

Prácticas

Estilo: ▼ Tamaño letra: ▼ Interlineado: ▼

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.
- Conocer los conceptos básicos del diseño adaptativo.
- Aprender a crear un estilo adaptativo.
- Aprender a utilizar una biblioteca de iconos.

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 español[\(5\)](#).
- **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.

¿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 maquetar 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 maquetar 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.
- **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 maquetar 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.

¿Qué es el diseño adaptativo?

- **Diseño web responsivo, sensible, adaptable, ...[\(31\)](#)**: ¿cómo se dice *responsive web design* en español?
- **Responsive Web Design**[\(32\)](#): artículo seminal en el que Ethan Marcotte acuñó el término *responsive web design*.
- **Multi-Device Web Design: An Evolution**[\(33\)](#): análisis del diseño adaptativo actual y futuro.
- **Responsive Web Design: Missing the Point**[\(34\)](#): discusión sobre la importancia actual y futura del diseño adaptativo.

¿Cómo se realiza un diseño adaptativo?

- **CSS Media Queries & Using Available Space**[\(35\)](#): técnicas básicas para el diseño adaptativo.
- **Responsive Web Design Techniques, Tools and Design Strategies**[\(36\)](#): colección de técnicas que se aplican en el diseño adaptativo.
- **Using media queries**[\(37\)](#): sintaxis y operadores que se pueden emplear en las *media queries*.
- **CSS3 @media Rule**[\(38\)](#): referencia de la sintaxis de las *media queries*.

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:

- **color**: para definir el color del texto.
- **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**[\(39\)](#) 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.

¿Hoy en día se accede a la Web exclusivamente desde un ordenador? No, es muy normal acceder a la Web desde un dispositivo móvil como un teléfono inteligente (*smartphone*) y es muy probable que en el futuro se acceda también desde otros tipos de dispositivos que en la actualidad no contemplamos. En realidad, según StatCounter, entre octubre y noviembre de 2016[\(40\)](#), el acceso a la Web mediante teléfono móvil superó al acceso mediante ordenador, tal como se puede observar en la Figura 1, donde la línea azul representa el acceso mediante ordenador y la línea verde representa el acceso mediante teléfono móvil. Este cambio de tendencia se mantiene hasta la actualidad.

Desktop vs Mobile vs Tablet Market Share Worldwide

May 2016 - Apr 2017

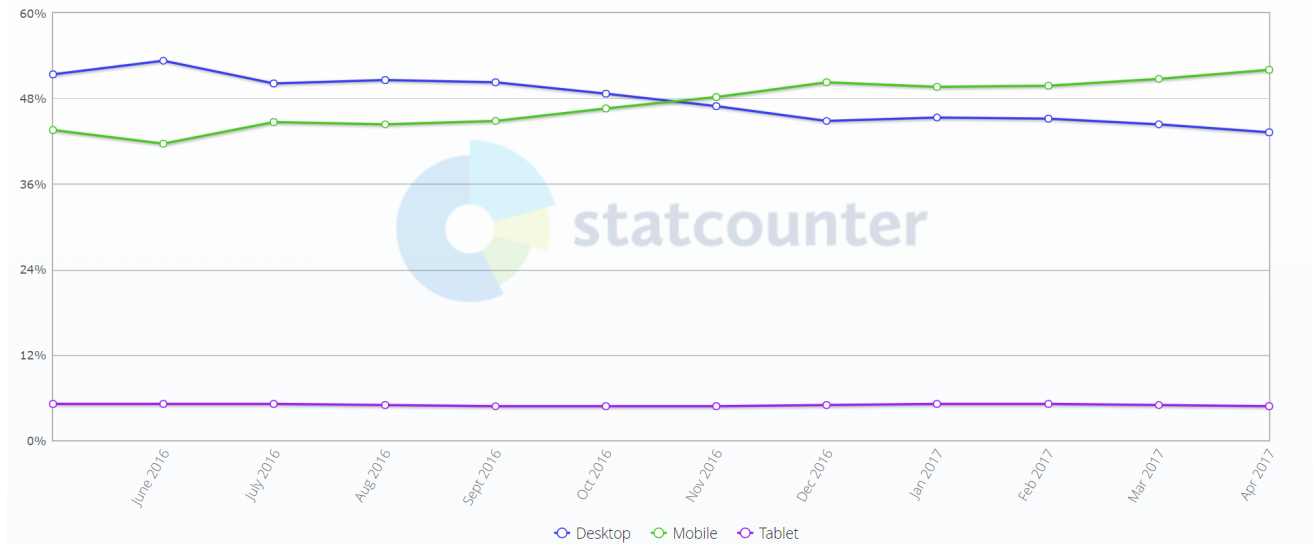


Figura 1: Estadísticas de acceso a la Web entre mayo 2016 y abril 2017 según StatCounter

