

UT3. 02_CSS. HERENCIA Y CASCADA

Curso: 2DAW.

Lenguajes de Marcas y sistemas de gestión de información


Herencia y cascada

- **Herencia** → cómo los elementos de HTML heredan propiedades de sus elementos padres y los transmiten a sus hijos.
- **Cascada** → declaraciones de CSS que se aplican a un documento
→ cómo las reglas contradictorias se anulan o no entre ellas.

Herencia

- Para entender la **herencia** es necesario entender el árbol del documento.

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <link href="html5_estilo2.css" rel="stylesheet">
6 <title>Árbol</title>
7 </head>
8 <body>
9 <div class="caja">
10   <header>
11     <h1>Cabecera: header</h1>
12   </header>
13   <nav>
14     <ul>
15       <li><a href="#">Menú 1</a></li>
16       <li><a href="#">Menú 2</a></li>
17       <li><a href="#">Menú 3</a></li>
18     </ul>
19   </nav>
20   <article>Contenido de la página </article>
21   <footer>Pie de página</footer>
22 </div>
23 </body>
24 </html>
```

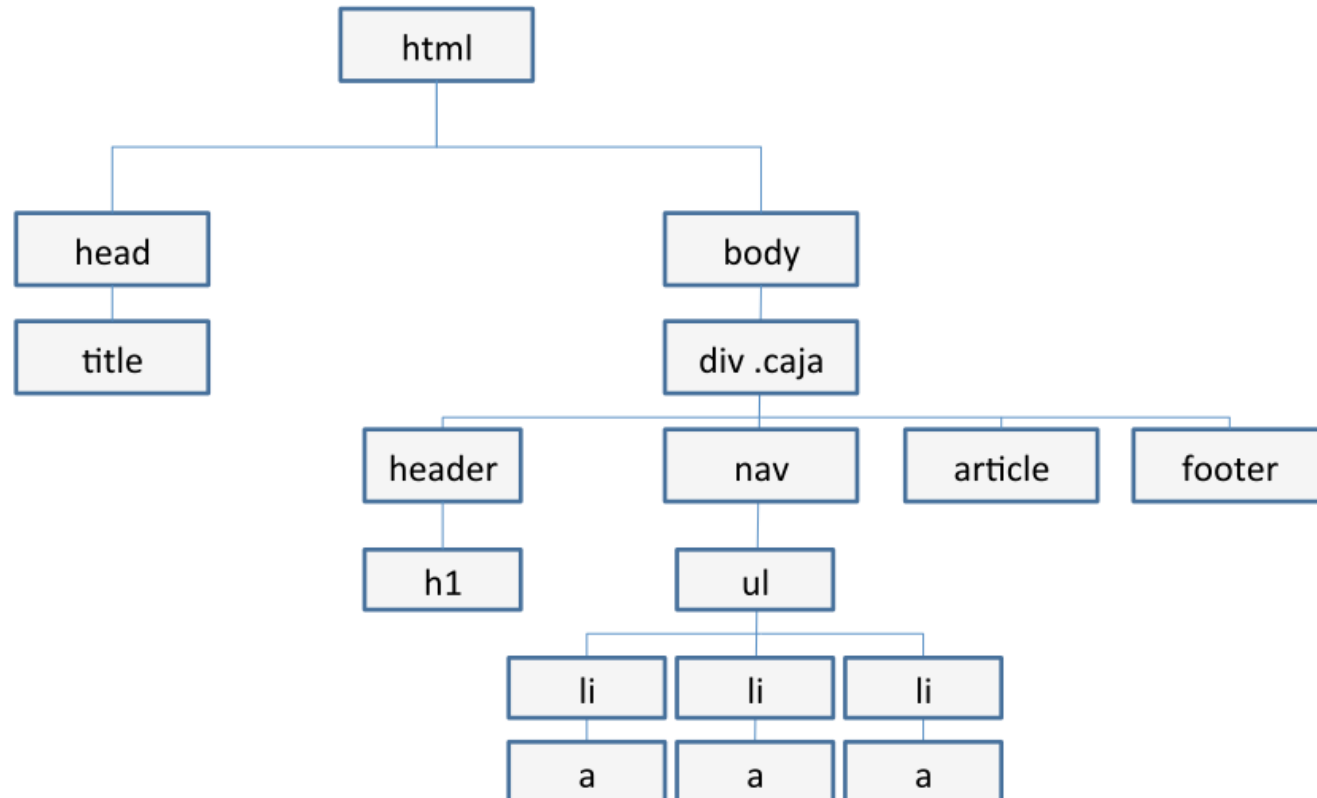


