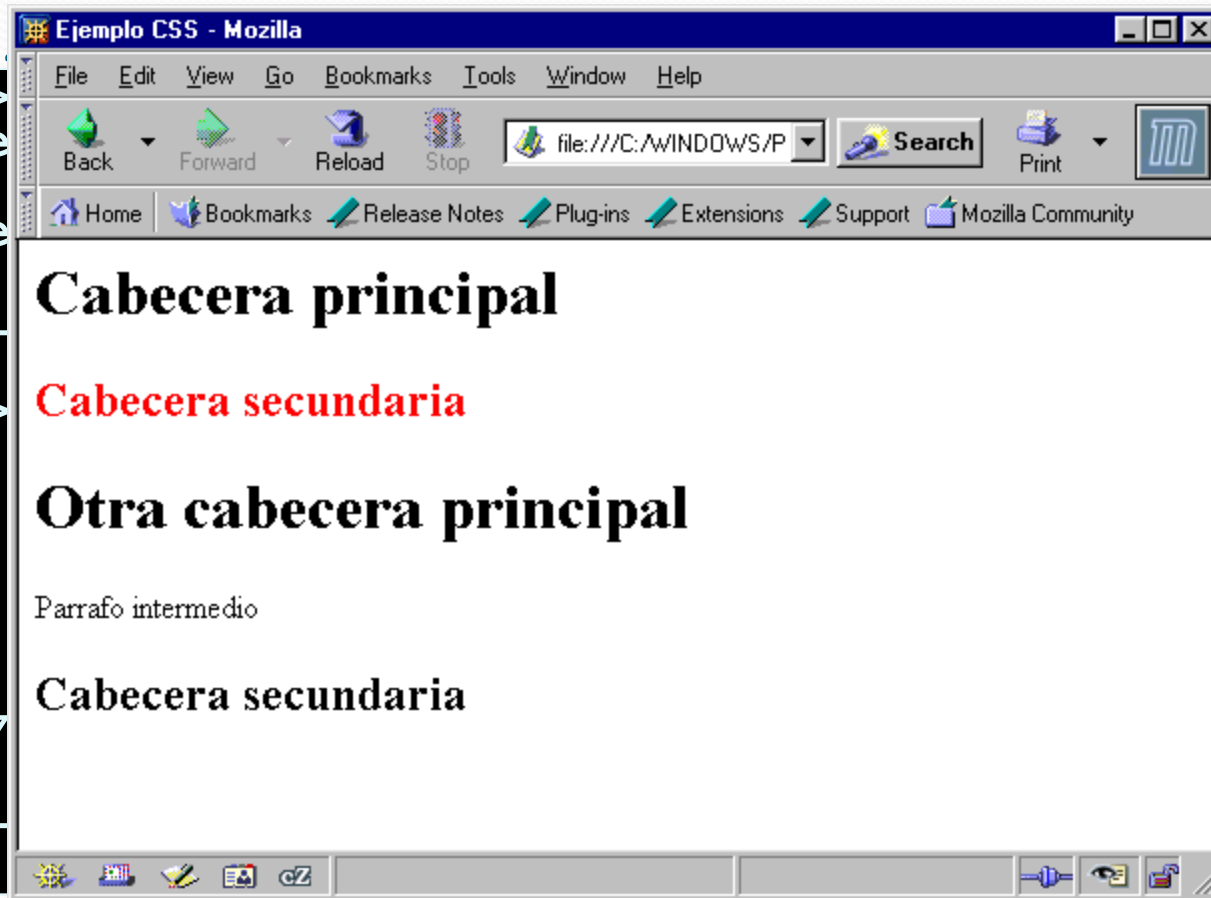
```
<style>
```

```
</style>
```

```
<body>
```

```
</body>
```

```
</html>
```