Por tanto, tienes que **crear una hoja de estilo que incluya un diseño adaptativo** (*responsive design*), que tu sitio web se visualice de forma adecuada en un dispositivo con una pantalla de baja resolución (480px de ancho) o de alta resolución (1920px de ancho). Puedes definir todos los puntos de ruptura (*breakpoints*) que consideres necesarios.

4. ¿Cómo lo hago?

4.1. Cómo empezar con CSS

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++** ([41](#)), 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** [\(42\)](#), que evita tener que conocer las propiedades y valores de CSS.

También existen herramientas online, como **CSS Generator** [\(43\)](#) y **CSS Style Editor** [\(44\)](#) 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 />`. En 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 (cuando no se indica el atributo `media` de forma explícita, este toma el valor `all` que significa que se aplica a cualquier dispositivo):

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

En el ejemplo anterior aparece el atributo `type="text/css"` que ya no es necesario ponerlo en HTML5, pero que lo puedes encontrar en muchos ejemplos.

4.2. Variables CSS

Las variables CSS, oficialmente conocidas como *CSS Custom Properties* [\(45\)](#), son un mecanismo que permite escribir hojas de estilo más fáciles de mantener. El primer borrador de la especificación oficial se publicó en el año 2012 [\(46\)](#), pero no fue hasta el año 2016 que la mayoría de los navegadores web implementaron las variables CSS. En la Figura 2 podemos ver el soporte actual (octubre 2019) de las variables CSS en los principales navegadores web. Solo Microsoft Internet Explorer y algunos navegadores minoritarios no soportan esta característica de CSS.