The screenshot shows a Firefox browser window with the DOM tree expanded. The tree structure is as follows:

- Árbol
 - Cabecera: header
 - Menú 1
 - Menú 2
 - Menú 3
 - Contenido de la página
 - Pie de página

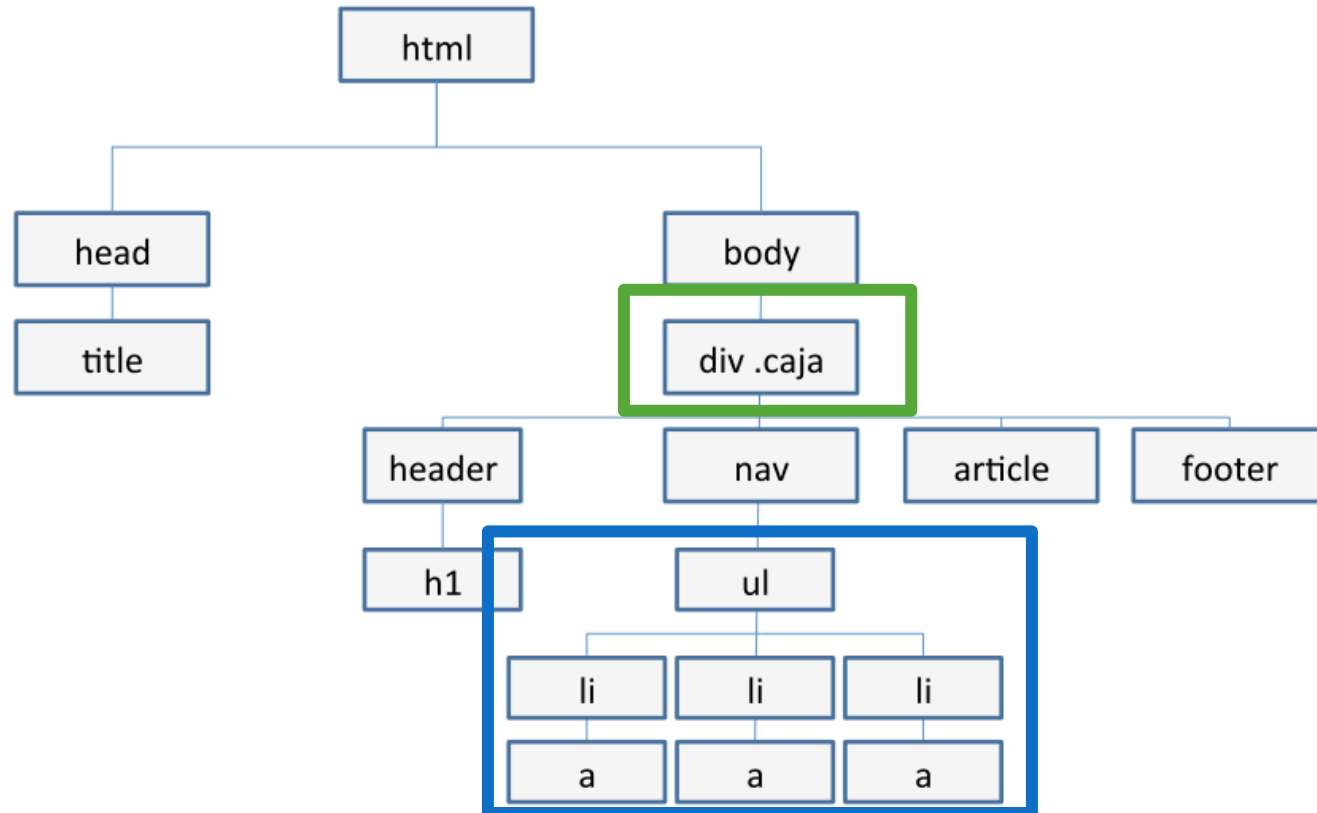
Herencia

- Ancestor
- Descendientes
- Parent
- Descendientes
- Parent
- Child
- Parent
- Siblings



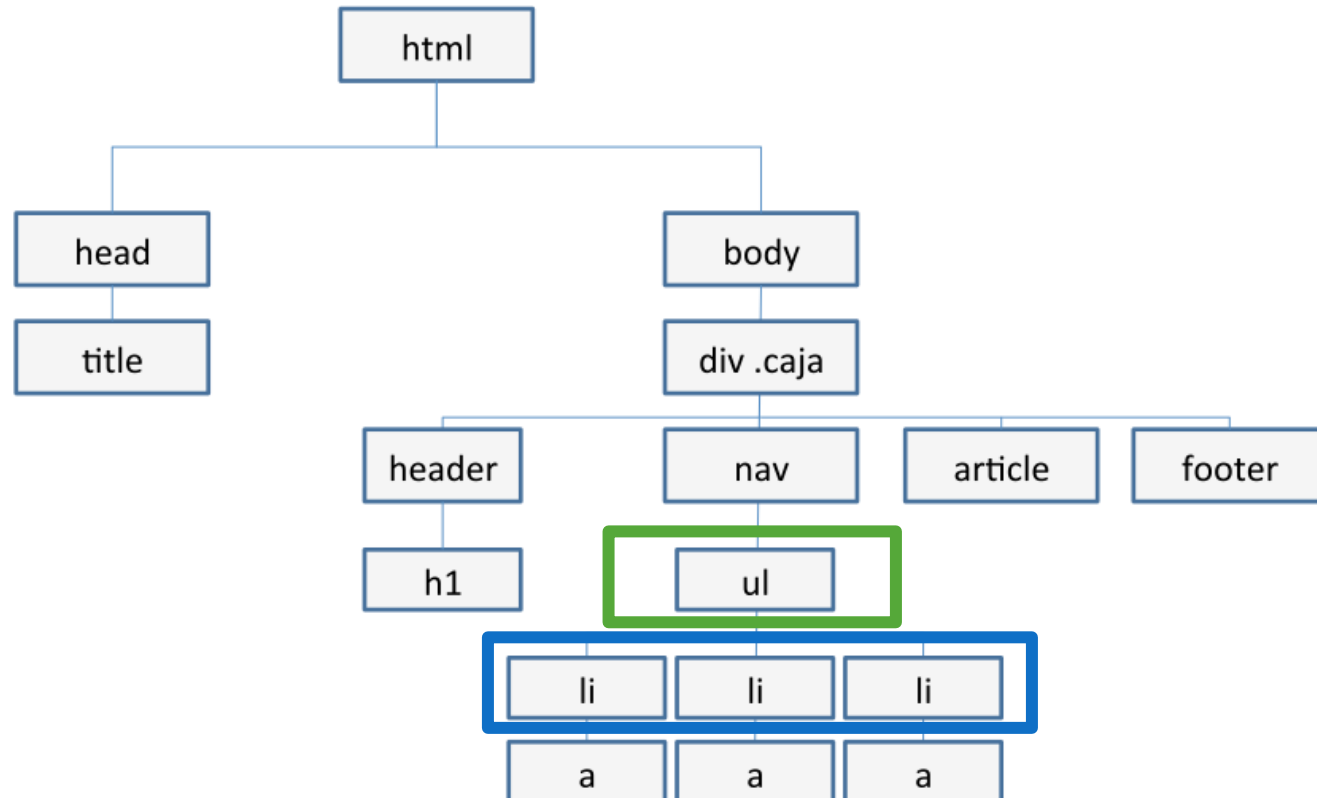
Herencia

- **Ancestor**
- **Descendientes**
- Parent
- Descendientes
- Parent
- Child
- Parent
- Siblings



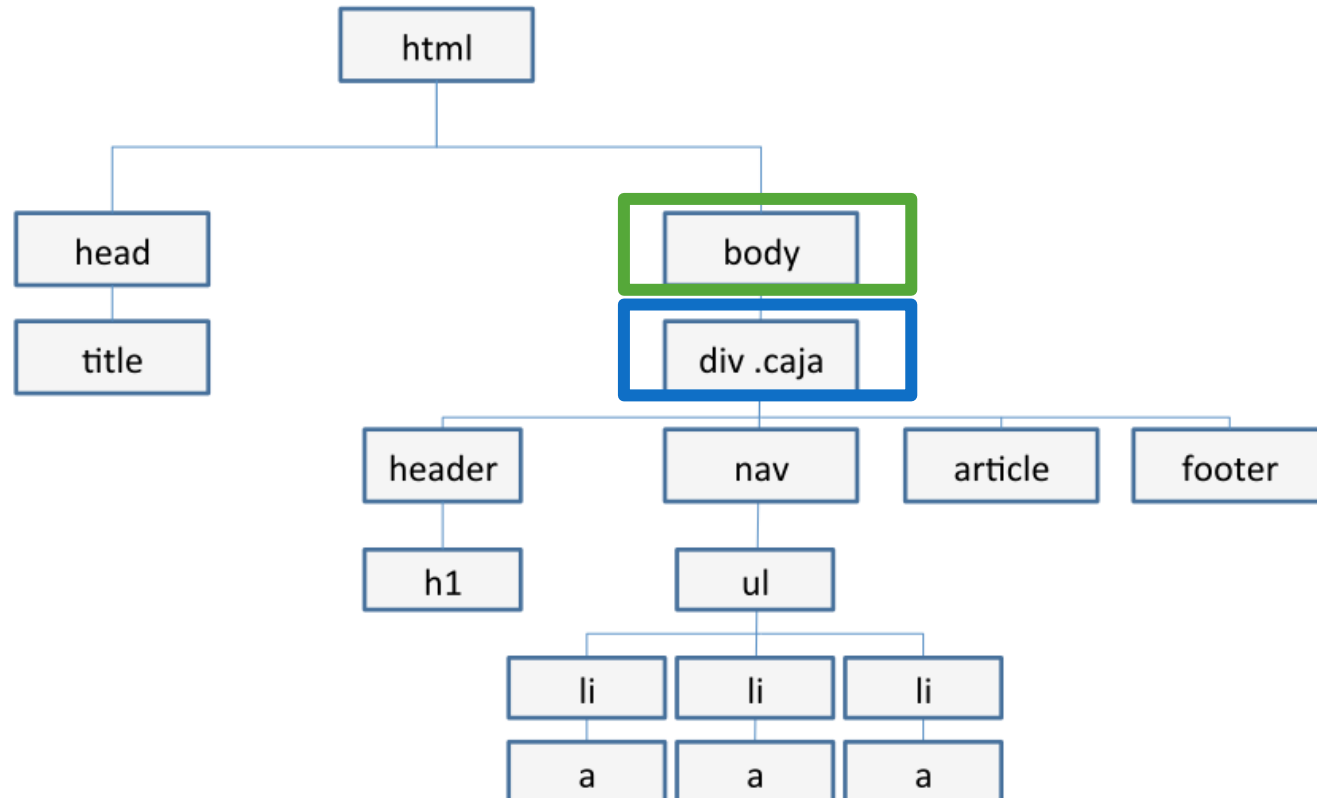
Herencia

- Ancestor