Declaración avanzada de CSS

```
<html>
<title>Ejemplo CSS</title>

<style>
  h1 + h2 { color: red }
</style>

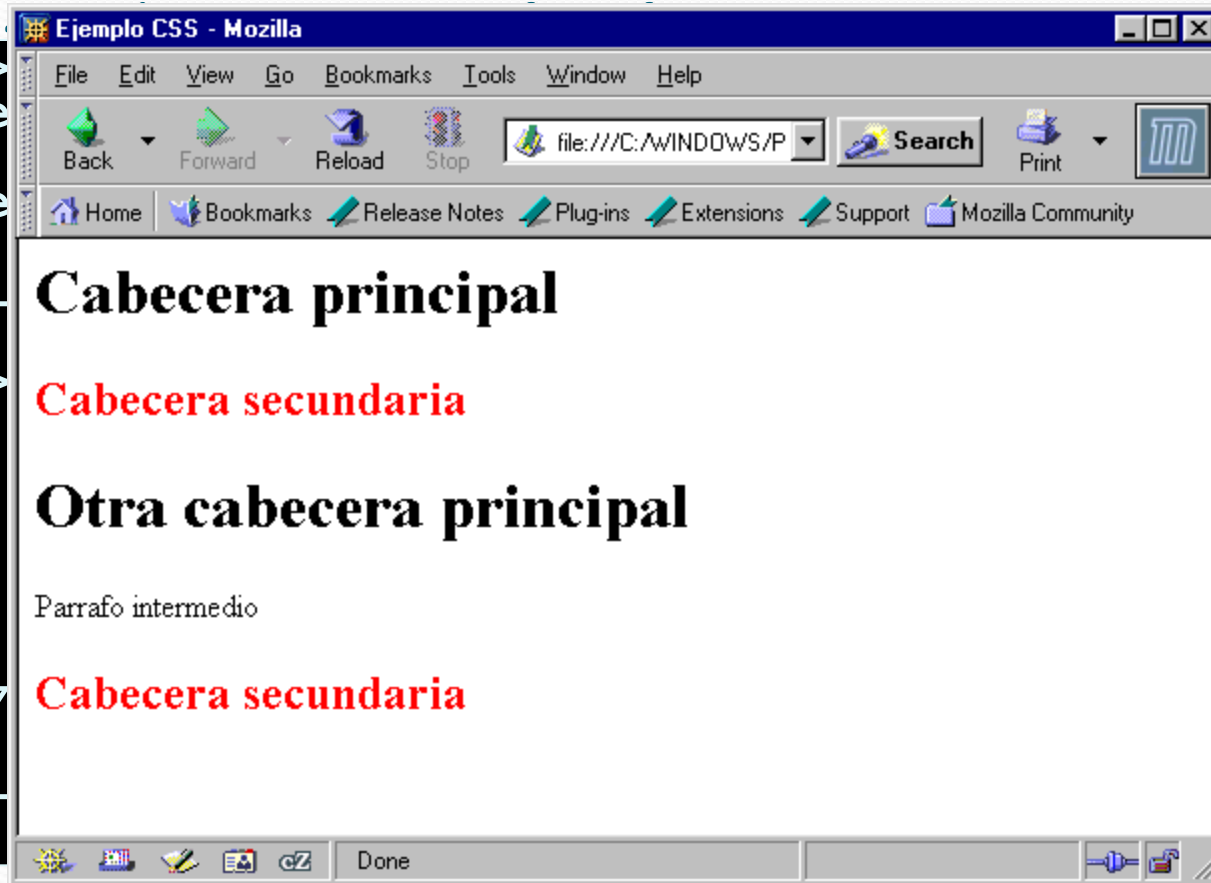
<body>
  <h1> Cabecera principal </h1>
  <h2> Cabecera secundaria </h2>

  <h1> Otra cabecera principal </h1>
  Parrafo intermedio
  <h2> Cabecera secundaria </h2>
</body>
</html>
```

El selector afecta en presencia de un texto intermedio (ya que éste no es un nuevo elemento)

Declar

```
<html>  
<title>  
<style>  
</style>  
<body>  
  
</body>  
</html>
```



Identificadores y clases I

- El uso de los selectores anteriores puede llegar a ser muy complejo y no proporciona la flexibilidad suficiente
- En muchos casos es más sencillo “nombrar” los elementos para aplicarles un estilo determinado.
- Se utiliza el atributo “id” para identificar unívocamente a un elemento (todos los id tienen que ser únicos en un documento)
- Se utiliza el atributo “class” para identificar a un conjunto de elementos que comparten atributos

Identificadores y clases II

- Selectores:

`E#myid` Cualquier elemento E cuyo atributo "id" sea "myid"

```
TABLE#precios {text-align: center}
```

`E.myid` Cualquier elemento E cuyo atributo "class" sea "myid"

```
TABLE.precios {text-align: center}
```

