



Programación web

Prácticas

Semana 7: CSS

Aurora Ramírez Quesada (aramirez@uco.es)

Departamento de Ciencia de la Computación e Inteligencia Artificial

Universidad de Córdoba

Índice de contenido

1. Aspectos básicos

- Introducción
- Mecanismos de selección
- Guía de diseño

2. Formato

- Texto
- Color

3. Distribución espacial

- Modelo de caja
- Posicionamiento

4. Objetivos de la semana

Aspectos básicos

Introducción

- 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 llama “hojas de estilo en cascada” (**Cascading Style Sheets**) porque las propiedades de un elemento se asignan en este orden:
 1. Estilos predeterminados del navegador
 2. Archivos CSS externos (etiqueta <link>)
 3. Hojas de estilo internas (etiqueta <style> en <head>)
 4. Estilo en línea (atributo “**style**” en el elemento HTML)



El estándar **CSS** define un conjunto de propiedades **muy extenso**. Para aprender sobre elementos concretos y ver más ejemplos: <https://www.w3schools.com/css/>

Aspectos básicos

Introducción

- Archivos CSS externos (etiqueta <link>): **Opción preferente**

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

- Hoja de estilos anidada (etiqueta <style>)

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

- Estilo en línea (atributo style)

```
<p style="font-family: sans-serif;
color: red;>Párrafo en color rojo</p>
```



El estilo en línea es el de **mayor prioridad**. Se utiliza para diseños particulares, a nivel de un elemento concreto. **No se recomienda su utilización.**

Aspectos básicos

Mecanismos de selección

https://www.w3schools.com/cssref/css_selectors.php

- Los archivos CSS se componen de reglas sobre un elemento o tipo de elemento (**selector**) donde se indican las propiedades que se le aplican
- Una regla puede aplicarse sobre varios selectores (separar con comas)
- Los selectores pueden aparecer repetidos para indicar estilos propios
 - Si los dos estilos del elemento tienen **valores en conflicto** para una propiedad, **el último estilo definido es el que tiene prioridad** (“en cascada”)
- Tipos de selectores:
 - Elemento HTML: <p>, <h1>, <div>, etc.
 - Elemento único (atributo *id*): #welcome-message
 - Clase de elemento (atributo *class*): .notification
 - El selector * representa a todos los elementos

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

Aspectos básicos

Mecanismos de selección

https://www.w3schools.com/cssref/css_ref_combinators.php

- Existe **herencia** de estilos para algunas propiedades
 - Propiedades como la fuente o el color se heredan: si se definen a nivel de <body> o <div>, todos los elementos que estén contenidos dentro de esas etiquetas heredan el estilo
 - La herencia puede anularse si una regla sobreescribe el estilo de otra más general
- Los selectores de contexto aplican las propiedades a un selector según dónde aparezca respecto a otro selector:

Aplicar las propiedades a **selector2** solo si está dentro de **selector1** en algún punto de la página

```
selector1 selector2 {  
    property: value;  
    ...  
}
```

Aplicar las propiedades de **selector1** a **selector2** solo si **selector2** está dentro y justo a continuación de **selector1**

```
selector1 > selector2 {  
    property: value;  
    ...  
}
```

Aspectos básicos

Guía de diseño

- HTML y CSS proporcionan muchas propiedades que deben usarse con criterio
- El formato **no debe distraer** del contenido, sino apoyarlo para que se vea mejor
- Recomendaciones generales:
 - Utilizar **unidades relativas** para tamaños y diseño adaptativo (**responsive**)
 - Utilizar paletas de colores: https://www.w3schools.com/colors/colors_palettes.asp
 - Dividir páginas grandes en varias pequeñas o proporcionar menú de navegación
 - Respetar las características estándar y probar en varios navegadores
 - Considerar que existen usuarios con problemas de visión o que usan lectores de pantalla
- Aspectos que deben **evitarse**:
 - Combinaciones poco usuales (colores llamativos, fuentes extrañas, fondos oscuros)
 - Imágenes de fondo o para enlazar (ralentizan la descarga de la página)

