

Desarrollo de Aplicaciones Web

Práctica 4: CSS (1)

1. Objetivos

- Aprender el lenguaje de estilos CSS.
- Conocer algunas herramientas que ayudan a escribir CSS correcto y compatible con distintos navegadores.
- Aprender a maquetar una página web con CSS.

2. Recursos

¿Cómo se escribe una regla en CSS? ¿Qué propiedades y valores existen en CSS? ¿Cómo se selecciona un elemento de una página web para aplicarle una regla de CSS?

- **W3Schools**¹: cursos de aprendizaje y guías de referencia de diversas tecnologías empleadas en la programación web.
- **W3C**²: organismo internacional que desarrolla las especificaciones de las principales tecnologías que se emplean en la programación web.
- **Cascading Style Sheets (CSS) Quick Reference**³: resumen en una página de lo más importante de CSS.
- **CSS Cheat Sheet**⁴: resumen en una sola cara de lo más importante de CSS. También está traducida al español⁵.
- **CSS 2.1 and CSS 3 Help Cheat Sheets**⁶: resumen de CSS 2.1 y de las nuevas características de CSS 3 realizado para de la famosa revista Smashing Magazine. Es del año 2010, así que seguramente faltan algunas de las últimas novedades.

¿Cómo puedo saber que el código CSS que he escrito es correcto?

- **W3C CSS Validation Service**⁷: servicio de validación de CSS. Se puede indicar una página web para que valide los CSS que contiene o directamente un CSS.

¿Cómo se representan los colores en CSS? Para trabajar con los colores necesitas conocer la codificación RGB (red, green, blue):

- **Colores HTML**⁸: explica la codificación RGB empleada en HTML y CSS.
- **Anexo: Colores HTML**⁹: contiene una tabla con los nombres y los códigos hexadecimales de diferentes colores.

¹<http://www.w3schools.com>

²<http://www.w3.org>

³<http://www.digilife.be/quickreferences/QR/Cascading%20Style%20Sheets%201.0.pdf>

⁴<http://www.addedbytes.com/cheat-sheets/css-cheat-sheet/>

⁵<http://www.webtutoriales.com/tutoriales/html-css/hoja-referencia-css.26.html>

⁶<http://www.smashingmagazine.com/2010/05/13/css-2-1-and-css-3-help-cheat-sheets-pdf/>

⁷<http://jigsaw.w3.org/css-validator/>

⁸http://es.wikipedia.org/wiki/Colores_HTML

⁹http://es.wikipedia.org/wiki/Anexo:Colores_HTML

¿Qué combinaciones de colores son correctas? Los colores armónicos son aquellos que funcionan bien juntos, es decir, que producen un esquema de color atractivo a la vista. Algunas páginas donde puedes encontrar combinaciones de colores armónicos:

- **El uso de los colores en la Web**¹⁰: analiza el número de colores que debe emplearse en una página web, explica las diferentes relaciones que pueden existir entre los colores (análogos, complementarios, triada, etc.) y define el contraste entre dos colores.
- **ColorCombos**¹¹: posee una librería de combinaciones de colores con diferente número de colores. Permite buscar una combinación que utilice un color deseado. También permite obtener, mediante la introducción de una URL, los colores empleados en una página web.

¿Existen plantillas de CSS que pueda emplear en mi sitio web? Existen multitud de sitios web donde se pueden encontrar plantillas gratuitas y de pago. Algunos sitios que ofrecen plantillas gratuitas son:

- **free CSS templates**¹²: cientos de plantillas CSS gratuitas.
- **Free CSS Templates**¹³: cientos de plantillas CSS gratuitas.

¿Existe alguna herramienta que me pueda ayudar a escribir el código CSS?

- **Notepad++**¹⁴: editor gratuito de código fuente que soporta varios lenguajes de programación. Entre otras características, posee sintaxis coloreada, envoltura de sintaxis y autocompletado.
- **Sublime Text**¹⁵: excelente editor compatible con múltiples lenguajes de programación y disponible para Windows, OS X y Linux; se puede probar de forma gratuita, pero para un uso continuo hay que adquirir una licencia.
- **EclipseStyle**¹⁶: editor gratuito de CSS.
- **CSSMate**¹⁷: editor de CSS en página web.