Pseudo clases: a veces es necesaria mayor flexibilidad. Se han definido pseudo clases y pseudo elementos, identificados con `:`, que permiten lo siguiente:

```
<A class="external" href="http://out.side/">external  
link</A>
```

```
a.external:visited { color: blue }
```

Los enlaces visitados se
mostrarán en azul

```
D <html>
<title>Ejemplo CSS</title>

<style>
    table#precios {
        text-align: center;
        font-style: italic
    }
</style>

<body>
    <table id="precios">
        <tr><td>1<td>2<td>3
        <tr><td>uno<td>dos<td>tres
    </table>

    <table class="otra">
        <tr><td>1<td>2<td>3
        <tr><td>uno<td>dos<td>tres
    </table>
</body>
</html>
```

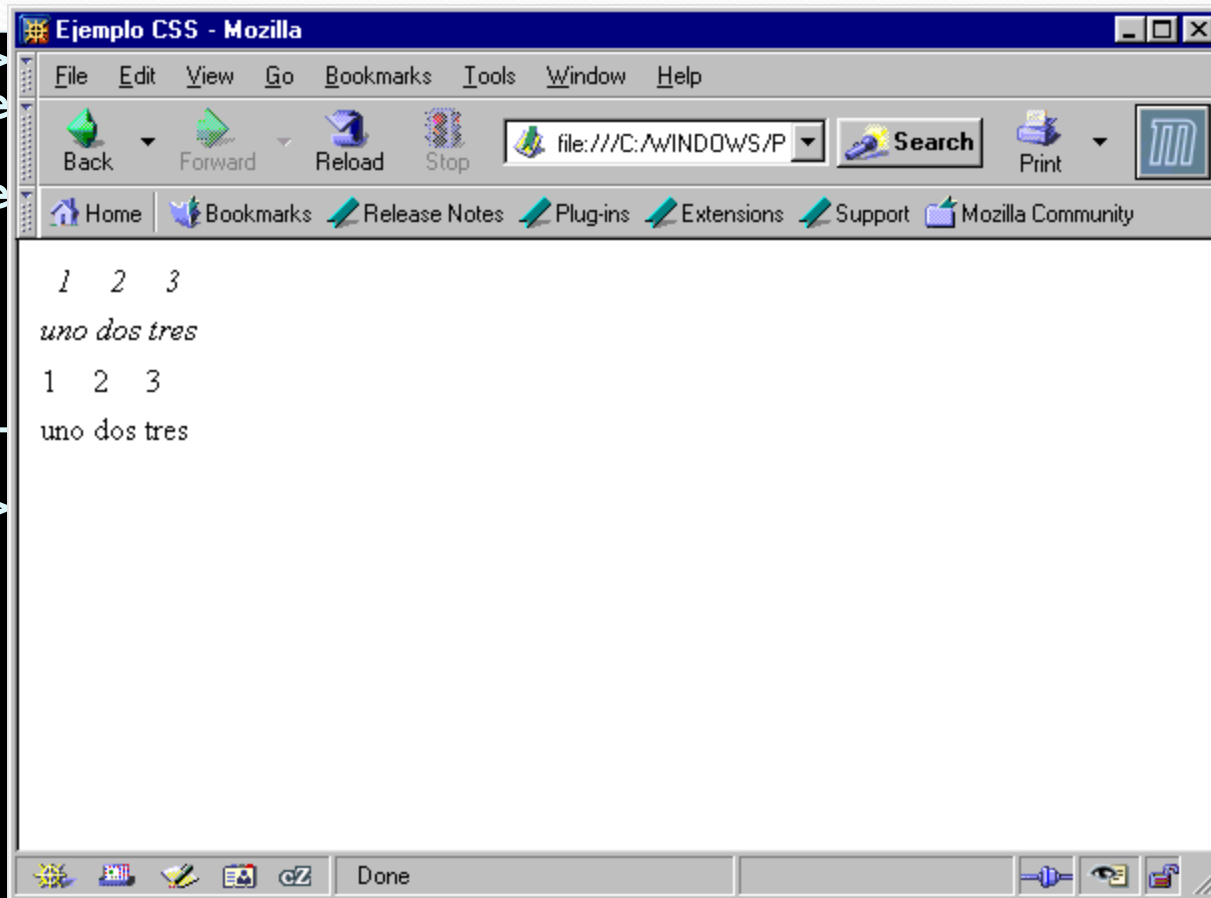

D

```
<html>  
<title
```

```
<style
```

```
</styl
```

```
<body>
```



```
</table>
```

```
</body>
```

```
</html>
```

