

Lenguaje de Marcas y Sistemas de Gestión

UA 2.8 - Introducción CSS: Aplicación de CSS



Centro Profesional
**Universidad
Europea**

Raúl Rodríguez Mercado

raul.rodriguez@universidadeuropea.es / @raulrodriguezue

Dpto. Ciencias y Tecnología de la Informática y Comunicación

UA 2.8: Introducción CSS: Aplicación de CSS



Objetivos

- Definir qué es CSS
- Conocer la estructura de las hojas de estilo y como se aplican a los documentos HTML



UA 2.8: Introducción CSS: Aplicación de CSS



Contenidos

- Estructura básica y características principales de CSS
- Prefijos de Navegadores
- Herencias y Cascadas



UA 2.8: Introducción CSS: Aplicación de CSS

Optimizar CSS

- Minimizar CSS para producción
- Borrar reglas innecesarias y duplicadas
- Ordenar las propiedades por orden alfabético
- CSS en la cabecera
- Si el CSS es muy grande partirlo en varios
- Organiza las reglas poniendo las que tengan relación juntas
- Documenta tus ficheros CSS

Herramientas

- <http://www.cssminifier.com/>
- <https://www.cssportal.com/css-optimize/>
- <http://unused-css.com/>



UA 2.8: Introducción CSS: Aplicación de CSS



Minificar CSS

Ejemplo de código CSS legible (index.css): 114 bytes

```
CSS #main {
    background-color: black;
    color: white;
    padding: 16px;
    margin: 4px;
    border: 2px solid blue;
}
```

Ejemplo de código CSS minificado (index.min.css): 86 bytes

```
CSS #main{background-color:#000;color:#fff;padding:16px;margin:4px;border:2px solid #00f}
```

- Si comparamos los dos archivos, vemos que al aplicar la minificación, el tamaño y la legibilidad del archivo se reduce considerablemente, así lo que se suele hacer es mantener un archivo legible (el primero) para realizar modificaciones y trabajar previamente con él, y luego, de forma opcional, un archivo con la información condensada (el segundo) que será el que se utilice en nuestro proyecto cuando esté terminado.

UA 2.8: Introducción CSS: Aplicación de CSS



Prefijos en CSS

- Son propiedades experimentales en los navegadores y a los que se le han añadido un prefijo para indicar que se les ha dado soporte antes de que sean soportadas por todos los demás.

VALORES COMUNES

- ▶ **-webkit-** (Chrome, Safari, nuevas versiones de Opera y todos los navegadores para iOS (Firefox tb))
- ▶ **-moz-** (Firefox)
- ▶ **-o-** (antiguas versiones de Opera)
- ▶ **-ms-** (Internet Explorer y Microsoft Edge)



```
a {  
  -webkit-transition: -webkit-transform  
  1s;  
  transition: -ms-transform 1s;  
  transition: transform 1s  
}
```

- ¿Cuándo Usarlo? Podemos comprobar si la propiedad de CSS3 es soportada en distintos navegadores en los siguientes enlaces:
 - ✓ caniuse.com → Web que indica qué propiedades son soportadas por el navegador
 - ✓ <http://shouldiprefix.com/> → Web que indica cuándo es necesario y donde usarlos prefijos
 - ✓ <https://github.com/postcss/autoprefixer> → Plugin para Visual Studio para utilizar prefijos.

UA 2.8: Introducción CSS: Aplicación de CSS

Prefijos en CSS

Ejemplo:

```
div {  
    transform: ... /* Navegador con especificación oficial */  
    -webkit-transform: ... /* Chrome, Safari, Opera... (Motor WebKit) */  
    -moz-transform: ... /* Mozilla Firefox */  
    -ms-transform: ... /* Microsoft Internet Explorer */  
    -o-transform: ... /* Antiguo opera */  
}
```

De esta forma, podemos utilizar varios prefijos para asegurarnos que aunque dichas funcionalidades tengan un comportamiento o sintaxis diferente, podemos hacer referencia a cada una de ellas por separado. Los navegadores que ya implimenten la funcionalidad **transform** harán lo que se define en la primera regla e ignorará el resto, y así con los casos siguientes.