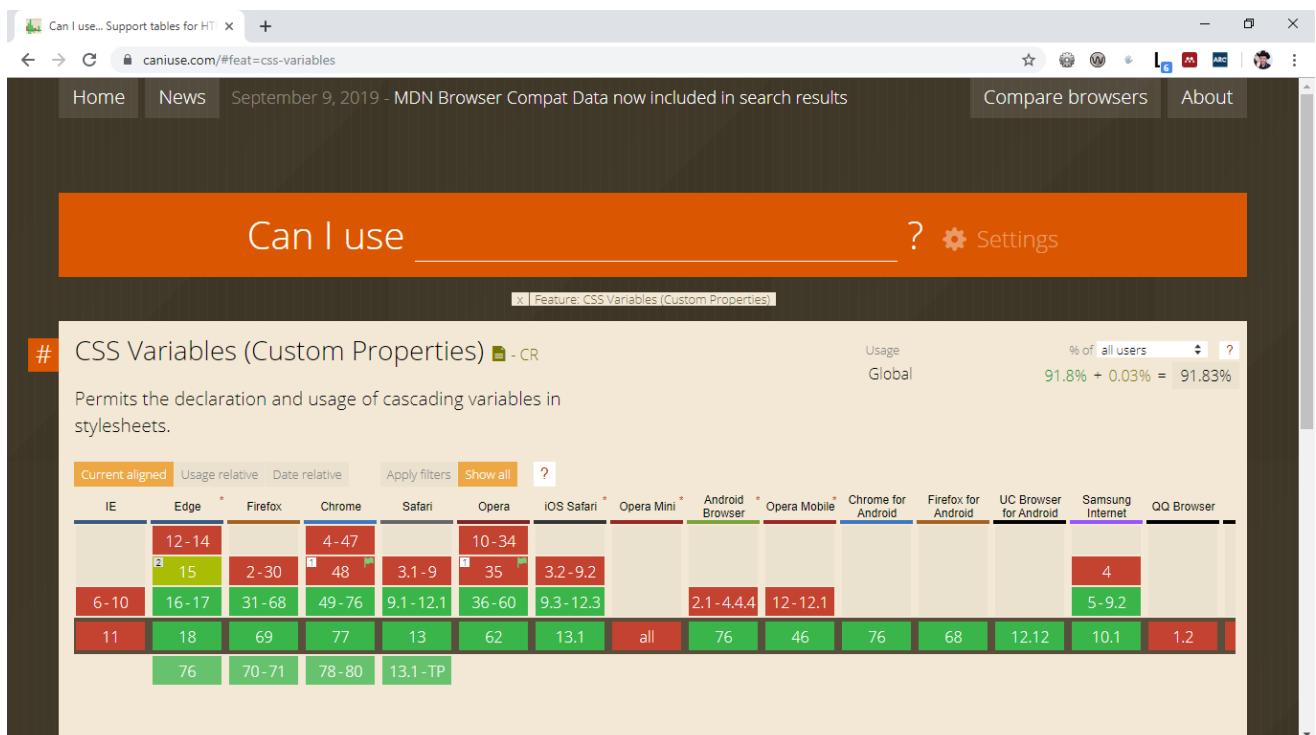


Figura 2: Can I use CSS Variables?

Las variables se declaran con el prefijo `--` (dos guiones) dentro de un selector. El selector `:root` permite declarar las variables en un ámbito global para que estén disponibles dentro de cualquier otro selector. Para utilizar una variable se emplea `var()` y entre paréntesis se escribe el nombre de la variable que se quiere utilizar.

Por ejemplo, el siguiente código define las variables `color1` y `color2` que se emplean para definir el color de dos párrafos:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8" />
<title>Variables CSS</title>
<style>
:root {
  --color1: blue;
  --color2: yellow;
}

#par1 {
  background-color: var(--color1);
  color: var(--color2);
}

#par2 {
  background-color: var(--color2);
  color: var(--color1);
}
</style>
</head>
<body>
<p id="par1">Este párrafo debería tener texto amarillo sobre fondo azul.</p>
<p id="par2">Este párrafo debería tener texto azul sobre fondo amarillo.</p>
</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 3 se puede ver el resultado del código anterior en un navegador web.

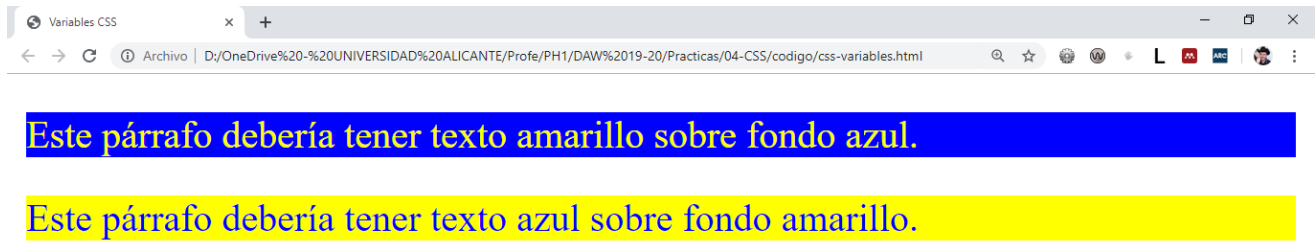


Figura 3: Ejemplo de uso de las variables CSS

En el siguiente ejemplo, dentro del selector `#contenedor` se redefine la variable `color1` que ha sido definido en el ámbito global. Como resultado de este cambio, los dos últimos párrafos no muestran los mismos colores que los dos primeros párrafos, aunque compartan las mismas reglas de CSS. Este ejemplo muestra que los mecanismos de la herencia y la cascada siguen funcionando con las variables.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8" />
<title>Variables CSS</title>
<style>
:root {
  --color1: blue;
  --color2: yellow;
}

#par1, #par3 {
  background-color: var(--color1);
  color: var(--color2);
}

#par2, #par4 {
  background-color: var(--color2);
  color: var(--color1);
}

#contenedor {
  --color1: green;
}
</style>
</head>
<body>
<p id="par1">Este párrafo debería tener texto amarillo sobre fondo azul.</p>
<p id="par2">Este párrafo debería tener texto azul sobre fondo amarillo.</p>
<div id="contenedor">
  <p id="par3">Este párrafo debería tener texto amarillo sobre fondo verde.</p>
  <p id="par4">Este párrafo debería tener texto verde sobre fondo amarillo.</p>
</div>
</body>
</html>
```