- Descendientes
- **Parent**
- **Descendientes**
- Parent
- Child
- Parent
- Siblings



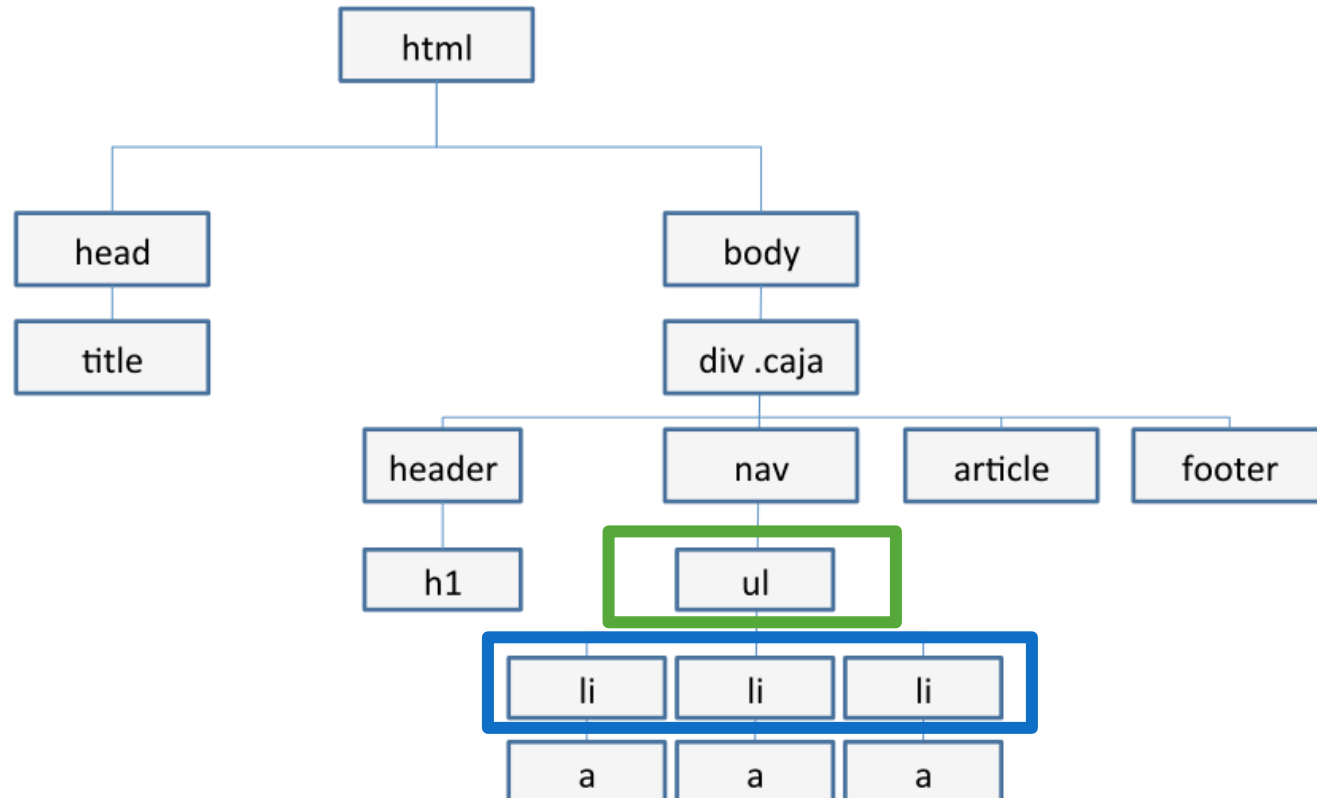
Herencia

- Ancestor
- Descendientes
- Parent
- Descendientes
- **Parent**
- **Child**
- Parent
- Siblings



Herencia

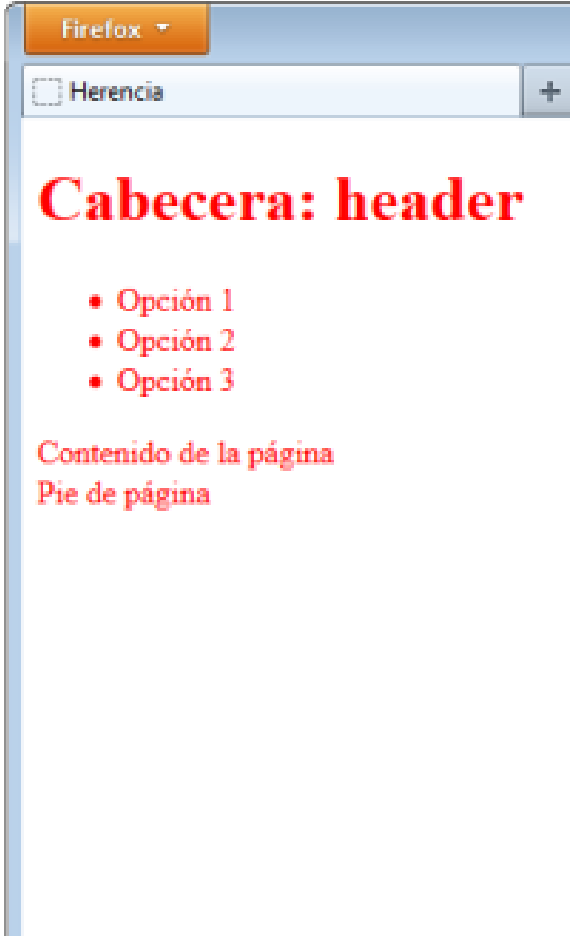
- Ancestor
- Descendientes
- Parent
- Descendientes
- Parent
- Child
- **Parent**
- **Siblings**



Herencia

- El atributo **color** es heredado por los elementos que están por debajo en el árbol.
- Algunas propiedades no se heredan.

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Herencia</title>
6 <style type="text/css">
7 body {
8     color: red;
9 }
10 </style>
11 </head>
12 <body>
13 <div class="caja">
14     <header>
15         <h1>Cabecera: header</h1>
16     </header>
17     <nav>
18         <ul>
19             <li>Opción 1</li>
20             <li>Opción 2</li>
21             <li>Opción 3</li>
22         </ul>
23     </nav>
24     <article>Contenido de la página </article>
25     <footer>Pie de página</footer>
26 </div>
27 </body>
28 </html>
```



