



UNIVERSIDAD DE CÓRDOBA

PROGRAMACIÓN WEB – BLOQUE III.2

CSS_ Aspectos básicos y selección. Guía de diseño.

Dr. José Raúl Romero Salguero
jrromero@uco.es

{ CSS }



Contenidos

1. Aspectos básicos
2. Mecanismos de selección
3. Guía de diseño
4. Texto y colores
5. Fondos
6. El modelo de caja
7. Posicionamiento


1.

Aspectos básicos

Selectores y vínculos

CSS – *Cascading style sheets*

Se llama “hojas de estilo en cascada” (**Cascading style sheets**) porque las propiedades de un elemento se asignan en cascada en este **orden**:

- 
1. Estilos predeterminados del navegador
 2. Archivos CSS externos en una etiqueta `<link>`
 3. Hojas de estilo internas en una etiqueta `<style>` en el `<head>`
 4. Estilo en línea con el atributo `style` en un elemento HTML

Estándar CSS3 en W3C: <https://www.w3.org/TR/css3-roadmap/>

Tutorial CSS3 en W3Schools: <https://www.w3schools.com/css/>

Tutorial CSS3 en Mozilla Dev:

<https://developer.mozilla.org/es/docs/Archive/CSS3>

Enlace <link>

```
<head>
  ...
  <link href="filename" type="text/css" rel="stylesheet" />
  ...
</head>
```

- CSS describe la apariencia y el diseño de la información en una página web – a diferencia de HTML, que describe el contenido de la página
- Se puede incrustar en HTML o colocar en un archivo **.css** separado (¡¡opción preferentemente!!)

Hojas de estilo anidadas: `<style>`



```
<head>
  <style type="text/css">
    p { font-family: sans-serif; color: red; }
    h2 { background-color: yellow; }
  </style>
</head>
```

- Código CSS incrustado en el encabezado de una página HTML



Estilos *inline*: atributo `style`

```
<p style="font-family: sans-serif; color: red;">  
Esto es un párrafo</p>
```

Esto es un párrafo

- Mayor prioridad que estilos incrustados o vinculados
- Se utiliza para anulaciones únicas y para diseñar un elemento en particular: No utilizar

Comentarios CSS /* ... */

```
/* Esto es un comentario en CSS
   de bloque pero el único admitido */
p {
    color: red;
    background-color: aqua;
}
```

- **CSS** (igual que HTML) no se comenta tanto como código (p.ej. Java)
- No se puede utilizar **//** como comentario de línea en **CSS**
- No se admiten comentarios HTML **<!-- ... -->** en CSS

2.

Mecanismos de selección

Estableciendo reglas y orden

Sintaxis básica de reglas

```
selector {  
    property: value;  
    property: value;  
    ...  
    property: value;  
}
```

```
p {  
    font-family: sans-serif;  
    color: red;  
}
```

- Un archivo CSS consta de **una o más reglas**
- El **selector** de una regla especifica sobre qué elementos HTML se aplican las propiedades de estilo
- Un **selector *** selecciona todos los elementos

Sintaxis básica de reglas

```
p, h1, h2 {color: green;}  
h2 {background-color: yellow;}
```

Párrafo del estilo definido arriba.

h2 con estilo definido arriba.

- Se pueden seleccionar múltiples elementos separados por comas
- Además, elementos individuales también pueden tener sus estilos propios

Sintaxis básica de reglas__ conflictos

```
p, h1, h2 { color: blue; font-style: italic; }  
h2 { color: red; background-color: yellow; }
```

Párrafo utilizando los estilos de arriba

h2 utilizando los estilos definidos arriba.

Si dos estilos establecen **valores en conflicto** para la misma propiedad, **el último estilo tiene prioridad** (hojas en cascada)

Herencia de estilos

- Propiedades específicas de fuentes – p.ej. `font-family` o `color` se heredan
- **Ejemplo:** Probar a definir una propiedad color en `<body>`, y definir dentro un `<div>` → el `<div>` hereda esta propiedad
- **Motivo:** **Algunas** propiedades son heredadas por elementos internos
- Otras propiedades no tiene sentido que sean heredadas, como `background-color`, `border`

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

Herencia de estilos

```
body { font-family: sans-serif; background-color: yellow; }  
p { color: red; background-color: aqua; }  
a { text-decoration: underline; }  
h2 { font-weight: bold; text-align: center; }
```

Esto es la cabecera

Un párrafo estiloso. [Este enlace](#) está disponible en la web.

