

CSS.

1. INTRODUCCIÓN AL CSS	3
1.1 ¿QUÉ ES CSS?	3
1.2 FUNCIONAMIENTO BÁSICO.....	3
1.3 CÓMO INCLUIR CSS EN UN DOCUMENTO HTML.....	4
1.3.1 Incluir CSS en el mismo documento HTML.....	4
1.3.2 Definir CSS en un archivo externo.....	5
1.3.3 Incluir CSS en los elementos HTML.....	6
2. SELECTORES	6
2.1 SELECTORES BÁSICOS	7
2.1.1 Selector de tipo o etiqueta	7
2.1.2 Selector de clase	8
2.1.3 Selectores de ID	10
2.1.4 Agrupación de reglas.....	11
2.2 HERENCIA	11
3. UNIDADES DE MEDIDA Y COLORES	12
3.1 UNIDADES DE MEDIDA	12
3.1.1 Unidades relativas	12
3.1.2 Unidades absolutas	13
3.2 COLORES	13
3.2.1 Palabras clave	14
3.2.2 RGB decimal	14
3.2.3 RGB porcentual	14
3.2.4 RGB hexadecimal.....	14
4. MODELO DE CAJAS (BOX MODEL)	14
4.1 ANCHURA Y ALTURA	15
4.1.1 Anchura	15
4.1.2 Altura	15
4.2 MARGEN Y RELLENO	16
4.2.1 Margen	16
4.2.2 Relleno.....	17
4.3 BORDES	
4.3.1 Anchura	19
4.3.2 Color	19
4.3.3 Estilo	19
4.4 FONDOS	19
5. TEXTO	21
5.1 TIPOGRAFÍA	21
5.2 TEXTO	23
6. ENLACES	25
6.1 ESTILOS BÁSICOS	25
6.1.1 Tamaño, color y decoración.....	25

6.1.2	Pseudo-clases	25
6.2	ESTILOS AVANZADOS	26
6.2.1	Decoración personalizada.....	26
6.2.2	Imágenes según el tipo de enlace	26
6.2.3	Mostrar los enlaces como si fueran botones	27
7.	IMÁGENES	27
7.1	ESTILOS BÁSICOS	27
7.1.1	Establecer la anchura y altura de las imágenes	27
7.1.2	Eliminar el borde de las imágenes con enlaces	28
8.	LISTAS	28
8.1	ESTILOS BÁSICOS	28
8.1.1	Viñetas personalizadas	28
9.	EJERCICIOS	30



1. Introducción al CSS

1.1 ¿Qué es CSS?

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

1.2 Funcionamiento básico

Antes de la adopción de CSS, los diseñadores de páginas web debían definir el aspecto de cada elemento dentro de las etiquetas HTML de la página. El siguiente ejemplo muestra una página HTML con estilos definidos sin utilizar CSS:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Ejemplo de estilos sin CSS</title>
</head>
<body>
<h1><font color="red" face="Arial" size="5">Titular de la página</font></h1>
<p><font color="gray" face="Verdana" size="2">Un párrafo de texto no muy largo.</font></p>
</body>
</html>
```

El ejemplo anterior utiliza la etiqueta con sus atributos color, face y size para definir el color, la tipografía y el tamaño del texto de cada elemento del documento.

El principal problema de esta forma de definir el aspecto de los elementos se puede ver claramente con el siguiente ejemplo: si la página tuviera 50 elementos diferentes, habría que insertar 50 etiquetas . Si el sitio web entero se compone de 10.000 páginas diferentes, habría que definir 500.000 etiquetas . Como cada etiqueta tiene 3 atributos, habría que definir 1.5 millones de atributos.

Por otra parte, el diseño de los sitios web está en constante evolución y es habitual modificar cada cierto tiempo los colores principales de las páginas o la tipografía utilizada para el texto. Si



se emplea la etiqueta ``, habría que modificar el valor de 1.5 millones de atributos para modificar el diseño general del sitio web.

La solución que propone CSS es mucho mejor, como se puede ver en el siguiente ejemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html >
<head>
<title>Ejemplo de estilos con CSS</title>
<style type="text/css">
h1 { color: red; font-family: Arial; font-size: large; }
p { color: gray; font-family: Verdana; font-size: medium; }
</style>
</head>
<body>
<h1>Titular de la página</h1>
<p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```

CSS permite separar los contenidos de la página y su aspecto o presentación. En el ejemplo anterior, dentro de la propia página HTML se reserva una zona en la que se incluye toda la información relacionada con los estilos de la página.

Utilizando CSS, en esa zona reservada se indica que todas las etiquetas `<h1>` de la página se deben ver de color rojo, con un tipo de letra Arial y con un tamaño de letra grande. Además, las etiquetas `<p>` de la página se deben ver de color gris, con un tipo de letra Verdana y con un tamaño de letra medio.

Definiendo los estilos de esta forma, no importa el número de elementos `<p>` que existan en la página, ya que todos tendrán el mismo aspecto establecido mediante CSS. Como se verá a continuación, esta forma de trabajar con CSS no es ideal, ya que si el sitio web dispone de 10.000 páginas, habría que definir 10.000 veces el mismo estilo CSS.

1.3 Cómo incluir CSS en un documento HTML

Una de las principales características de CSS es su flexibilidad y las diferentes opciones que ofrece para realizar una misma tarea. De hecho, existen tres opciones para incluir CSS en un documento HTML.

1.3.1 Incluir CSS en el mismo documento HTML

Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta `<style>` de HTML y solamente se pueden incluir en la cabecera del documento (sólo dentro de la sección `<head>`).

Ejemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```