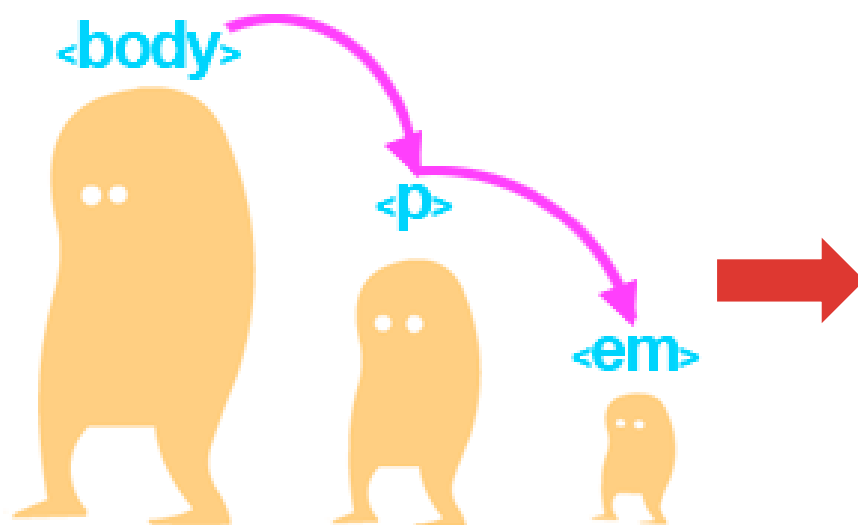


UA 2.8: Introducción CSS: Aplicación de CSS



Herencia en CSS

- La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un elemento padre se transmiten a sus hijos.
- Es muy similar a la herencia genética: si un padre es rubio, lo más seguro es que los hijos también puedan ser rubios...



```
<title>Ejemplo de herencia de estilos</title>
<style type="text/css">
  body { color: blue; }
</style>
</head>

<body>
  <h1>Titular de la página</h1>
  <p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```


UA 2.8: Introducción CSS: Aplicación de CSS



Ejemplo Herencia en CSS

```
<!Doctype html>
<html>
<head>
  <title>Ejemplo de herencia de estilos</title>
  <style type="text/css">
    body { color: blue; }
  </style>
</head>
<body>
  <h1>Prueba CSS</h1>
  <p>Un párrafo de texto no muy largo.</p>
  <div>
    <h2>Aquí vemos como el div también hereda las propiedades CSS</h2>
  </div>
</body>
</html>
```

Prueba CSS

Un párrafo de texto no muy largo.

Aquí vemos como el div también hereda las propiedades CSS

En este ejemplo, aplicamos a la etiqueta HTML **<body>** el **color de texto azul**. En principio, esta propiedad aplicará dicho color a los textos que estén dentro de dicha etiqueta **<body>**, sin embargo, si tenemos más etiquetas dentro, como por ejemplo, la **<div>** que le hemos introducido, también heredará esta propiedad. Esto ocurre porque la propiedad **color** es una de las propiedades CSS que hereda el valor de su elemento padre.

UA 2.8: Introducción CSS: Aplicación de CSS



Herencia en CSS

- En las propiedades donde solo se puede aplicar la herencia directamente son en **color** y en **font**
- Si queremos aplicar ciertos valores especiales comunes a la gran mayoría de propiedades, tenemos que aplicar los siguientes valores:

Valor	Significado
inherit	Hereda el valor de la propiedad del elemento padre.
initial	Establece el valor que tenía la propiedad inicialmente.
unset	Combinación de las dos anteriores. Hereda el valor de la propiedad del elemento padre, y en caso de no existir, de su valor inicial.

UA 2.8: Introducción CSS: Aplicación de CSS



Ejemplo Herencia en CSS

```
<!Doctype html>
<html>
<head>
  <title>Ejemplo de herencia de estilos</title>
  <style type="text/css">
    body {
      color:blue;
      border-width: 2px;
      border-style: solid;
      border-color: red;
    }
    div {
      border:inherit;
    }
  </style>
</head>
<body>
  <h1>Prueba CSS</h1>
  <p>Un párrafo de texto no muy largo.</p>
  <div>
    <h2>Aquí vemos como el div también hereda las propiedades CSS</h2>
  </div>
</body>
</html>
```

UA 2.8: Introducción CSS: Aplicación de CSS



Ejemplo Herencia en CSS