```

</div>
</body>
</html>

```

En la Figura 4 se puede ver el resultado del código anterior en un navegador web.

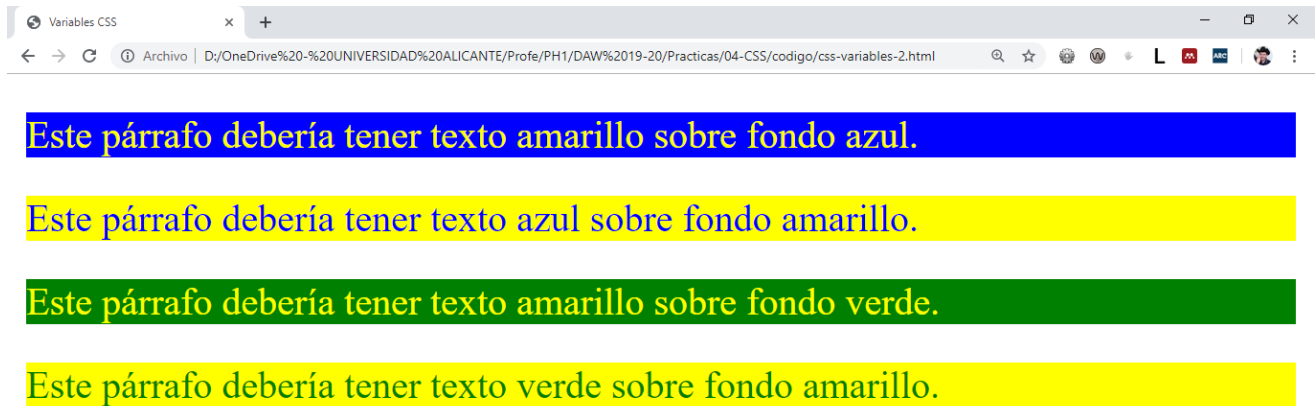


Figura 4: Ejemplo de uso de las variables CSS

4.3. Maquetación de un formulario

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: block;}
.centrado {margin: 10px auto;}
</style>
</head>
<body>
<form id="form1" action="" method="get">
<fieldset>
<legend>Datos personales</legend>
<label for="nombre">Nombre:</label>
<input type="text" id="nombre" />
<br />
<label for="apellidos">Apellidos:</label>
<input type="text" id="apellidos" />

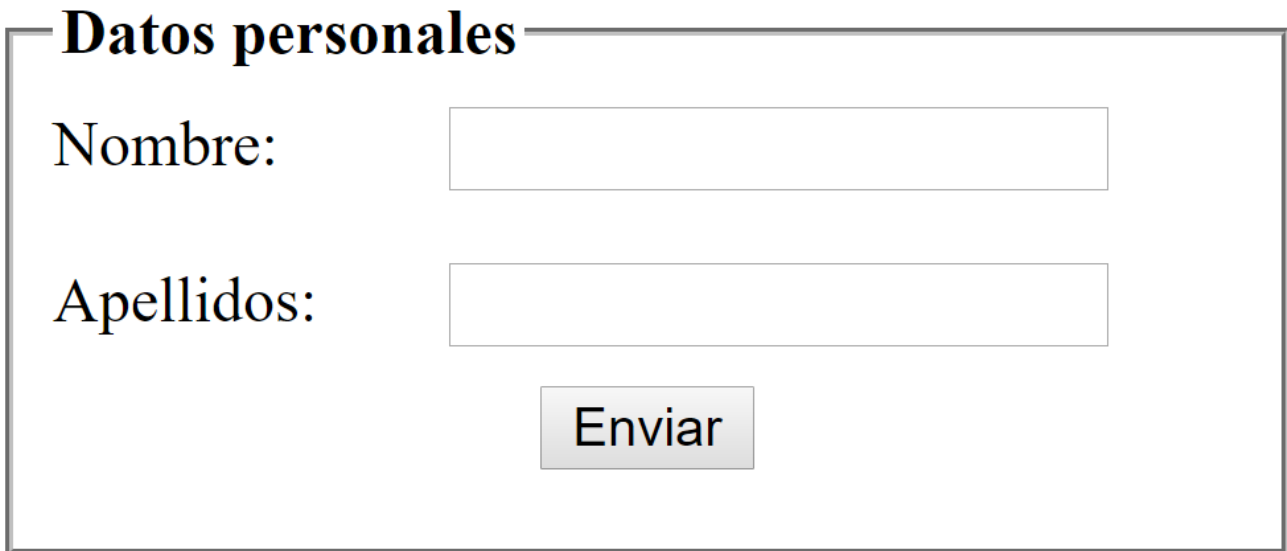
```



```
<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 5 podemos observar como se visualiza esta página en un navegador web:



Datos personales

Nombre:

Apellidos:

Figura 5: Formulario de ejemplo

Y si se añade una propiedad `text-align: right` se puede lograr otro diseño, tal como podemos ver en la Figura 6:

```
<style>
fieldset {width: 300px; padding: 10px;}
legend {font-weight: bold;}
label {float: left; display: block; width: 100px; text-align: right;}
input {float: none; display: block;}
.centrado {margin: 10px auto;}
</style>
```

Figura 6: Formulario de ejemplo con alineación de las etiquetas a la derecha

Sin embargo, un tamaño fijo no es la mejor opción. Otra opción un poco mejor es basar el tamaño no en píxeles, sino en el tamaño del texto:

```
<style>
fieldset {width: 20em; padding: 0.7em;}
legend {font-weight: bold;}
label {float: left; display: block; width: 6em;}
input {float: none; display: block;}
.centrado {margin: 10px auto;}
</style>
```

Pero todavía se puede hacer mejor, esta solución no es adaptativa. En **How to structure an HTML form** [\(47\)](#) y **Styling HTML forms** [\(48\)](#) se explican algunas técnicas básicas de maquetación de formularios. Y en **10 CSS HTML Form Designs** [\(49\)](#) se proporcionan 10 estilos visuales de formularios, con el código HTML y CSS.

4.4. Estilo adaptativo

Un buen diseño y una buena maquetación es aquella que está preparada para adaptarse a diferentes dispositivos. ¿Cómo se puede lograr? Una solución que se aplica en la actualidad es el diseño adaptativo, adaptable o flexible (*responsive design*).

El diseño adaptativo se basa principalmente en el uso de **Media Queries** [\(50\)](#), un estándar del W3C. Una *media query* es una expresión que implica la evaluación de una o varias *media features*, que da como resultado un valor booleano de verdadero o falso.

La *media query* se puede utilizar desde HTML en la etiqueta `<link />` con el atributo `media`:

```
<link href="sans-serif.css" rel="stylesheet" type="text/css" media="screen" />

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

En el ejemplo anterior, si el dispositivo de visualización es una pantalla, se usa el estilo `sans-serif.css`. Pero si el dispositivo de visualización es la impresora, se usa el estilo `serif.css`.

También se puede utilizar directamente desde CSS con la declaración `@media`:

```
* {font-family: serif;}

@media screen {
* {font-family: sans-serif;}
}
```

En el ejemplo anterior, se define un tipo de letra serif para todos los dispositivos de visualización, pero si el dispositivo de visualización es una pantalla, el tipo de letra se redefine como sans-serif para todos los elementos de la página web.

Una *media query* suele evaluar alguna de estos atributos:

- Tipo de dispositivo.
- Anchura y altura del dispositivo de visualización.
- Orientación del dispositivo (vertical o apaisado).
- Resolución del dispositivo.
- Número de colores del dispositivo.

Por ejemplo, en el siguiente fragmento de código se definen tres hojas de estilo CSS que se aplicarán según la anchura que tenga el dispositivo en el que se visualice la página web:

```
<link href="minimum.css" rel="stylesheet" type="text/css" media="screen and (max-width: 480px)" />
<link href="medium.css" rel="stylesheet" type="text/css" media="screen and (min-width: 481px) and (max-width: 1024px)" />
<link href="maximum.css" rel="stylesheet" type="text/css" media="screen and (min-width: 1025px)" />
```

También se puede emplear la instrucción `@import` [\(51\)](#), que permite importar un CSS desde otro CSS. Esta instrucción es muy útil cuando se quiera modularizar y reutilizar ciertas partes de un CSS.

El ejemplo anterior con tres etiquetas `<link>` en el código HTML, se puede convertir en una única etiqueta:

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

Y en el fichero `estilos.css` se importan las tres hojas de estilo:

```
@import url("minimum.css") screen and (max-width: 480px);
@import url("medium.css") screen and (min-width: 481px) and (max-width: 1024px);
@import url("maximum.css") screen and (min-width: 1025px);
```

Para que un diseño adaptativo funcione correctamente, se debe añadir la siguiente instrucción [\(52\)](#) en la sección `<head>` de la página web:

```
<meta name="viewport" content="width=device-width,initial-scale=1.0" />
```

Los siguientes vídeos te pueden ayudar a realizar tu diseño adaptativo:

- **CSS: creación de un diseño adaptable, adaptativo o flexible (parte 1)** [\(53\)](#): explica el diseño adaptativo, `@media`, patrones de diseño adaptativo (casi fluido, caída de columna, cambio de la estructura, pequeños cambios, fuera de la pantalla).