- Una lista no ordenada

- Cuando se aplican varios estilos a un elemento, se heredan
- Una regla puede anular una regla heredada más general
- **No todas las propiedades se heredan** (obsérvese el color del enlace arriba)

Selectores de ID (#)

```
#quijote {  
    font-style: italic;  
    font-family: "Garamond", "Century Gothic", serif;  
}
```

Don Quijote de la Mancha! [Don Quijote de la Mancha!](#)
En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor

- Se aplica el estilo solo al párrafo (o elemento) que tiene la ID
- En el ejemplo, el elemento HTML se puede especificar explícitamente:

```
p#quijote { ... }
```

Selectores de clase (.)

- Las clases son una forma de agrupar algunos elementos y dar un estilo solo a ese grupo (*"No quiero que TODOS los párrafos sean amarillos, solo estos tres ..."*)
- A diferencia de **id**, **class** se puede reutilizar tanto como desee en la página

Selectores de clase

```
.desc {                      /* cualquier elemento con class="desc" */  
  font-weight: bold;  
}  
  
p.especifica {              /* solo <p> con class="especifica" */  
  color: red;  
  font-family: cursive;  
}
```

El magnifico Don Quijote de la Mancha

Don Quijote correrá mil y una aventuras.

Su Escudero Sancho estará ahí para advertirle de los gigantes molinos

- Aplica la regla a cualquier elemento con clase `desc`, o a un `<p>` de clase `especifica`

Selectores de clase__ múltiples clases

```
.titulo {  
    background-color: yellow;  
    font-weight: bold;  
}  
.desc {  
    color: red;  
    font-family: cursive;  
}
```

Tomemos como ejemplo base esta hoja....

Selectores de clase__ múltiples clases

```
<h2 class="titulo">Don Quijote de la Mancha!</h2>
<p class="desc">Vea a Sancho junto al Quijote!</p>
<p class="titulo">Rocinante.</p>
<p class="desc">Se vende caballo por módico precio!</p>
```

Don Quijote de la Mancha!

Vea a Sancho junto al Quijote!

Rocinante.

Se vende caballo por módico precio!

Distintos elementos pueden ser miembros de la misma clase, y un elemento puede tener múltiples clases (separadas por espacios)

Selectores de contexto

- Aplica las propiedades dadas a selector2 solo si está dentro de un selector1 en algún punto de la página

```
selector1 selector2 {  
    properties  
}
```

- Aplica las propiedades dadas a selector2 sólo si se encuentra en el interior de un selector1 en la página (etiqueta selector2 está inmediatamente dentro selector1 sin etiquetas de por medio)

```
selector1 > selector2 {  
    properties  
}
```

Selectores de contexto__

ejemplo

```
<p>Lea el <strong>mejor libro de nuestra literatura</strong>  
universal:</p>  
<ul>  
  <li>Es un <strong>aguerrido</strong> aventurero.</li>  
  <li>Es un loco sin arreglo!</li>  
</ul>
```

```
li strong { text-decoration: underline; }
```

Lea el **mejor libro de nuestra literatura** universal:

- Es un **aguerrido** aventurero.
- Es un loco sin arreglo!

Selectores de contexto__ ejemplo

```
<div id="anuncio">
  <p>Lea el <strong>mejor libro de nuestra literatura</strong>
universal:</p>
<ul>
  <li class="importante">Es un <strong>aguerrido</strong>
aventurero.</li>
  <li>Es un <strong>loco sin arreglo!</strong></li>
</div>
```

```
#anuncio li.importante strong { text-decoration: underline; }
```

Lea el **mejor libro de nuestra literatura** universal:

- Es un **aguerrido** aventurero.
- Es un **loco sin arreglo!**

Selectores de contexto__ ejemplo

```
<div id="anuncio">  
  <p>Lea el <strong>mejor libro de nuestra literatura</strong>  
  universal:</p>  
<ul>  
  <li class="importante">Es un <strong>aguerrido</strong>  
  aventurero.</li>  
  <li>Es un <strong>loco sin arreglo!</strong></li>  
</div>
```

```
#anuncio > li.importante strong { text-decoration: underline; }
```



Recopilación de selectores CSS

➤ Selectores CSS comúnmente utilizados:

- ☐ Selector universal `* { }`
- ☐ Selector de tipo de elemento `h1, h2, h4 { }`
- ☐ Selector de clase `.note { }`
- ☐ Selector de ID `#listfriends { }`
- ☐ Selector hijo `li>a { }`
- ☐ Selector descendiente `p a { }`
- ☐ Selector de hermanos adyacente `h2+p { }`
- ☐ Selector de hermanos general `h1~p { }`

<https://www.w3schools.com/cssref/trysel.asp>

Mecanismo en cascada de CSS

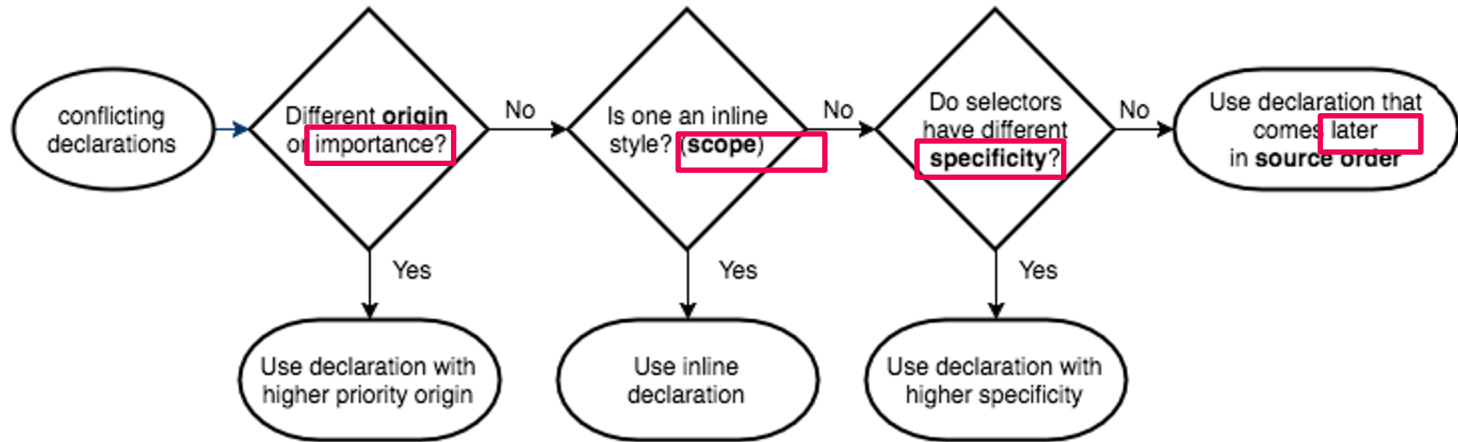
¿Qué prioridad se aplica al tener dos o más declaraciones para el mismo elemento?

- La última declaración
 - ❑ Si todos los selectores son idénticos, el último tiene prioridad
- Declaración más específica
 - ❑ Selector más específico tiene prioridad sobre más general
 - ❑ Por ejemplo, `h1` sobre `*`; `p b` sobre `p`
- Declaración “importante” (uso moderado o – preferiblemente – no usar)
 - ❑ Agregar **!important** después de cualquier valor de propiedad indica que debe considerarse más importante que otras declaraciones que se aplican al mismo elemento.