```
<!doctype html>
<html>
<head>
  <title>Ejemplo de herencia de estilos</title>
  <style type="text/css">
    body {
      color:blue;
      border-width: 2px;
      border-style: solid;
      border-color: red;
    }
    div {
      border:inherit;
    }
  </style>
</head>
<body>
  <h1>Prueba CSS</h1>
  <p>Un párrafo de texto no muy largo.</p>
  <div>
    <h2>Aquí vemos como el div también hereda las propiedades CSS</h2>
  </div>
</body>
</html>
```

Prueba CSS

Un párrafo de texto no muy largo.

Aquí vemos como el div también hereda las propiedades CSS

Si tenemos un elemento `<div>` dentro del `<body>`, el primero heredará los estilos del elemento `<body>`, ya que le hemos especificado el valor `inherit` en la propiedad `border`.

UA 2.8: Introducción CSS: Aplicación de CSS



Cascada en CSS

- Es el mecanismo que controla el resultado final cuando se aplican varias declaraciones CSS contrapuestas al mismo elemento.
- Antes de continuar, supongamos que nos encontramos ante el siguiente escenario:

```
div {  
  color:red;  
  padding:8px  
}
```

```
div {  
  color:blue;  
  background-color:grey  
}
```

PREGUNTA: *En este caso, ¿cuál de las dos reglas prevalece, si tenemos en cuenta que se refieren al mismo elemento y están al mismo nivel?*

RESPUESTA: *Prevalece siempre la última regla definida, la cuál mezcla y sobrescribe las propiedades anteriores.*

```
div {  
  color:blue;      /* Se sobrescribe la última */  
  padding:8px;  
  background-color:grey  
}
```

UA 2.8: Introducción CSS: Aplicación de CSS



Cascada en CSS

Orden de Aplicación de Declaraciones CSS

- Hay tres conceptos principales que controlan el orden en el que se aplican las declaraciones de CSS:
 - ✓ *Importancia*
 - ✓ *Especificidad*
 - ✓ *Orden en las fuentes*



UA 2.8: Introducción CSS: Aplicación de CSS



Cascada en CSS

Importancia

- La **importancia** de un código CSS se determina dependiendo de las hojas de estilo donde está colocado.
- Generalmente, no necesitaremos preocuparnos de este factor, pero siempre es una buena idea conocer como funciona. Existen varios tipos de hojas de estilo, de menor a mayor importancia:

Tipo de hojas de estilo	Descripción
Agente de usuario	Son los estilos CSS que aplica el navegador por defecto.
CSS de usuario	Son los estilos CSS que añade el usuario, por razones específicas.
CSS de autor	Son los estilos CSS que coloca el autor de la página.

UA 2.8: Introducción CSS: Aplicación de CSS



Cascada en CSS

Importancia

- Una **hoja de estilos de agente de usuario** es la hoja de estilo integrada del navegador. Cada navegador tiene sus propias reglas sobre cómo mostrar varios elementos de HTML si el usuario o diseñador de la página no especifica ningún estilo. Por ejemplo, los enlaces no visitados suelen ser azules y estar subrayados.
- Una **hoja de estilos de usuario** es una hoja de estilo que ha especificado el usuario. No todos los navegadores son compatibles con las hojas de estilo de usuario, pero pueden ser muy prácticas, sobre todo para los usuarios con determinados tipos de minusvalía. Por ejemplo, una persona disléxica o daltónica puede tener una hoja de estilo de usuario que especifique determinadas fuentes y colores que le faciliten la lectura.



UA 2.8: Introducción CSS: Aplicación de CSS



Cascada en CSS

Importancia

- Aunque no es recomendable utilizarlo frecuentemente (puede convertirse en una mala costumbre), se puede añadir al final de cada regla el texto **!important**, consiguiendo que la regla en cuestión tenga prioridad sobre las demás, independientemente del nivel o la altura a la que estén:

```
div {
  color:red!important;
  padding:8px
}

div {
  color:blue;
  background-color:grey
}
```



```
div {
  color:red;
  padding:8px;
  background-color:grey
}
```

NOTA: En el caso de que una misma propiedad del CSS de usuario y una propiedad del CSS de autor tuvieran **!important**, como caso excepcional tendría prioridad la del CSS de usuario sobre la del CSS de autor.

UA 2.8: Introducción CSS: Aplicación de CSS



Cascada en CSS

Especificidad

- En segundo caso, y si la importancia no elimina la ambigüedad, se pasa a determinar la especificidad de los selectores CSS. Para ello, se sigue un cálculo matemático basado en 4 componentes: **A,B,C,D**.

Componente	Descripción
Componente A	Número de estilos aplicados mediante un atributo <code>style</code> .
Componente B	Número de veces que aparece un <code>id</code> en el selector.
Componente C	Número de veces que aparece una <code>clase</code> , <code>pseudoclase</code> o <code>atributo</code> en el selector.
Componente D	Número de veces que aparece un <code>elemento</code> o un <code>pseudoelementos</code> en el selector.

UA 2.8: Introducción CSS: Aplicación de CSS



Cascada en CSS

Ejemplos de Especificidad (1)

Componente	Descripción
Componente A	Número de estilos aplicados mediante un atributo <code>style</code> .
Componente B	Número de veces que aparece un <code>id</code> en el selector.
Componente C	Número de veces que aparece una <code>clase</code> , <code>pseudoclase</code> o <code>atributo</code> en el selector.
Componente D	Número de veces que aparece un <code>elemento</code> o un <code>pseudoelementos</code> en el selector.

```
div { ... } /* Especificidad: 0,0,0,1 */
div div { ... } /* Especificidad: 0,0,0,2 */
#pagina div { ... } /* Especificidad: 0,1,0,1 */
#pagina div:hover { ... } /* Especificidad: 0,1,1,1 */
#pagina div:hover a { ... } /* Especificidad: 0,1,1,2 */
#pagina .sel:hover > a { ... } /* Especificidad: 0,1,2,1 */
```

UA 2.8: Introducción CSS: Aplicación de CSS



Cascada en CSS

Ejemplos de Especificidad (2)

- En <https://specificity.keegan.st/> hay una excelente "calculadora de especificidad" CSS donde podemos calcular la especificidad de un selector CSS de una manera rápida y cómoda.



UA 2.8: Introducción CSS: Aplicación de CSS



Cascada en CSS

Ejemplos de Especificidad (2)

Imaginar, que aplicamos esta regla a los personajes de Star Wars...

- ✓ 1º Asignamos a cada personaje (selector) una fuerza Sith (valor de especificidad) según su posición en el Lado Oscuro.
- ✓ De esta manera, tendríamos que un soldado de asalto imperial (Stormtroopers) tiene menos importancia y poder que el poderoso Darth Vader, y que este a su vez, es menos fuerte que el Emperador Palpatine, Maestro Sith...



Elemento Selector
Poder Sith
(especificidad): 0, 0, 1
(1)



Clase Selector
Poder Sith
(especificidad): 0, 1, 0
(10)



id selector
Poder Sith
(especificidad): 1, 0, 0
(100)

UA 2.8: Introducción CSS: Aplicación de CSS



Ejemplos de Especificidad: comprender como es su aplicación



a
1 x element selector
Sith: **0, 0, 1**



p a
2 x element selectors
Sith: **0, 0, 2**



.whatever
1 x class selector
Sith: **0, 1, 0**



a.whatever
1 x element selector
1 x class selector
Sith: **0, 1, 1**



p a.whatever
2 x element selectors
1 x class selector
Sith: **0, 1, 2**



.whatever .whatever
2 x class selectors
Sith: **0, 2, 0**



p.whatever a.whatever
2 x element selectors
2 x class selectors
Sith: **0, 2, 2**



#whatever
1 x id selector
Sith: **1, 0, 0**



a#whatever
1 x element selector
1 x id selector
Sith: **1, 0, 1**



.whatever a#whatever
1 x element selectors
1 x class selector
1 x id selector
Sith: **1, 1, 1**



.whatever .whatever #whatever
2 x class selectors
1 x id selector
Sith: **1, 2, 0**



#whatever #whatever
2 x id selectors
Sith: **2, 0, 0**

UA 2.8: Introducción CSS: Aplicación de CSS



Cascada en CSS

Orden

- En CSS, es posible crear múltiples reglas CSS para definir un mismo concepto.
- En este caso, la que prevalece ante todas las demás depende de ciertos factores, como es la «altura» a la que está colocada la regla:
 - ✓ El **CSS embebido** es el que tiene mayor precedencia, por lo que siempre será el que tenga prioridad sobre otras reglas CSS.
 - ✓ En segundo lugar, el **CSS interno** definido a través de bloques **style** en el propio documento HTML será el siguiente a tener en cuenta en orden de prioridad.
 - ✓ Por último, los documentos **CSS externos** son la tercera opción de prioridad a la hora de tomar en cuenta las reglas CSS.
- Teniendo esto en cuenta, hay que recordar que las propiedades que prevalecerán serán las que estén en último lugar, siempre respetando la prioridad de la lista anterior.



**Universidad
Europea**