- **CSS: creación de un diseño adaptable, adaptativo o flexible (parte 2)**[\(54\)](#): muestra el desarrollo de un ejemplo completo con diseño adaptativo desde cero y algunas herramientas (Window Resizer, Opera Mobile Emulator).
- **CSS: creación de un diseño adaptable, adaptativo o flexible (parte 3)**[\(55\)](#): proporciona referencias y fuentes de información para aprender más cosas sobre diseño adaptativo.

4.5. Biblioteca de iconos

El empleo de iconos en una página web ayuda a mejorar la usabilidad, ya que permite una identificación visual de los elementos de la página. Por ejemplo, en la Figura 7 se puede ver un formulario que tiene un icono de una persona para representar el nombre de usuario y un icono para representar la contraseña. Y a la derecha hay un panel para identificarse mediante redes sociales, donde cada opción tiene su icono correspondiente.

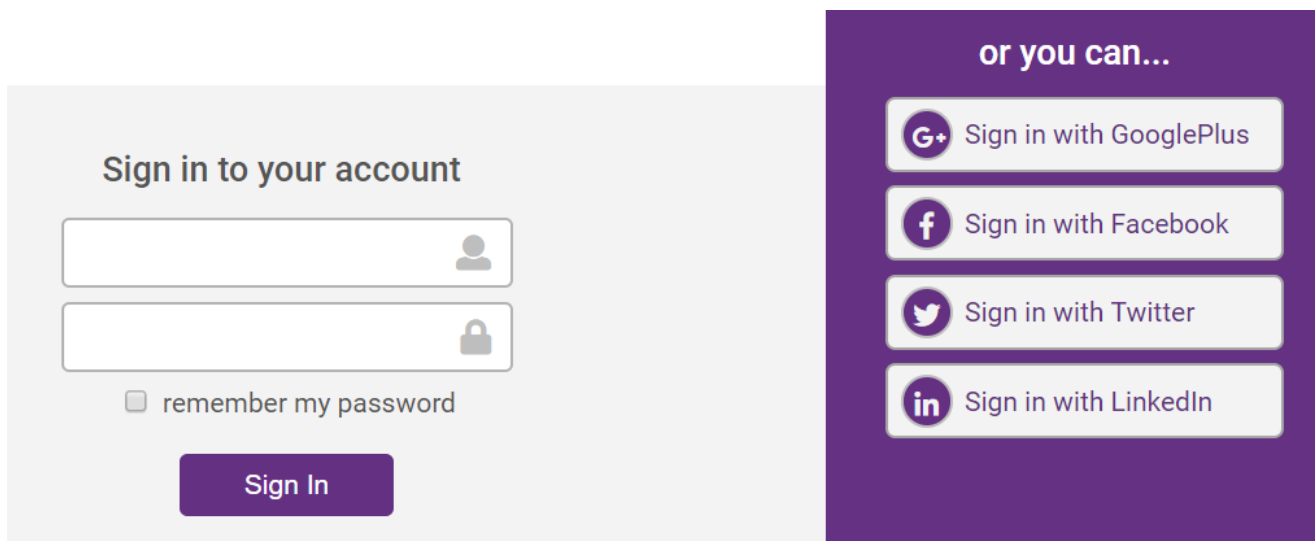
The image shows a login interface. On the left, a light gray box titled "Sign in to your account" contains two input fields: the first has a person icon on the right, and the second has a lock icon on the right. Below these is a checkbox labeled "remember my password" and a purple "Sign In" button. To the right, a purple box titled "or you can..." contains four buttons: "Sign in with GooglePlus" (with a G+ icon), "Sign in with Facebook" (with a Facebook 'f' icon), "Sign in with Twitter" (with a Twitter bird icon), and "Sign in with LinkedIn" (with a LinkedIn 'in' icon).

Figura 7: Formulario de acceso con iconos

Sin embargo, si se eliminan los iconos (por ejemplo, porque el usuario decide no cargar las imágenes), aparece un problema importante de usabilidad porque no se sabe qué escribir en los dos cuadros de texto del formulario, tal como se puede ver en la Figura 8. Además, este problema de usabilidad evidencia que también hay un problema de accesibilidad: los usuarios que utilicen un lector de pantalla (*screen reader*) no sabrán qué escribir en cada cuadro de texto, aunque aparezcan los iconos.