Firefox ▾

Herencia +

Cabecera: header

- Opción 1
- Opción 2
- Opción 3

Contenido de la página

Pie de página

Herencia

- Todos los elementos de un documento HTML heredan todas las propiedades **heredables** de su padre excepto el elemento raíz (html), que no tiene progenitor.
- No todas las propiedades se heredan. Por ejemplo, la propiedad **font-family** es heredable, pero los márgenes (propiedad **margin**) no se heredan excepto si se indica explícitamente que deben heredarse.
- ¿Por qué unas propiedades se heredan y otras no?
 - Algunas propiedades se heredarán porque suelen mantenerse constantes dentro de una caja. Por ejemplo, lo más normal es usar un tipo de letra en un contenedor y no mezclar distintos tipos de letra.
 - En cambio, otras propiedades no, como los márgenes. Por ejemplo, las fotografías dentro de un artículo normalmente tienen un margen distinto al que tiene el propio artículo respecto a su contenedor o los párrafos.
 - Atributo **“inherit”**: puede estar establecido en **“yes”** o en **“no”** según la propiedad de que se trate. **font-family** es una propiedad cuyo valor inherit es **“yes”**: esta propiedad se hereda. En cambio la propiedad margin tiene un valor inherit **“no”**, lo cual significa que no se hereda.

Herencia

- **background-color** no es heredable (pero lo parece)
 - El color de fondo por defecto: transparente.
 - Se superpone el color transparente del hijo, y el efecto visual es que se hereda el color del padre, aunque formalmente no es así.

```
<header>
  <h1>Mi primer Portal Web</h1>
  <h2>Práctica con la herencia</h2>
</header>
<br />
<nav>
  <div>Menú</div>
  <hr/>
  <ul>
    <li><a href="#">Inicio</a></li>
    <li><a href="libros.html">Libros de LM</a> </li>
    <li><a href="articulos.html">Artículos de HTML</a> </li>
    <li><a href="colaboraciones.html">Colaboraciones en la Web</a>
  </li>
  </ul>
</nav>
```

Mi primer Portal Web

práctica con la

Menú

- [Inicio](#)
- [Libros de LM](#)
- [Artículos de HTML](#)
- [Colaboraciones en la Web](#)

Herencia

Mi primer Portal Web

Práctica con la herencia

Menú

- [Inicio](#)
- [Libros de LM](#)
- [Artículos de HTML](#)
- [Colaboraciones en la Web](#)

```
body {font-family: Arial, sans-serif;}  
nav { color: red; text-decoration:none;}
```

color heredable y **text-decoration** no heredable.
No se heredan ninguna porque **<a>** sobrescribe las propiedades que pueda heredar