¿Cómo se maqueta con CSS?

- **Porqué el diseñar con tablas es estúpido: problemas definidos, soluciones ofrecidas**¹⁸: explica porque no hay que diseñar con tablas y las ventajas de emplear CSS.
- **Simple 2 column CSS layout**¹⁹: explica paso por paso cómo se crea una página web para que tenga un diseño en 2 columnas.
- **CSS Layouts**²⁰: proporciona diferentes estilos de diseño (2 columnas, 3 columnas, fluido y otros).
- **CSS Layout Techniques: for Fun and Profit**²¹: proporciona diferentes estilos de diseño.
- **Sample CSS Page Layouts**²²: explica paso a paso como lograr diferentes estilos de diseño.

¿Cómo se maqueta un formulario con CSS?

- **Accessible CSS Forms: Using CSS to Create a Two-Column Layout**²³: explica paso a paso como maquetar un formulario con CSS.
- **How to Style Forms in CSS**²⁴: explica paso a paso como maquetar un formulario con CSS.

¹⁰<http://www.webusable.com/coloursMix.htm>

¹¹<http://www.colorcombos.com/>

¹²<http://www.freecsstemplates.org/>

¹³<http://www.free-css-templates.com/>

¹⁴<http://notepad-plus.sourceforge.net/es/site.htm>

¹⁵<http://www.sublimetext.com/>

¹⁶<http://www.greeneclipse.com/eclipsestyle.html>

¹⁷<http://cssmate.com/csseditor.htm>

¹⁸http://www.effectivetranslations.com/stupidtables/everything_es.html

¹⁹http://www.456bereastreet.com/lab/developing_with_web_standards/csslayout/2-col/

²⁰<http://css-discuss.incutio.com/?page=CssLayouts>

²¹<http://www.glish.com/css/>

²²http://www.maxdesign.com.au/presentation/page_layouts/

²³<http://www.websiteoptimization.com/speed/tweak/forms/>

²⁴<http://www.informit.com/articles/article.aspx?p=456144>

- **Practical CSS Layout Tips, Tricks, & Techniques**²⁵: algunos trucos de maquetación con CSS, explica un método de alinear los controles de un formulario.
- **CSS-Only, Table-less Forms**²⁶: muestra un ejemplo muy completo de formulario, con todo lo que puede tener, pero también es muy complejo.

¿Cómo se maqueta una tabla con CSS?

- **CSS Table Gallery**²⁷: plantillas para maquetar tablas con CSS.

¿Existe alguna herramienta que me pueda ayudar a escribir el código CSS necesario para maquetar una página web?

- **CSS Layout Generator**²⁸: generador del código CSS para maquetar.
- **CSS Layout Generator**²⁹: generador del código CSS para maquetar.
- **CSS Layout Generator**³⁰: generador del código CSS para maquetar.

3. ¿Qué tengo que hacer?

En esta práctica tienes que crear el estilo visual de las páginas web de la práctica anterior y del resto de páginas que harás en las próximas prácticas. Como mínimo, tienes que emplear las siguientes propiedades de CSS:

- **font**: para definir las propiedades del texto.
- **background**: para definir las propiedades del fondo de un elemento.
- **border**: para definir las propiedades del borde de un elemento.
- **margin**: para definir la distancia que debe existir entre un elemento y los elementos que lo rodean.
- **padding**: para definir la distancia que debe existir entre el borde de un elemento y su contenido.

Además, define al menos un par clases para definir el estilo particular de algunos elementos, como por ejemplo, `.anuncio` para aplicarlo a un párrafo que define un anuncio o `.conBorde` para aplicarlo a aquellos elementos que tienen que aparecer con un borde.

En esta práctica debes realizar la maquetación (*layout*) de la página. Según el **Diccionario de la Lengua Española**³¹ de la Real Academia Española, maquetar es “Hacer la maqueta de una publicación que se va a imprimir”. Si buscamos maqueta, obtenemos varias acepciones y una de ellas dice “Boceto previo de la composición de un texto que se va a publicar, usado para determinar sus características definitivas”.

En el desarrollo de las páginas web, se entiende por maquetación a la acción de distribuir o posicionar los distintos elementos que queremos que aparezcan en una página web. En especial, haz una correcta maquetación de los controles de los formularios para que aparezcan alineados.