Este problema de usabilidad y accesibilidad no surge en el panel de la derecha porque, además de los iconos, existe la explicación textual de cada opción.

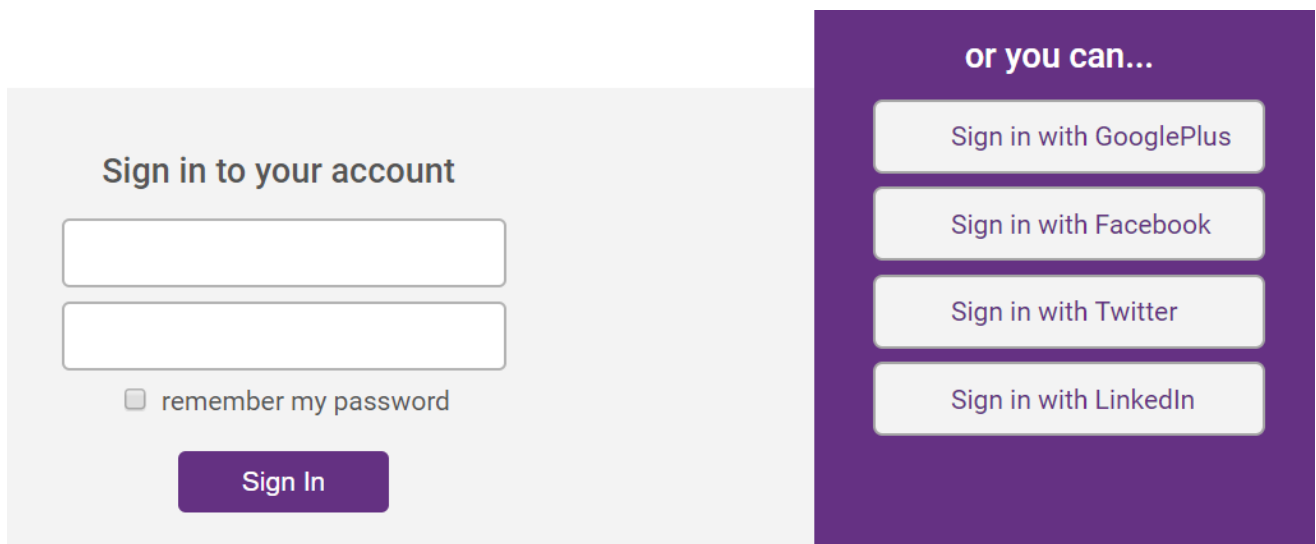


Figura 8: Formulario de acceso sin iconos

Utiliza alguna biblioteca de iconos, como Font Awesome [\(56\)](#) o Fontello [\(57\)](#), para añadir iconos a tu sitio web y así mejorar su usabilidad y su apariencia visual. Los puedes añadir a los formularios, a las barras de navegación o a otros elementos de las páginas de tu sitio web que consideres apropiado. Pero asegúrate de que todo lo que hagas no perjudica la usabilidad y la accesibilidad de las páginas web.

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. **Y no es necesario que implementes animaciones y transiciones: si lo quieres hacer para aprender más, puedes hacerlo, pero no cuenta para la evaluación de la práctica.**

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, o para destacar algo en el texto principal). Si quieres usar tipos de letra especiales, prueba con Google Fonts [\(58\)](#).

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`. Pero a veces un elemento no hereda una propiedad, aunque otros elementos sí que lo hagan. El ejemplo típico de esto es el enlace, que no hereda el color, aunque otros elementos sí que lo hacen.

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.
- 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** [\(59\)](#) es una herramienta online que formatea el aspecto de un CSS: introduce saltos de línea, espacios en blanco, etc. **CSS Optimizer** [\(60\)](#) 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.

¿Estás realizando un diseño adaptativo o adaptable? En español nos “faltan palabras”, cuesta distinguir adaptativo (*Perteneciente o relativo a la adaptación o a la capacidad de adaptación* [\(61\)](#)) y adaptable (*Capaz de ser adaptado* [\(62\)](#)), pero en inglés se distinguen dos tipos de diseño, *responsive design* y *adaptive design*. Los siguientes artículos explican las diferencias entre estos dos tipos de diseño:

- Responsive vs Adaptive Design ? Which is Best for Mobile Viewing of Your Website? [\(63\)](#)
- The Difference Between Responsive and Adaptive Design [\(64\)](#)
- RWD Is Not AWD, What Is the Difference Between Responsive and Adaptive Design? [\(65\)](#)

Y recuerda: evita la divitis [\(66\)](#), **la spanmania** y **la classitis** [\(67\)](#).