Se recomienda crear **primero las reglas genéricas** para la mayoría de los elementos y después sobrescribir las propiedades de individuos que necesiten una **apariciencia específica**

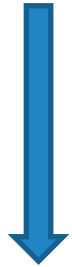
Mecanismo en cascada de CSS

CSS analiza (por orden) cuatro conceptos clave en su código CSS: **importancia**, **ámbito**, **especificidad** y **orden**



Mecanismo en cascada de CSS_ importancia

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, a modo de configuración global del navegador – para todas las páginas que visita
CSS de autor	Son los estilos CSS que coloca el autor de la página, esto es, diseñadores y desarrolladores

!important podría modificar este orden, dando prioridad al usuario

Mecanismo en cascada de CSS_ámbito

Un conjunto de estilos puede estar limitado a un elemento en particular y sus elementos secundarios.

Se debe tener precaución con el uso del atributo `style`

```
<a href="/ofertas" class="destacado" style="background-color: orange;">
Nuestras ofertas </a>
```

Mecanismo en cascada de CSS_ especificidad

Indica cuán específicos podemos ser al determinar los selectores cuyas reglas / declaraciones tienen prioridad

En general, **si la importancia y el ámbito no eliminan ambigüedad**, se aplican las siguientes **reglas de especificidad**:

más identificadores (HTML ID) en el selector, gana;
si empate: más clases, pseudoclasas o atributos en el selector, gana;
si empate: más nombres de elementos HTML en el selector, gana

Calculadora de especificidad: <https://specificity.keegan.st/>

Mecanismo en cascada de CSS_ especificidad

más identificadores (HTML ID) en el selector, gana; [A]

si empate: más clases, pseudoclases o atributos en el selector, gana; [B]

si empate: más nombres de elementos HTML en el selector, gana [C]

Selector	/* Especificidad: A,B,C */
div { ... }	/* Especificidad: 0,0,1 */
div div { ... }	/* Especificidad: 0,0,2 */
#pagina div { ... }	/* Especificidad: 1,0,1 */
#pagina div:hover { ... }	/* Especificidad: 1,1,1 */
#pagina div:hover a { ... }	/* Especificidad: 1,1,2 */
#pagina .sel:hover > a { ... }	/* Especificidad: 1,2,1 */

Mecanismo en cascada de CSS_orden

Si ninguna de las anteriores desambigua, cuán temprano o tarde aparece la declaración en la hoja de estilo servirá como último criterio

CSS embebido en el elemento HTML, gana; [A]
si empate: CSS interno en bloques <style> del documento, gana; [B]
si empate: CSS externos, gana [C]

Considerando el orden anterior, **la declaración que aparece más adelante en la hoja de estilo, o aparece en una hoja de estilo incluida más adelante en la página, gana**

3.

Guía de diseño

Comentarios generales

- HTML y CSS proporcionan muchas funciones que **deben usarse con criterio**
 - ~ solo porque exista una característica no significa que se use
- **No utilice funciones que distraigan** del contenido de la página
- Utilizar los **colores** y **fuentes** (no predeterminados) con cuidado:
 - ~ no inventar: p.ej. texto morado sobre fondo rosa
 - ~ no utilizar fuentes "extrañas" (que incluyen Comic Sans)
 - ~ aplicar principalmente una fuente oscura sobre un fondo claro
- Aproximadamente **el 8-10% de las personas tienen algún tipo de daltonismo o problema de visión**
 - ~ evitar combinaciones de color rojo / verde, fuentes demasiado pequeñas, etc.
- Algunas personas usan **lectores de pantalla** para leer el contenido
 - ~ siempre **¡debe favorecerse su integración y considerarlas!**

Comentarios generales

- Utilizar **unidades relativas** para tamaños de fuente, nunca *px* fijos
- Utilizar las **imágenes** apropiadamente.
 - ~ **evitar** imágenes de fondo brillantes que oculten el texto en 1^{er} plano
 - ~ **evitar** imágenes “clicables” en lugar de botones estándar para enlaces → **ralentizan la descarga de la página**
- **No existe un tamaño específico** de ventana o de fuente para el diseño, ya que el usuario lo podría cambiar
 - ~ use un diseño **adaptativo** o *responsive*
- Dividir una página grande en varias más pequeñas o proporcionar un menú adecuado para la navegación
- Utilizar **hojas de estilo** para realizar fácilmente cambios en el estilo y diseño
- Respetar las características estándar y probar todo en varios navegadores



Programación Web

Presentación de la asignatura__ Curso 2021/22