Mi primer Portal Web

Práctica con la herencia

Menú

- Inicio
- Libros de LM
- Artículos de HTML
- Colaboraciones en la Web

```
1 body {font-family: Arial, sans-serif;}  
2 nav, nav a { color: red; text-decoration:none;}
```

Para modificar **<a>** se hace de forma explícita

```
body {font-family: Arial, sans-serif;}  
nav { color: red; text-decoration:none;}  
nav a { color: inherit; text-decoration:inherit; }
```

○ forzando la herencia

Herencia

- **Un ejemplo de herencia**

- Creamos un archivo **herencia1.html** con (h1-párrafo)
- Creamos un archivo con la regla CSS en **herencia1.css** en la misma ubicación que el fichero HTML.
 - Fuente para toda la página (html) Verdana, Sans-serif y al 75%
- Enlazamos el .html con el .css. ¿Qué sucede?
 - Los elementos html heredan del antecesor (html)
 - 75% de ... el valor computado (valor por defecto de los navegadores que suele ser 16px → 12px)

Herencia

- Añadimos dos declaraciones más a la regla de la hoja de estilo de CSS.
 - Color de fondo azul y color de la letra en blanco.
 - **Heredan el color.**
- Añadimos: tamaño de la fuente a 300% en h1
 - Cambia el tamaño de la fuente al 300% del 75% ➔ 36px.
 - El valor que se hereda **no** es el valor especificado, es decir, el valor que escribimos en la hoja de estilo, sino algo que se llama *el valor computado*.
 - El valor computado es, en el caso del tamaño de la fuente, un valor absoluto medido en píxeles.
- Buscar el tamaño que tiene por defecto cada una de los títulos (h1, h2, ...)

Valores por defecto de <h1>

```
h1 {  
  display: block;  
  font-size: 2em;  
  margin-top: 0.67em;  
  margin-bottom: 0.67em;  
  margin-left: 0;  
  margin-right: 0;  
  font-weight: bold;  
}
```

```
h2 {  
  display: block;  
  font-size: 1.5em;  
  margin-top: 0.83em;  
  margin-bottom: 0.83em;  
  margin-left: 0;  
  margin-right: 0;  
  font-weight: bold;  
}
```

```
h3 {  
  display: block;  
  font-size: 1.17em;  
  margin-top: 1em;  
  margin-bottom: 1em;  
  margin-left: 0;  
  margin-right: 0;  
  font-weight: bold;  
}
```

```
h4 {  
  display: block;  
  margin-top: 1.33em;  
  margin-bottom: 1.33em;  
  margin-left: 0;  
  margin-right: 0;  
  font-weight: bold;  
}
```

```
h5 {  
  display: block;  
  font-size: .83em;  
  margin-top: 1.67em;  
  margin-bottom: 1.67em;  
  margin-left: 0;  
  margin-right: 0;  
  font-weight: bold;  
}
```

```
h6 {  
  display: block;  
  font-size: .67em;  
  margin-top: 2.33em;  
  margin-bottom: 2.33em;  
  margin-left: 0;  
  margin-right: 0;  
  font-weight: bold;  
}
```


Herencia

- **inherit** para forzar la herencia, incluso con propiedades que no se heredan
 - No se debe abusar de inherit
- **herencia2.html** → **herencia2.css**. Se trata de hacer un menú horizontal.

Inicio	Lenguaje de Marcas	Desarrollo_Servidor	Desarrollo_Cliente	Aplicaciones_Web	
--------	--------------------	---------------------	--------------------	------------------	--

- Color de toda la lista: azul (**nav**)
- ¿Qué sucede si anulo la especificación color: **inherit**?
 - Se pone para anular la especificación del navegador por defecto para los enlaces (en color azul, como el fondo)

Cascada

- Conceptos principales que controlan el orden en el que se aplican las declaraciones de CSS:
 - Importancia.
 - Especificidad.
 - Orden en las fuentes

Cascada. Importancia

- Depende de dónde se ha especificado.
- Especificaciones contrapuestas → las nuevas anularán a las más antiguas en el siguiente orden:
 - Hoja de estilos de agente de usuario.
 - Declaraciones normales en hojas de estilo de usuario.
 - Declaraciones normales en hojas de estilo de autor.
 - Declaraciones importantes en hojas de estilo de autor.
 - Declaraciones importantes en hojas de estilo de usuario.