Importante:

- **Antes de añadir una etiqueta `<div>` porque quieres aplicar un estilo desde CSS, párate a pensar si puedes usar alguna de las etiquetas que ya tiene tu página.**
- **Se recomienda usar el juego de caracteres UTF-8. En las hojas de estilo también se puede indicar el juego de caracteres usado con la instrucción `@charset`** [\(68\)](#).
- **Ten cuidado con los nombres de los ficheros, utiliza únicamente letras del alfabeto inglés y números, no uses espacios en blanco y emplea únicamente minúsculas.**
- **Ten cuidado con la caché del navegador, consulta una explicación sobre los problemas y soluciones en el artículo “Ayuda: Cómo limpiar la caché”** [\(69\)](#). Recuerda esto durante toda la asignatura.

(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/>
- (8) http://es.wikipedia.org/wiki/Colores_HTML
- (9) http://es.wikipedia.org/wiki/Anexo:Colores_HTML
- (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.greenclipse.com/eclipsestyle.html>
- (17) <http://cssmate.com/csseditor.htm>
- (18) http://www.effectivetranslations.com/stupidtables/everything_es.html
- (19) http://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/>
- (22) http://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>
- (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://blogs.ua.es/pi/2012/04/17/disenio-web-responsivo-sensible-adaptable/>
- (32) <http://alistapart.com/article/responsive-web-design>
- (33) <http://www.lukew.com/ff/entry.asp?1436>
- (34) <http://bradfrost.com/blog/web/responsive-web-design-missing-the-point/>
- (35) <https://css-tricks.com/css-media-queries/>
- (36) <https://www.smashingmagazine.com/2011/07/responsive-web-design-techniques-tools-and-design-strategies/>
- (37) https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries
- (38) http://www.w3schools.com/cssref/css3_pr_mediaquery.asp
- (39) <http://www.rae.es/>
- (40) <http://www.digilife.be/quickreferences/ORC/Cascading%20Style%20Sheets%201.0.pdf>
- (41) <http://notepad-plus.sourceforge.net/es/site.htm>
- (42) <http://www.greenclipse.com/eclipsestyle.html>
- (43) <http://csscreator.com/tools/cssgenerate>
- (44) <http://www.cssportal.com/css-style-editor/>
- (45) <https://www.w3.org/TR/css-variables-1/>
- (46) <https://www.w3.org/TR/2012/WD-css-variables-20120410/>
- (47) https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms/How_to_structure_an_HTML_form
- (48) https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms/Styling_HTML_forms
- (49) <https://www.sanwebe.com/2014/08/css-html-forms-designs>

- (50) <https://www.w3.org/TR/css3-mediaqueries/>
- (51) <https://developer.mozilla.org/en-US/docs/Web/CSS/@import>
- (52) [https://developer.mozilla.org/en-US/docs/Mozilla/Mobile/Viewport meta tag](https://developer.mozilla.org/en-US/docs/Mozilla/Mobile/Viewport_meta_tag)
- (53) <https://youtu.be/ktGF7Dda0wI>
- (54) <https://youtu.be/96U4zPhyq9c>
- (55) <https://youtu.be/cbtJ4udg7sM>
- (56) <https://fontawesome.com/>
- (57) <http://fontello.com/>
- (58) <https://fonts.google.com/>
- (59) <http://www.cssportal.com/format-css/>
- (60) <http://www.cssportal.com/css-optimize/>
- (61) <http://dle.rae.es/adaptativo>
- (62) <http://dle.rae.es/adaptable>
- (63) <http://mediumwell.com/responsive-adaptive-mobile/>
- (64) <https://css-tricks.com/the-difference-between-responsive-and-adaptive-design/>
- (65) <https://www.mockplus.com/blog/post/difference-between-responsive-and-adaptive>
- (66) <https://csscreator.com/divitis>
- (67) https://www.456bereastreet.com/lab/web_development_mistakes/es/
- (68) <https://developer.mozilla.org/en-US/docs/Web/CSS/@charset>
- (69) <http://www.digilife.be/quickreferences/ORC/Cascading%20Style%20Sheets%201.0.pdf>