Aspectos básicos

Guía de diseño

- Existen plantillas predefinidas que se ajustan bien a diferentes tipos o propósitos: páginas personales, blog, comercio electrónico, etc.
- Suelen ser de diseño adaptativo (*responsive*), lo cual mejora la experiencia de usuario
 - Facilita que el contenido se adapte bien a cualquier tipo de dispositivo y tamaño de pantalla
 - De forma dinámica, los elementos (como bloques o menús) se esconden, ensanchan o mueven
 - Más detalles:
 - https://www.w3schools.com/css/css_rwd_intro.asp
 - https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/CSS_layout/Responsive_Design
- Algunas plantillas disponibles:
 - W3Schools: https://www.w3schools.com/css/css_rwd_templates.asp
 - HTML5 UP: <https://html5up.net/>
 - Colorlib: <https://colorlib.com/wp/templates/>

Formato

Texto

- **Alineamiento de texto: horizontal o vertical**
- Alineamiento horizontal: **text-align**
 - Posibles valores: `left`, `right`, `center`, `justify`
- Alineamiento vertical: **vertical-align**
 - Alinea en relación con los elementos en línea y debe aplicarse al elemento vecino
 - Valores más comunes:
 - `baseline`
 - `length (%)`
 - `sub`
 - `super`
 - `top`
 - `text-top`
 - `middle`
 - `bottom`
 - `text-bottom`

https://www.w3schools.com/css/css_text_align.asp

Formato

Texto

- **Decoración de texto: línea, color, estilo y grosor**
- Tipo de decoración (línea): **text-decoration-line**
 - Posibles valores: `none`, `underline`, `overline`, `line-through`
- Color (asociado a línea): **text-decoration-color**
 - Los colores pueden escribirse con nombre (predeterminados) o código (RGB, HEX, HSL)
- Estilo de decoración (línea): **text-decoration-style**
 - Posibles valores: `solid`, `double`, `dotted`, `dashed`, `wavy`
- Grosor de la decoración (línea): **text-decoration-thickness**
 - Los valores pueden ser `auto`, expresados en píxeles o en porcentaje

https://www.w3schools.com/css/css_text_decoration.asp

Formato Texto



Es recomendable utilizar **medidas relativas** (em o porcentaje)

- **Indentación y espaciado de texto**
- Sangrado de la primera línea de texto: **text-indent**
 - Se expresa en píxeles o porcentaje. Acepta valores negativos
- Espaciado entre caracteres: **letter-spacing**
 - Se expresa en píxeles o unidad relativa. Acepta valores negativos
- Espaciado entre líneas: **line-height**
 - Se expresa como unidad relativa, y no puede ser negativo
- Espaciado entre palabras: **word-spacing**
 - Se expresa en píxeles o unidad relativa, y acepta valores negativos

https://www.w3schools.com/css/css_text_spacing.asp

https://www.w3schools.com/cssref/css_units.php

Formato

Texto

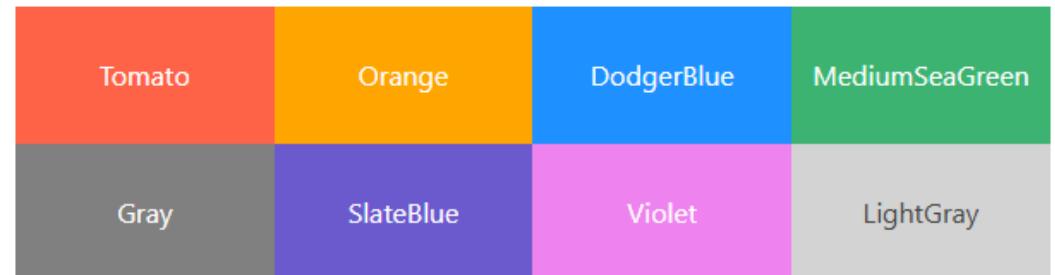
- **Fuente de texto: familia, estilo, tamaño, etc.**
- La fuente de texto tiene un impacto importante en la experiencia de lectura y da identidad al contenido de la aplicación
- Familia de fuente (general o específica): **font-family**
 - Pueden especificarse varias (disponibilidad): `Arial, sans-serif, "Times New Roman"`
- Estilo: **font-style** y **font-weight**
 - Valores posibles: `normal, italic, oblique` (*style*) / `normal, bold, lighter` (*weight*)
- Tamaño: **font-size**
 - Tamaños absolutos: `px, small, medium, large` (predeterminados en los navegadores)
 - Tamaños relativos: `em, rem, %, smaller, larger`