Cascada. Importancia.

- **Hoja de estilos de agente de usuario** ➔ hoja de estilo integrada del navegador.
 - P.e. enlaces no visitados (azules y subrayados).
- **Hoja de estilos de usuario** ➔ hoja de estilo que ha especificado el usuario. No compatibles con todos los navegadores.
 - P.e. estilo con fuentes y colores determinados para personas disléxicas.
 - P.e. Chrome ➔ Configuración ➔ Avanzadas ➔ Accesibilidad

Cascada. Importancia.

- **Hoja de estilo de autor** → creado o enlazado por el autor del documento o diseñador de la web.

- Normal
- Importante

→ Directiva **!important** →

```
font-family: "Times New Roman" !important;
```

- No hay que preocuparse demasiado por la importancia, no depende del autor.

Cascada. Especificidad

- Puede calcularse
- Cuatro componentes (a, b, c y d)
- **"a"**
 - 1 para una declaración en un atributo **style**
 - 0 si no es.
- **"b"**
 - Número de selectores de **id** en el selector (los que empiezan con #).
- **"c"**
 - Número de **selectores** de atributo, incluidos los selectores de clase y pseudoclases.
- **"d"**
 - Número de **tipo de elementos y pseudoelementos** del selector.

Cascada. Especificidad.

Selector	a	b	c	d	Especificidad
h1	0	0	0	1	0,0,0,1
.foo	0	0	1	0	0,0,1,0
#bar	0	1	0	0	0,1,0,0
html >head+body ul#menu *.home a:link	0	1	2	5	0,1,2,5

- Último ejemplo
 - a=0, no una declaración de un atributo style.
 - b=1, hay un selector ID (#menu).
 - c=2, hay un selector de atributos (.home) y una pseudoclase (:link)
 - d=5, hay cinco tipos de elemento (html, head, body, ul y a),
- Los combinadores (como >, + y el espacio en blanco) no afectan a la especificidad de un selector.
- El selector universal (*) tampoco cuenta para calcular la especificidad.

Cascada. Especificidad.

- **Ejemplo. `cascada1.html` y `cascada1.css`**
 - Un html con un título y dos párrafos.
 - Un css con el párrafo en cyan.
 - Identificamos el primer párrafo “especial”
 - Creamos una regla para especial: fondo rojo y color amarillo
 - Creamos una regla para el párrafo: fondo amarillo y color negro
 - Resultado.
 - `p` → 0001
 - `#especial` → 0100
 - → color rojo

Cascada. Orden de las fuentes.

- **Orden de las fuentes**

- Si dos reglas tienen la misma importancia y especificidad → orden de las fuentes
 - La declaración última va por delante de las anteriores
- Añadimos una especificación a **cascada1.css**
 - Para los párrafos: fondo amarillo y color negro
 - Observa los resultados.
 - Tienen la misma importancia y la misma especificidad (ya que el selector es el mismo) → orden de las fuentes.
 - Última regla específica → **color: black;** → anula **color: cyan;** de la regla anterior.
 - Esta regla nueva no afecta en absoluto al primer párrafo. Aunque la regla nueva aparece en último lugar, su selector tiene una especificidad más baja que la de #especial. Esto demuestra claramente cómo la especificidad tiene prioridad sobre el orden de las fuentes.

Ejercicios

- ¿Se puede heredar la propiedad **width**? ¿Tendría sentido?
- Si añadimos **!important** a la declaración **color: black;** de la última regla de la hoja de estilo de ejemplo, ¿afectará al color del texto del primer párrafo "especial"?
- ¿Qué selector es más específico, **#especial** o **html>head+body>h1+p**?
- ¿Qué apariencia debería tener una hoja de estilo de usuario para que nuestro documento de prueba aparezca con Helvética negra sobre fondo blanco, independientemente de la hoja de estilo del autor?