Intenta que el diseño sea fluido (evita trabajar con tamaños fijos) para que se adapte correctamente a distintas resoluciones de pantalla. Sin embargo, **en esta práctica no debes realizar un diseño adaptativo (*responsive design*) de la página, eso lo harás en la siguiente práctica. No debes utilizar las *media queries* (el atributo `media` de HTML y la regla `@media` de CSS).**

²⁵<http://www.alistapart.com/stories/practicalcss/>

²⁶<http://jeffhowden.com/code/css/forms/>

²⁷<http://icant.co.uk/csstablegallery/>

²⁸<http://csslayoutgenerator.com/>

²⁹<http://csscreator.com/version2/pagelayout.php>

³⁰<http://www.cssportal.com/layout-generator/>

³¹<http://www.rae.es/>

4. ¿Cómo lo hago?

Para definir el estilo visual tienes que emplear el lenguaje de estilos CSS. No tienes que crear un fichero distinto para cada página, sino que el mismo fichero lo utilizarás en todas las páginas, a no ser que existan algunas páginas con estilos visuales muy diferentes (por ejemplo, la página de inicio, la página principal y el resto de páginas).

Tienes varias formas de crear una hoja de estilo CSS:

1. Crear una hoja CSS desde cero: esto lo podrás hacer cuando ya tengas un conocimiento amplio del lenguaje CSS.
2. Tomar un CSS como “inspiración” para crear un CSS propio. En este caso, es una buena práctica citar al autor de la o de las hojas CSS en que te has basado, pero si existen grandes diferencias no es del todo necesario. La cita la puedes indicar mediante un comentario en la misma hoja CSS.
3. Adaptar una hoja CSS a la estructura de tu página o, al revés, adaptar la estructura de tu página a una hoja CSS. Cuando usas directamente el CSS de otra persona, primero tienes que consultar si es posible hacerlo, es decir, si el autor de la hoja CSS te da permiso. Lo normal es que te pida (y si no lo hace, hazlo) que cites claramente al autor del CSS en la página web, pero a veces con un simple comentario en la misma hoja CSS puede ser suficiente.

Si quieres escribir directamente el código CSS, cualquier editor de texto sencillo como el **Bloc de notas** es suficiente. Sin embargo, puedes emplear un programa como **Notepad++**³², que posee algunas características que ayudan a escribir el código, como la sintaxis coloreada o la función de autocompletado (pulsa Ctrl+Espacio).

Si prefieres emplear una herramienta que te permita crear un CSS de forma visual, emplea **EclipseStyle**³³, que evita tener que conocer las propiedades y valores de CSS.

También existen herramientas online, como **CSS Generator**³⁴ y **CSS Style Editor**³⁵ que permiten escribir un CSS sin conocer el nombre de las propiedades y valores.

Para indicar que se está usando un fichero CSS en una página web se emplea la etiqueta `<link />`. Es una misma página web se pueden indicar varios ficheros CSS que se combinan todos ellos o también se puede indicar de forma que definan estilos alternativos. Por ejemplo, en el siguiente código se define un estilo CSS que se debe emplear cuando se imprima una página web (**print**) y un estilo que se debe emplear en el resto de casos:

```
<link rel="stylesheet" type="text/css" href="home.css" />
<link rel="stylesheet" type="text/css" href="print.css" media="print" />
```

Para maquetar con CSS un formulario existen varias técnicas. Una sencilla consiste en definir un tamaño fijo para las etiquetas de los controles. Por ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8" />
<title>Formulario alineado</title>
<style>
fieldset {width: 300px; padding: 10px;}
legend {font-weight: bold;}
label {float: left; display: block; width: 100px;}
input {float: none; display: inline;}
.centrado {margin: 10px 100px;}
</style>
</head>
<body>
<form id="form1" action="" method="get">
<fieldset>
<legend>Datos personales</legend>
<label for="nombre">Nombre:</label>
```

³²<http://notepad-plus.sourceforge.net/es/site.htm>

³³<http://www.greeneclipse.com/eclipsestyle.html>

³⁴<http://csscreator.com/tools/cssgenerate>

³⁵<http://www.cssportal.com/css-style-editor/>