https://www.w3schools.com/css/css_font.asp

Formato

Color

- CSS/HTML soporta 140 colores con **nombres predeterminados**
- Para colores específicos, se puede utilizar la codificación RGB, HEX, HSL, RGBA o HSLA
 - Ejemplos: `rgb(255, 99, 71)`, `#ff6347`, `hsl(9, 100%, 64%)`
 - El componente “A” indica el porcentaje de transparencia (entre 0 y 1)
- Aplicable a distintos elementos
 - Fondo: **background-color**
 - Texto: **color**
 - Bordes: **border**



https://www.w3schools.com/css/css_colors.asp
https://www.w3schools.com/colors/colors_names.asp

`rgba(255, 99, 71, 0.5)`

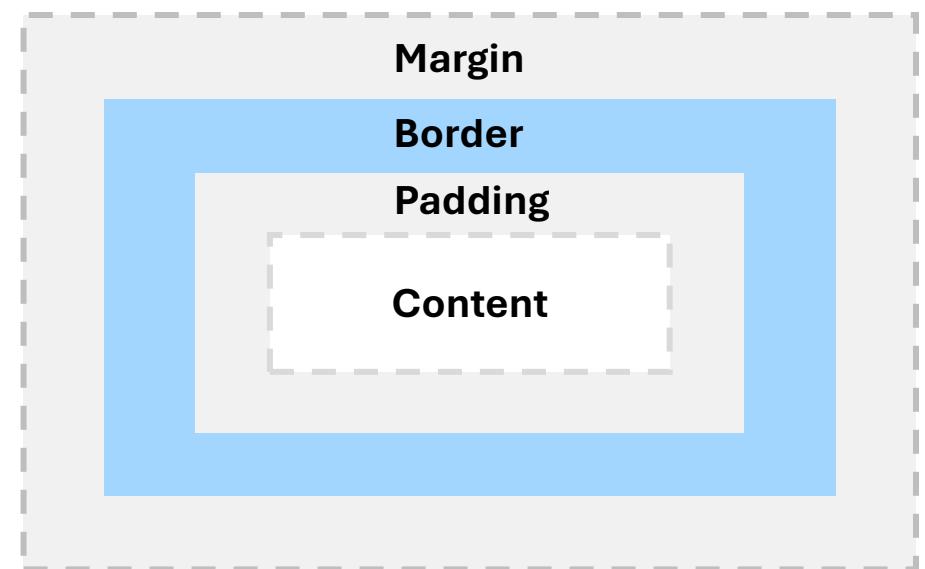
`hsla(9, 100%, 64%, 0.5)`

Distribución espacial

Modelo de caja

- Hace referencia a la “caja” que rodea a todo elemento HTML, y se compone de:
 - **Content**: donde aparece el contenido como texto o imágenes
 - **Padding**: área transparente alrededor del contenido
 - **Border**: borde entre “padding” y “margin”
 - **Margin**: área transparente externa al borde
- Permite añadir bordes y establecer espacio entre los elementos
- Deben tenerse en cuenta en las dimensiones:
 - **Width**: content-width + L/R-padding + L/R-border + L/R-margin
 - **Height**: content-height + T/B-padding + T/B-border + T/B-margin

https://www.w3schools.com/css/css_boxmodel.asp



Distribución espacial

Modelo de caja

- Propiedades para ajustar dimensiones:
 - Ancho y alto (para elementos de bloque): `width`, `height`
 - Dimensiones mínimas y máximas: `max-width`, `max-height`, `min-width`, `min-height`
- Propiedades aplicables al borde:
 - Admiten **de uno a cuatro valores** para indicar si debe aplicarse a todo el borde o por zonas: *top – right – bottom – left*
 - Estilo: `border-style` (`none`, `dotted`, `dashed`, `solid`, `double`, etc.)
 - Anchura: `border-width` (`thin`, `medium`, `thick` o expresado en px, tp, cm, em, etc.)
 - Color: `border-color` (`transparent`, por nombre, o con código HEX, RGB o HSL)