```

<html>
<head>
<title>Ejemplo de estilos CSS en el propio documento</title>
<style type="text/css">
p { color: black; font-family: Verdana; }
</style>
</head>
<body>
<p>Un párrafo de texto.</p>
</body>
</html>

```

Este método se emplea cuando se define un número pequeño de estilos o cuando se quieren incluir estilos específicos en una determinada página HTML que completen los estilos que se incluyen por defecto en todas las páginas del sitio web.

El principal inconveniente es que si se quiere hacer una modificación en los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a modificar.

1.3.2 Definir CSS en un archivo externo

En este caso, todos los estilos CSS se incluyen en un archivo de tipo CSS que las páginas HTML enlazan mediante la etiqueta <link>. Un archivo de tipo CSS no es más que un archivo simple de texto cuya extensión es .css. Se pueden crear todos los archivos CSS que sean necesarios y cada página HTML puede enlazar tantos archivos CSS como necesite.

En el siguiente ejemplo, se crea un archivo de texto, se cambia su nombre a estilos.css y se incluye el siguiente contenido:

```
p { color: red; font-family: Verdana; }
```

A continuación, en la página HTML se utiliza la etiqueta <link> para enlazar el archivo CSS externo que tiene los estilos que va a utilizar la página:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Ejemplo de estilos CSS en un archivo externo</title>
<link rel="stylesheet" type="text/css" href="estilos.css" media="screen" />
</head>
<body>
<p>Un párrafo de texto.</p>
</body>
</html>

```

Normalmente, la etiqueta <link> incluye cuatro atributos cuando se enlaza un archivo CSS:

- rel: indica el tipo de relación que tiene el recurso enlazado (en este caso, el archivo CSS) y la página HTML. Para los archivos CSS, siempre se utiliza el valor stylesheet.



- type: indica el tipo de recurso enlazado. Sus valores están estandarizados y para los archivos CSS su valor siempre es text/css.
- href: indica la URL del archivo CSS que contiene los estilos. La URL indicada puede ser relativa o absoluta y puede apuntar a un recurso interno o externo al sitio web.
- media: indica el medio en el que se van a aplicar los estilos del archivo CSS.

De todas las formas de incluir CSS en las páginas HTML, esta es la más utilizada con mucha diferencia. La principal ventaja es que se puede incluir un mismo archivo CSS en multitud de páginas HTML, por lo que se garantiza la aplicación homogénea de los mismos estilos a todas las páginas que forman un sitio web.

Con este método, el mantenimiento del sitio web se simplifica al máximo, ya que un solo cambio en un solo archivo CSS permite variar de forma instantánea los estilos de todas las páginas HTML que enlazan ese archivo.

1.3.3 Incluir CSS en los elementos HTML

El último método para incluir estilos CSS en documentos HTML es el peor y el menos utilizado, ya que tiene los mismos problemas que la utilización de las etiquetas .

Ejemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html >
<head>
<title>Ejemplo de estilos CSS en el propio documento</title>
</head>
<body>
<p style="color: black; font-family: Verdana;">Un párrafo de texto.</p>
</body>
</html>
```

Esta forma de incluir CSS directamente en los elementos HTML solamente se utiliza en determinadas situaciones en las que se debe incluir un estilo muy específico para un solo elemento concreto.

2. Selectores

Para crear diseños web profesionales, es imprescindible conocer y dominar los selectores de CSS. Como se vio en el capítulo anterior, una regla de CSS está formada por una parte llamada "selector" y otra parte llamada "declaración".

La declaración indica "qué hay que hacer" y el selector indica "a quién hay que hacérselo". Por lo tanto, los selectores son imprescindibles para aplicar de forma correcta los estilos CSS en una página.

A un mismo elemento HTML se le pueden asignar infinitas reglas CSS y cada regla CSS puede aplicarse a un número infinito de elementos. En otras palabras, una misma regla puede aplicarse sobre varios selectores y un mismo selector se puede utilizar en varias reglas.

El estándar de CSS 2.1 incluye una docena de tipos diferentes de selectores, que permiten seleccionar de forma muy precisa elementos individuales o conjuntos de elementos dentro de una página web.

No obstante, la mayoría de páginas de los sitios web se pueden diseñar utilizando solamente los cinco selectores básicos.



2.1 Selectores básicos

2.1.1 Selector de tipo o etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector. El siguiente ejemplo selecciona todos los párrafos de la página:

```
p {
...
}
```

Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML (sin los caracteres < y >) correspondiente a los elementos que se quieren seleccionar.

El siguiente ejemplo aplica diferentes estilos a los titulares y a los párrafos de una página HTML:

```
h1 {
color: red;
}
h2 {
color: blue;
}
p {
color: black;
}
```

Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, se pueden encadenar los selectores. En el siguiente ejemplo, los títulos de sección h1, h2 y h3 comparten los mismos estilos:

```
h1 {
color: #8A8E27;
font-weight: normal;
font-family: Arial, Helvetica, sans-serif;
}
h2 {
color: #8A8E27;
font-weight: normal;
font-family: Arial, Helvetica, sans-serif;
}
h3 {
color: #8A8E27;
font-weight: normal;
font-family: Arial, Helvetica, sans-serif;
}
```

En este caso, CSS permite agrupar todas las reglas individuales en una sola regla con un selector múltiple. Para ello, se incluyen todos los selectores separados por una coma (,) y el resultado es que la siguiente regla CSS es equivalente a las tres reglas anteriores:

```
h1, h2, h3 {
```



```
color: #8A8E27;
font-weight: normal;
font-family: Arial, Helvetica, sans-serif;
}
```

En las hojas de estilo complejas, es habitual agrupar las propiedades comunes de varios elementos en una única regla CSS y posteriormente definir las propiedades específicas de esos mismos elementos. El siguiente ejemplo establece en primer lugar las propiedades comunes de los títulos de sección (color y tipo de letra) y a continuación, establece el tamaño de letra de cada uno de ellos:

```
h1, h2, h3 {
color: #8A8E27;
font-weight: normal;
font-family: Arial, Helvetica, sans-serif;
}
h1 { font-size: 2em; }
h2 { font-size: 1.5em; }
h3 { font-size: 1.2em; }
```

2.1.2 Selector de clase

Si se considera el siguiente código HTML de ejemplo:

```
<body>
<p>Lorem ipsum dolor sit amet...</p>
<p>Nunc sed lacus et est adipiscing accumsan...</p>
<p>Class aptent taciti sociosqu ad litora...</p>
</body>
```

¿Cómo se pueden aplicar estilos CSS sólo al primer párrafo? El selector de tipo o etiqueta (p) tampoco se puede utilizar porque seleccionaría todos los párrafos.

Una de las soluciones más sencillas consiste en indicar directamente en el elemento la regla CSS que se le va a aplicar. El atributo class de HTML permite indicar el nombre de la regla CSS que se aplica sobre el elemento:

```
<body>
<p class="destacado">Lorem ipsum dolor sit amet...</p>
<p>Nunc sed lacus et est adipiscing accumsan...</p>
<p>Class aptent taciti sociosqu ad litora...</p>
</body>
```

A continuación, se crea en el archivo CSS una nueva regla llamada destacado con todos los estilos que se van a aplicar al elemento. Para que el navegador no confunda este selector con los otros tipos de selectores, se prefija el valor del atributo class con un punto (.) tal y como muestra el siguiente ejemplo:



```
.destacado { color: red; }
```

El selector `.destacado` se interpreta como "cualquier elemento de la página cuyo atributo `class` sea igual a `destacado`", por lo que solamente el primer párrafo cumple esa condición. Este tipo de selectores se llaman selectores de clase y son los más utilizados junto con los selectores de ID que se verán a continuación. La principal característica de este selector es que en una misma página HTML varios elementos diferentes pueden incluir el mismo valor en el atributo `class`:

```
<body>
<p class="destacado">Lorem ipsum dolor sit amet...</p>
<p>Nunc sed lacus et <a href="#" class="destacado">est adipiscing</a> accumsan...</p>
<p>Class aptent taciti <em class="destacado">sociosqu ad</em> litora...</p>
</body>
```

Los selectores de clase son imprescindibles para diseñar páginas web complejas, ya que permiten disponer de una precisión total al seleccionar los elementos. Además, estos selectores permiten reutilizar los mismos estilos para varios elementos diferentes. A continuación se muestra otro ejemplo de selectores de clase:

```
.aviso {
padding: 0.5em;
border: 1px solid #98be10;
background: #f6feda;
}
.error {
color: #930;
font-weight: bold;
}
<span class="error">...</span>
<div class="aviso">...</div>
```

El elemento `` tiene un atributo `class="error"`, por lo que se le aplican las reglas CSS indicadas por el selector `.error`. Por su parte, el elemento `<div>` tiene un atributo `class="aviso"`, por lo que su estilo es el que definen las reglas CSS del selector `.aviso`. En ocasiones, es necesario restringir el alcance del selector de clase. Si se considera de nuevo el ejemplo anterior:

```
<body>
<p class="destacado">Lorem ipsum dolor sit amet...</p>
<p>Nunc sed lacus et <a href="#" class="destacado">est adipiscing</a> accumsan...</p>
<p>Class aptent taciti <em class="destacado">sociosqu ad</em> litora...</p>
</body>
```

¿Cómo es posible aplicar estilos solamente al párrafo cuyo atributo `class` sea igual a `destacado`?



Combinando el selector de tipo y el selector de clase, se obtiene un selector mucho más específico:

```
p.destacado { color: red }
```

El selector p.destacado se interpreta como "aquellos elementos de tipo <p> que dispongan de un atributo class con valor destacado". De la misma forma, el selector a.destacado solamente selecciona los enlaces cuyo atributo class sea igual a destacado.

De lo anterior se deduce que el atributo .destacado es equivalente a *.destacado, por lo que todos los diseñadores obvian el símbolo * al escribir un selector de clase normal.

Por último, es posible aplicar los estilos de varias clases CSS sobre un mismo elemento. La sintaxis es similar, pero los diferentes valores del atributo class se separan con espacios en blanco. En el siguiente ejemplo:

```
<p class="especial destacado error">Párrafo de texto...</p>
```

Al párrafo anterior se le aplican los estilos definidos en las reglas .especial, .destacado y .error, por lo que en el siguiente ejemplo, el texto del párrafo se vería de color rojo, en negrita y con un tamaño de letra de 15 píxel:

```
.error { color: red; }
.destacado { font-size: 15px; }
.especial { font-weight: bold; }
<p class="especial destacado error">Párrafo de texto...</p>
```

Si un elemento dispone de un atributo class con más de un valor, es posible utilizar un selector más avanzado:

```
.error { color: red; }
.error.destacado { color: blue; }
.destacado { font-size: 15px; }
.especial { font-weight: bold; }
<p class="especial destacado error">Párrafo de texto...</p>
```

En el ejemplo anterior, el color de la letra del texto es azul y no rojo. El motivo es que se ha utilizado un selector de clase múltiple .error.destacado, que se interpreta como "aquellos elementos de la página que dispongan de un atributo class con al menos los valores error y destacado".

2.1.3 Selectores de ID

En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.

El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo id. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo id no se puede repetir en dos elementos diferentes de una misma página.

La sintaxis que utilizan los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

```
#destacado { color: red; }
<p>Primer párrafo</p>
```



```
<p id="destacado">Segundo párrafo</p>
```

```
<p>Tercer párrafo</p>
```

En el ejemplo anterior, el selector #destacado solamente selecciona el segundo párrafo (cuyo atributo id es igual a destacado).

La principal diferencia entre este tipo de selector y el selector de clase tiene que ver con HTML y no con CSS. Como se sabe, en una misma página, el valor del atributo id debe ser único, de forma que dos elementos diferentes no pueden tener el mismo valor de id. Sin embargo, el atributo class no es obligatorio que sea único, de forma que muchos elementos HTML diferentes pueden compartir el mismo valor para su atributo class.

De esta forma, la recomendación general es la de utilizar el selector de ID cuando se quiere aplicar un estilo a un solo elemento específico de la página y utilizar el selector de clase cuando se quiere aplicar un estilo a varios elementos diferentes de la página HTML.

2.1.4 Agrupación de reglas

Cuando se crean archivos CSS complejos con decenas o cientos de reglas, es habitual que los estilos que se aplican a un mismo selector se definan en diferentes reglas:

```
h1 { color: red; }
```

```
...
```

```
h1 { font-size: 2em; }
```

```
...
```

```
h1 { font-family: Verdana; }
```

Las tres reglas anteriores establecen el valor de tres propiedades diferentes de los elementos <h1>. Antes de que el navegador muestre la página, procesa todas las reglas CSS de la página para tener en cuenta todos los estilos definidos para cada elemento.

Cuando el selector de dos o más reglas CSS es idéntico, se pueden agrupar las declaraciones de las reglas para hacer las hojas de estilos más eficientes:

```
h1 {
color: red;
font-size: 2em;
font-family: Verdana;
}
```

El ejemplo anterior tiene el mismo efecto que las tres reglas anteriores, pero es más eficiente y es más fácil de modificar y mantener por parte de los diseñadores. Como CSS ignora los espacios en blanco y las nuevas líneas, también se pueden agrupar las reglas de la siguiente forma:

```
h1 { color: red; font-size: 2em; font-family: Verdana; }
```

2.2 Herencia

Uno de los conceptos más característicos de CSS es la herencia de los estilos definidos para los elementos. Cuando se establece el valor de alguna propiedad en un elemento, todos sus descendientes heredan inicialmente ese mismo valor.

Si se indica por ejemplo un tipo de letra al elemento <body> de una página, todos los elementos de la página mostrarán ese tipo de letra, salvo que se indique lo contrario:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Ejemplo de herencia de estilos</title>
```



```
<style type="text/css">
body { font-family: Arial; color: black; }
h1 { font-family: Verdana; }
p { color: red; }
</style>
</head>
<body>
<h1>Titular de la página</h1>
<p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```

En el ejemplo anterior, se ha indicado que la etiqueta `<body>` tiene asignado un tipo de letra Arial y un color de letra negro. Así, todos los elementos de la página (salvo que se indique lo contrario) se muestran de color negro y con la fuente Arial.

La segunda regla indica que los elementos `<h1>` se muestran con otra tipografía diferente a la heredada. La tercera regla indica que los elementos `<p>` varían su color respecto del color que han heredado.

La herencia de estilos no funciona en todas las propiedades CSS, por lo que se debe estudiar cada propiedad de forma individual.

3. Unidades de medida y colores

Muchas de las propiedades de CSS permiten indicar unidades de medida y colores en sus valores. Además, CSS es tan flexible que permite indicar las medidas y colores de muchas formas diferentes. Por este motivo, se presentan a continuación todas las alternativas disponibles en CSS para indicar las medidas y los colores.

3.1 Unidades de medida

Las medidas en CSS se emplean, entre otras, para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto. Todas las medidas se indican como un valor numérico entero o decimal seguido de una unidad de medida (sin ningún espacio en blanco entre el número y la unidad de medida).

CSS divide todas las unidades de medida en dos grupos: absolutas y relativas. Las medidas relativas definen su valor en relación con otra medida, por lo que para obtener su valor real, se debe realizar alguna operación con el valor indicado. Las unidades absolutas establecen de forma completa el valor de una medida, por lo que su valor real es directamente el valor indicado.

Si el valor es 0, la unidad de medida es opcional. Si el valor es distinto a 0 y no se indica ninguna unidad, la medida se ignora completamente, lo que suele ser una fuente habitual de errores para los diseñadores que empiezan con CSS. Algunas propiedades permiten indicar medidas negativas, aunque habitualmente sus valores son positivos.

3.1.1 Unidades relativas

Las unidades relativas son más flexibles que las unidades absolutas porque se adaptan más fácilmente a los diferentes medios. A continuación se muestra la lista de unidades de medida relativas y la referencia que se toma para determinar su valor real:



- em, relativa respecto del tamaño de letra empleado. Aunque no es una definición exacta, el valor de 1 em se puede aproximar por la anchura de la letra M ("eme mayúscula") del tipo de letra que se esté utilizando
- ex, relativa respecto de la altura de la letra x ("equis minúscula") del tipo de letra que se esté utilizando
- px, (píxel) relativa respecto de la pantalla del usuario

Las unidades em y ex no han sido definidas por CSS, sino que llevan décadas utilizándose en el campo de la tipografía.

En el siguiente ejemplo, se indica que el tamaño de letra del texto de la página debe ser el 90% del tamaño por defecto (que depende de cada navegador, aunque es muy similar entre ellos):

```
body { font-size: 0.9em; }
```

Como em es una unidad relativa, el valor 0.9 indicado sólo tiene sentido cuando se tiene en consideración su referencia. Para la unidad em, la referencia es el tamaño de letra por defecto del sistema (ordenador, dispositivo móvil, etc.) del usuario.

Por lo tanto, 0.9em significa que se debe multiplicar 0.9 por el tamaño de letra por defecto, lo que en la práctica significa que la medida indicada es igual al 90% del tamaño de letra por defecto.

Cuando el valor decimal de una medida es inferior a 1, se puede omitir el 0 de la izquierda, por lo que el código anterior es equivalente al código siguiente:

```
body { font-size: .9em; }
```

3.1.2 Unidades absolutas

Las unidades absolutas definen las medidas de forma completa, ya que sus valores reales no se calculan a partir de otro valor de referencia, sino que son directamente los valores indicados. A continuación se muestra la lista completa de unidades absolutas definidas por CSS y su significado:

- in, del inglés "inches", pulgadas (1 pulgada son 2.54 centímetros)
- cm, centímetros
- mm, milímetros
- pt, puntos (1 punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros)
- pc, picas (1 pica equivale a 12 puntos, es decir, unos 4.23 milímetros)

A continuación se muestran ejemplos de utilización de unidades absolutas:

```
body { margin: 0.5in; }
h1 { line-height: 2cm; }
p { word-spacing: 4mm; }
a { font-size: 12pt }
span { font-size: 1pc }
```

Su uso es idéntico al de las unidades relativas, siendo su única diferencia que los valores indicados son directamente los valores que se utilizan, sin necesidad de calcular los valores reales en función de otras referencias.

De todas las unidades absolutas, la única que se utiliza con cierta frecuencia es la de los puntos (pt). El motivo es que se trata de la unidad preferida para indicar el tamaño de letra del texto para los documentos que se van a imprimir.

3.2 Colores

Los colores en CSS se pueden indicar de cinco formas diferentes: palabras clave, colores del sistema, RGB hexadecimal, RGB numérico y RGB porcentual. Aunque el método más habitual es el del RGB hexadecimal, a continuación se muestran todas las alternativas que ofrece CSS.



3.2.1 Palabras clave

CSS define 17 palabras clave para referirse a los colores básicos. Las palabras se corresponden con el nombre en inglés de cada color:

aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow

3.2.2 RGB decimal

En el campo del diseño gráfico, se han definido varios modelos diferentes para referirse a los colores. Los dos modelos más conocidos son RGB y CMYK. Aunque no es una definición exacta, el modelo RGB consiste en definir un color indicando que cantidad de color rojo, verde y azul se debe mezclar para obtener el color. Técnicamente, el modelo RGB es un modelo de tipo "aditivo", ya que se suman colores para obtener el color deseado.

Por lo tanto, en el modelo RGB un color se define indicando sus tres componentes R (rojo), G (verde) y B (azul). Cada una de las componentes puede tomar un valor entre cero y un valor máximo. De esta forma, el color rojo puro en RGB se crea mediante el máximo valor de la componente R y un valor de 0 para las componentes G y B.

Si todas las componentes valen 0, el color creado es el negro; si todas las componentes toman su valor máximo, el color obtenido es el blanco. En CSS, las componentes de los colores definidos mediante RGB decimal pueden tomar valores entre 0 y 255. El siguiente ejemplo establece el color del texto de un párrafo:

```
p { color: rgb(71, 98, 176); }
```

La sintaxis que se utiliza para indicar los colores es rgb() y entre paréntesis se indican las tres componentes RGB, en ese mismo orden y separadas por comas.

Si se indica un valor menor que 0 para una componente, automáticamente se transforma su valor en 0. Igualmente, si se indica un valor mayor que 255, se transforma automáticamente su valor a 255.

3.2.3 RGB porcentual

Otra forma de indicar las componentes RGB de un color es mediante un porcentaje. El funcionamiento y la sintaxis de este método es el mismo que el del RGB decimal. La única diferencia en este caso es que el valor de las componentes RGB puede tomar valores entre 0% y 100%. El mismo color del ejemplo anterior se puede representar de forma porcentual:

```
p { color: rgb(27%, 38%, 69%); }
```

Al igual que sucede con el RGB decimal, si se indica un valor inferior a 0%, se transforma automáticamente en 0% y si se indica un valor superior a 100%, se trunca su valor a 100%.

3.2.4 RGB hexadecimal

Aunque es el método más complicado de indicar los colores, se trata del método más utilizado con mucha diferencia. De hecho, prácticamente todos los sitios web reales utilizan exclusivamente este método.

Con esta nueva notación, el color del mismo ejemplo anterior se indica de la siguiente forma:

```
p { color: #4762B0; }
```

4. Modelo de cajas (box model)

El modelo de cajas o "box model" es seguramente la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web. El "box model" es el comportamiento de CSS que provoca que todos los elementos incluidos en una página HTML se representen mediante cajas rectangulares. Además, CSS permite controlar las propiedades de las



cajas y también su representación visual.

El diseño de cualquier página web está compuesto por cajas rectangulares. CSS permite definir la altura y anchura de cada caja, el margen que se deja entre cada caja y el espacio de relleno interior que muestra cada caja.

Además, CSS permite controlar la forma en la que se visualizan las cajas: se pueden ocultar, desplazar respecto de su posición original, fijarlas en una posición concreta dentro del documento, etc.

Las partes que componen cada caja y su orden de visualización desde el punto de vista del usuario son las siguientes:

- Contenido (content): se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
- Relleno (padding): está formado por el espacio libre entre el contenido y el borde que lo encierra.
- Borde (border): línea que encierra completamente el contenido y su relleno.
- Imagen de fondo (background image): imagen que se muestra por detrás del contenido.
- Color de fondo (background color): color que se muestra en el espacio ocupado por el contenido y su posible relleno.
- Margen (margin): espacio libre entre la caja y las posibles cajas adyacentes.

El relleno y el margen son transparentes, por lo que en el espacio ocupado por el relleno se muestra el color o imagen de fondo (si están definidos) y en el espacio ocupado por el margen se muestra el color o imagen de fondo de su elemento padre (si están definidos). Si ningún elemento padre tiene definido un color o imagen de fondo, se muestra el color o imagen de fondo de la propia página (si están definidos).

Si una caja define tanto un color como una imagen de fondo, la imagen tiene más prioridad y es la que se visualiza.

4.1 Anchura y altura

4.1.1 Anchura

La propiedad CSS que controla la anchura de los elementos se denomina width.

La propiedad width no admite valores negativos y los valores en porcentaje se calculan a partir de la anchura de su elemento padre. El valor inherit indica que la anchura del elemento se hereda de su elemento padre. El valor auto, que es el que se utiliza si no se establece de forma explícita un valor a esta propiedad, indica que el navegador debe calcular automáticamente la anchura del elemento, teniendo en cuenta sus contenidos y el sitio disponible en la página.

El siguiente ejemplo establece el valor de la anchura del elemento <div> lateral:

```
#lateral { width: 200px; }  
<div id="lateral">  
...  
</div>
```

CSS define otras dos propiedades relacionadas con la anchura de los elementos: min-width y max-width, que se verán más adelante.

4.1.2 Altura

La propiedad CSS que controla la altura de los elementos se denomina height.

Al igual que sucede con width, la propiedad height no admite valores negativos. Si se indica un porcentaje, se toma como referencia la altura del elemento padre. Si el elemento padre no tiene una altura definida explícitamente, se asigna el valor auto a la altura.

El valor inherit indica que la altura del elemento se hereda de su elemento padre. El valor auto, que es el que se utiliza si no se establece de forma explícita un valor a esta propiedad, indica que



el navegador debe calcular automáticamente la altura del elemento, teniendo en cuenta sus contenidos y el sitio disponible en la página.

El siguiente ejemplo establece el valor de la altura del elemento <div> de cabecera:

```
#cabecera { height: 60px; }
<div id="cabecera">
...
</div>
```

CSS define otras dos propiedades relacionadas con la altura de los elementos: min-height y max-height, que se verán más adelante.

4.2 Margen y relleno

4.2.1 Margen

CSS define cuatro propiedades para controlar cada uno de los márgenes horizontales y verticales de un elemento.

Aunque no es habitual, el valor de los márgenes puede ser negativo. El siguiente ejemplo añade un margen izquierdo al segundo párrafo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo propiedad margin-left</title>
<style type="text/css">
.destacado {
margin-left: 2em;
}
</style>
</head>
<body>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam et elit.
Vivamus placerat lorem. Maecenas sapien. Integer ut massa. Cras diam ipsum,
laoreet non, tincidunt a, viverra sed, tortor.</p>
<p class="destacado">Vestibulum lectus diam, luctus vel, venenatis ultrices,
cursus vel, tellus. Etiam placerat erat non sem. Nulla molestie odio non
nisi tincidunt faucibus.</p>
<p>Aliquam euismod sapien eu libero. Ut tempor orci at nulla. Nam in eros
egestas massa vehicula nonummy. Morbi posuere, nibh ultricies consectetur tincidunt,
risus turpis laoreet elit, ut tincidunt risus sem et nunc.</p> </body>
```




</html>

El siguiente ejemplo añade un margen a todos los iconos para facilitar su identificación y mejorar el diseño general de la página:

El código CSS del ejemplo anterior se muestra a continuación:

```
div img {
margin-top: .5em;
margin-bottom: .5em;
margin-left: 1em;
margin-right: .5em;
}
```

Además de las cuatro propiedades que controlan cada uno de los márgenes del elemento, CSS define una propiedad que permite establecer los cuatro márgenes de forma directa empleando una única propiedad. Este tipo de propiedades resumidas se denominan propiedades de tipo "shorthand" y CSS define varias propiedades de este tipo, como se verá más adelante.