```

<input type="text" id="nombre" />
<br />
<label for="apellidos">Apellidos:</label>
<input type="text" id="apellidos" />
<input type="submit" value="Enviar" class="centrado" />
</fieldset>
</form>
</body>
</html>

```

Aunque en el ejemplo anterior se han escrito las reglas de CSS directamente en la página web con la etiqueta `<style>`, recuerda que lo correcto es escribirlas en un fichero separado y enlazarlo con la etiqueta `<link />`.

En la Figura 1 podemos observar como se visualiza esta página en Microsoft Internet Explorer 7:

Figura 1: Formulario de ejemplo

Sin embargo, un tamaño fijo no es la mejor opción. En **How to structure an HTML form**³⁶ y **Styling HTML forms**³⁷ se explican algunas técnicas básicas de maquetación de formularios.

5. Recomendaciones

El estilo que definas no debe ser muy simple (no vale con cambiar el color de fondo de la página y definir un estilo para el texto), pero tampoco es necesario que sea muy complejo. Con unas pocas reglas de CSS bien elegidas se puede lograr un estilo visual impactante.

En general, se recomienda que como mínimo se utilicen tres colores diferentes (y suele ser lo suficiente en la mayoría de los casos) y no más de seis.

En general, se recomienda que se utilicen dos tipos de letra (un tipo de letra para los títulos y otro para el contenido principal) y como máximo tres (un tipo de letra para contenido adicional, como puede ser una barra lateral).

Una mala práctica es definir múltiples clases (`class`) e identificadores (`id`). Siempre que sea posible, emplea selectores simples que hagan referencia a elementos de HTML.

Recuerda que hay propiedades que se heredan, por lo que no es necesario definir las para todos los elementos, sino que si se define en un elemento padre, todos sus hijos la tomarán. Por ejemplo, para aplicar un mismo estilo de texto en toda una página web, lo más sencillo es definirlo en el elemento `html` o `body`.

Un fichero CSS puede llegar a tener cientos o incluso miles de líneas, por lo que pueden aparecer problemas importantes:

- Reglas no utilizadas.
- Reglas duplicadas, que indican lo mismo o son contradictorias.
- Propiedades heredadas que pasan desapercibidas.
- Por supuesto, el tiempo que se pierde para localizar una regla entre miles de líneas.

Para evitar estos problemas, es necesario organizar un fichero CSS de alguna forma. Cada uno tiene que desarrollar su estilo propio, con el que se sienta más a gusto. Algunas posibilidades son:

- Organizar las reglas por orden alfabético. Por ejemplo: `a`, `div`, `em`, `.importante`, `#principal`, `table`.

³⁶https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms/How_to_structure_an_HTML_form

³⁷https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms/Styling_HTML_forms

- Organizar las reglas por orden de aparición. Por ejemplo: `html`, `body`, `h1`, `h2`, `div`, `#principal`, `p`, `.importante`.
- Separar las reglas según la función que realizan. Por ejemplo: por un lado las reglas que modifican los bordes, por otro lado las reglas que modifican los colores, por otro lado las reglas que definen la maquetación.
- Organizar las propiedades de una regla por orden alfabético. Por ejemplo: `border`, `font-size`, `margin`, `padding`, `text-align`.
- Organizar las propiedades de una regla del exterior del elemento al interior. Por ejemplo: `margin`, `border`, `padding`, `text-align`, `font-size`.

También existen algunas herramientas que te pueden ayudar a organizar un fichero CSS. **Format CSS Code**³⁸ es una herramienta online que formatea el aspecto de un CSS: introduce saltos de línea, espacios en blanco, etc. **CSS Optimizer**³⁹ es otra herramienta online que ayuda a reducir el tamaño de un fichero CSS al aplicar algunas transformaciones, como por ejemplo, cambiar el nombre de algunas propiedades y valores.

Utiliza los comentarios de CSS `/* . . . */` para separar las diferentes partes de un CSS y para desactivar ciertas propiedades cuando quieras realizar pruebas.

Y recuerda: evita la divitis y la spanmania.

³⁸<http://www.cssportal.com/format-css/>

³⁹<http://www.cssportal.com/css-optimize/>