Declaración avanzada de CSS

- Selectores:

REGLA DE AGRUPACIÓN:

```
H1 {color: red}  
H2 {color: red}  
H3 {color: red}
```

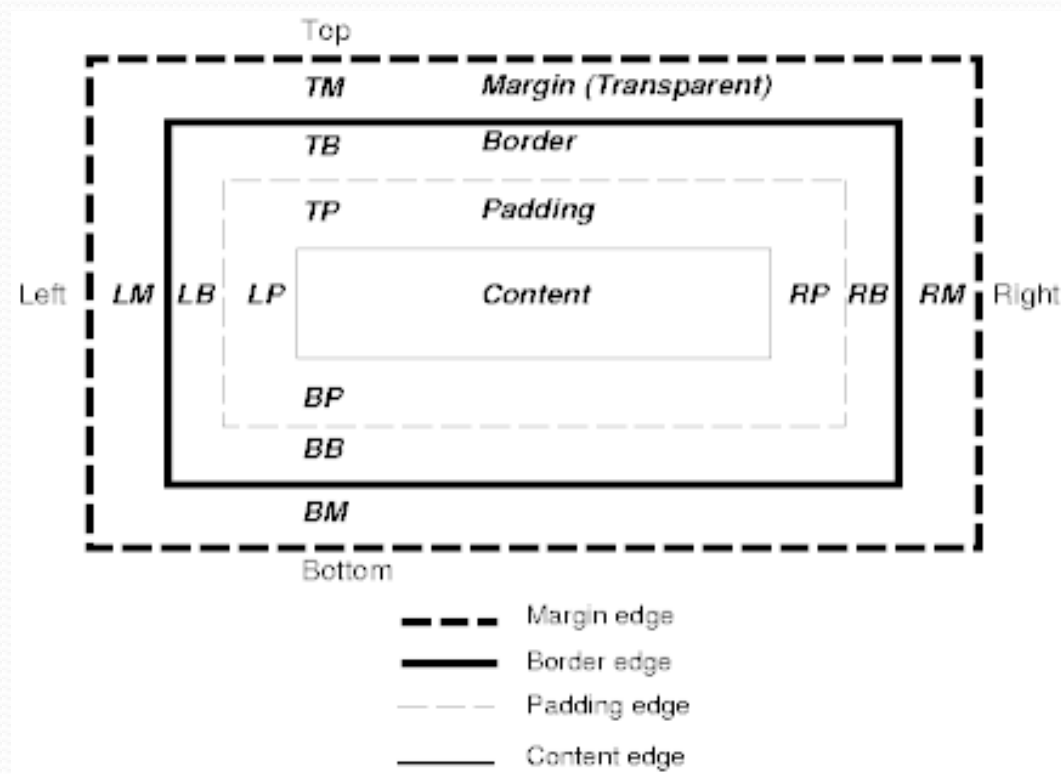
Es similar a:

```
H1 , H2 , H3 {color: red}
```

Forma preferida: más clara,
y ocupa menos bytes.

Modelo de cajas (I)

- Cada elemento en el árbol de un documento genera cero o más cajas anidadas como esta



Posicionamiento

- La parte más compleja de CSS (y probablemente la de mayor utilidad)
- Se definen varios esquemas de posicionamiento: flujo normal, flotante (float) y absoluto.
- Las cajas de los elementos se posicionarán según el esquema elegido
- Además se pueden emplear adicionalmente capas: la propiedad z-index permite asignar un nivel en la capa, a mayor valor, más arriba se sitúa la capa

Estructura (layout)

- Normalmente se define una estructura común para todo un sitio
 - Por ejemplo: tres columnas, cabecera y pie.
 - En cada contenedor de la estructura se insertan nuevos elementos
- La estructura se suele especificar mediante el uso de elementos `<div>` e ids asociados