https://www.w3schools.com/css/css_dimension.asp

https://www.w3schools.com/css/css_border.asp

Distribución espacial

Modelo de caja

- Propiedades aplicables al margen:
 - En una única propiedad (de uno a cuatro valores): `margin`
 - Separando por cada zona: `margin-top`, `margin-right`, `margin-bottom`, `margin-left`
 - Valores permitidos: `auto` (predeterminado por el navegador), unidad (`px`, `pt`, `cm`, etc.), `%` (respecto al contenedor) o `inherit` (hereda del parente)
 - Si se quiere reducir, se utilizan valores negativos
- Propiedades aplicables al relleno (*padding*):
 - En una única propiedad (de uno a cuatro valores): `padding`
 - Separando cada zona: `padding-top`, `padding-right`, `padding-bottom`, `padding-left`
 - Comportamiento similar al margen, excepto valor `auto` (no definido)

https://www.w3schools.com/css/css_margin.asp

https://www.w3schools.com/css/css_padding.asp

Distribución espacial

Posicionamiento

- Propiedades que permiten modificar la disposición original de los elementos HTML
 - **position**: indica la posición exacta que debe ocupar el elemento
 - **overflow**: permite controlar qué hacer con elementos demasiado grandes para su área
 - **float**: permite desplazar al elemento dentro de su contenedor
- La posición es **estática por defecto**, pero puede tomar otros valores:
 - **relative**: relativa al documento, y puede desviarse con **top**, **right**, **bottom** y **left**
 - **fixed**: fija respecto al *viewport* (no cambia si se hace scroll), y admite las cuatro posiciones
 - **absolute**: relativa al ancestro más próximo (o al documento si no hay), y puede solapar
 - **sticky**: relativa a una posición de scroll (p. ej., para mantenerlo siempre arriba)

https://www.w3schools.com/css/css_positioning.asp

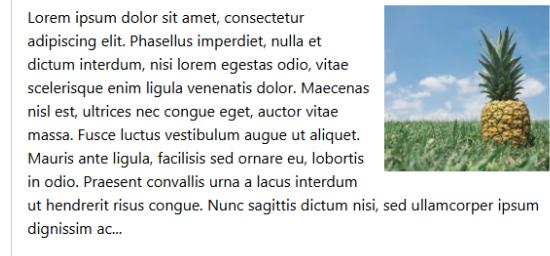
Distribución espacial

Posicionamiento

- Propiedad **overflow**
 - **visible**: el contenido que se sale de la caja del elemento continúa siendo visible
 - **hidden**: el contenido que no cabe en la caja del elemento queda oculto
 - **scroll**: se añaden barras de desplazamiento horizontal y vertical para ver el contenido
 - **auto**: Añade barra de desplazamiento solo cuando es necesario (horizontal o vertical)
 - Puede especificarse de forma diferente para cada dimensión: **overflow-x**, **overflow-y**
- Propiedad **float**
 - Uso habitual para ubicar imágenes al lado del texto, en lugar de en una nueva línea debajo
 - Valores posibles: **none**, **left**, **right**, **inherit**

https://www.w3schools.com/css/css_overflow.asp

https://www.w3schools.com/css/css_float.asp



Objetivos de la semana



1. Mejora las vistas de tu aplicación web:
 - Incorpora elementos de marcado estructural para mejorar la distribución de contenido
 - Asigna identificadores y/o clases a los elementos HTML para poder personalizar su estilo con CSS
2. Incorpora estilos CSS a tu aplicación web:
 - Puedes buscar una plantilla CSS y personalizarla, o crear una propia
 - Comprueba que los estilos se ajustan bien a distintos tamaños de pantalla (*responsive*)
3. Consulta la bibliografía recomendada:
 - Estándar CSS: <https://www.w3.org/TR/css/>
 - Tutorial CSS: <https://www.w3schools.com/css/>
 - Otros tutoriales: <https://developer.mozilla.org/es/docs/Web/CSS/Tutorials>