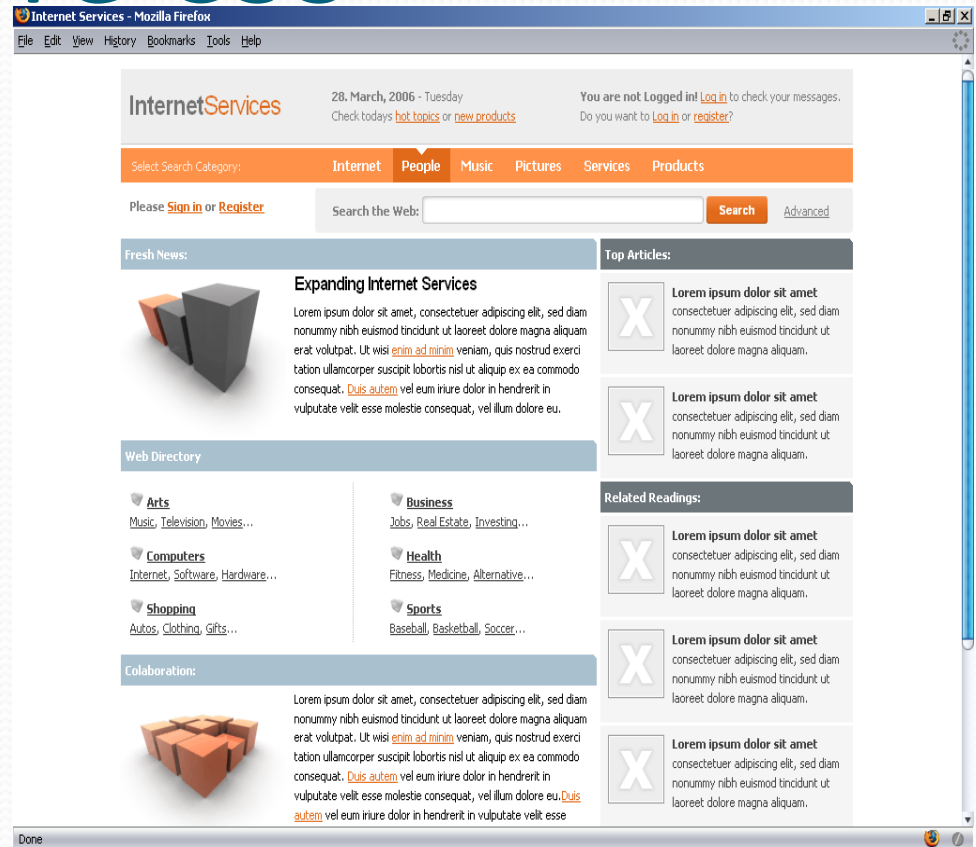


Conclusión sobre CSS

- Permite de forma sencilla separar el contenido de su formato.
 - Revise los ejemplos: Observará que el formato siempre se definió de modo independiente!
- Su uso es imprescindible en cualquier aplicación profesional.
- Hay miles de plantillas disponibles
 - <http://www.oswd.org/> Open Source Web Design
 - Y otras muchas

Conclusión sobre CSS

```
body {  
    padding: 0;  
    margin: 0;  
    font: 0.7em Tahoma, sans-serif;  
    line-height: 1.5em;  
    background: #fff;  
    color: #454545;  
}  
  
a {  
    color: #E0691A; background: inherit;  
}  
a:hover { color: #6C757A; background: inherit; }  
  
h1 {  
    font: normal 2.1em Arial, Sans-Serif; letter-  
    spacing: -1px; padding: 7px 0 0 8px; margin: 0;  
}  
.dark { color: #737373; background: inherit; font-weight: bold; }  
h1 a, h2 a { text-decoration: none; }  
h1 a:hover, h2 a:hover { color: #FF6600; background: inherit; }  
h2 {  
    margin: 0; padding: 0; font: bold 1.7em Arial,  
    Sans-Serif; letter-spacing: -1px;  
}  
  
p {margin: 0 0 5px 0; }  
  
ul { margin: 0; padding : 0; list-style : none; }
```



Referencias y bibliografía

- Libros:
 - “Cascading Style Sheets, designing for the Web”, Hakon Wium Lie, Bert Bos, Addison Wesley.
- En Web:
 - <http://www.w3c.org/Style/CSS/> → Especificación
 - <http://www.w3schools.com/css/default.asp> → Tutorial
 - <http://www.oswd.org/> -> Open Source Web Design: miles de plantillas gratuitas