La propiedad que permite definir de forma simultánea los cuatro márgenes se denomina margin.

El ejemplo anterior de márgenes se puede reescribir utilizando la propiedad margin:

Código CSS original:

```
div img {
margin-top: .5em;
margin-bottom: .5em;
margin-left: 1em;
margin-right: .5em;
}
```

Alternativa directa:

```
div img {
margin: .5em .5em .5em 1em;
}
```

Otra alternativa:

```
div img {
margin: .5em;
margin-left: 1em;
}
```

4.2.2 Relleno

CSS define cuatro propiedades para controlar cada uno de los espacios de relleno horizontales y verticales de un elemento.



El código HTML y CSS del ejemplo se muestra a continuación:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo diferencia entre margin y padding</title>
<style type="text/css">
.margen {
margin-top: 2em; margin-right: 2em; margin-bottom: 2em; margin-left: 2em;
}
.relleno {
padding-top: 2em; padding-right: 2em; padding-bottom: 2em; padding-left: 2em;
}
</style>
</head>
<body>
<p class="margen">Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Cras vitae dolor eu enim dignissim lacinia. Maecenas blandit. Morbi mi.</p>
<p class="relleno">Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Cras vitae dolor eu enim dignissim lacinia. Maecenas blandit. Morbi mi.</p>
</body>
</html>
```

Como sucede con la propiedad margin, CSS también define una propiedad de tipo "shorthand" para establecer los cuatro rellenos de un elemento de forma directa. La propiedad que permite definir de forma simultánea los cuatro márgenes se denomina padding.

La notación {1, 4} de la definición anterior significa que la propiedad padding admite entre uno y cuatro valores, con el mismo significado que el de la propiedad margin. Ejemplo:

```
body {padding: 2em} /* Todos los rellenos valen 2em */
body {padding: 1em 2em} /* Superior e inferior = 1em, Izquierdo y derecho = 2em */
body {padding: 1em 2em 3em} /* Superior = 1em, derecho = 2em, inferior = 3em, izquierdo
= 2em */
body {padding: 1em 2em 3em 4em} /* Superior = 1em, derecho = 2em, inferior = 3em,
izquierdo = 4em */
```

4.3 Bordes

CSS permite definir el aspecto de cada uno de los cuatro bordes horizontales y verticales de los elementos. Para cada borde se puede establecer su anchura, su color y su estilo.



4.3.1 Anchura

La anchura de los bordes se controla con las cuatro propiedades siguientes:

La anchura de los bordes se puede indicar mediante una medida (absoluta o relativa y en cualquier unidad de medida de las definidas) o mediante las palabras clave thin (borde delgado), medium (borde normal) y thick (borde ancho).

Normalmente se utilizan medidas expresadas en píxel o en em, ya que las palabras clave definidas no son muy comunes.

Las reglas CSS utilizadas se muestran a continuación:

```
div {  
border-top-width: 10px;  
border-right-width: 1em;  
border-bottom-width: thick;  
border-left-width: thin;  
}
```

4.3.2 Color

Las reglas CSS necesarias para mostrar los colores anteriores son las siguientes:

```
div {  
border-top-color: #CC0000;  
border-right-color: blue;  
border-bottom-color: #00FF00;  
border-left-color: #CCC;  
}
```

4.3.3 Estilo

Las reglas CSS necesarias para mostrar los estilos anteriores son las siguientes:

```
div {  
border-top-style: dashed;  
border-right-style: double;  
border-bottom-style: dotted;  
border-left-style: solid;  
}
```

4.4 Fondos

El último elemento que forma el box model es el fondo de la caja del elemento. El fondo puede ser un color simple o una imagen. El fondo solamente se visualiza en el área ocupada por el contenido y su relleno, ya que el color de los bordes se controla directamente desde los bordes y las zonas de los márgenes siempre son transparentes.

Para establecer un color o imagen de fondo en la página entera, se debe establecer un fondo al elemento <body>. Si se establece un fondo a la página, como el valor inicial del fondo de los



elementos es transparente, todos los elementos de la página se visualizan con el mismo fondo a menos que algún elemento especifique su propio fondo.

CSS define cinco propiedades para establecer el fondo de cada elemento (background-color, background-image, background-repeat, background-attachment, background-position) y otra propiedad de tipo "shorthand" (background).

El siguiente ejemplo muestra una página web con un color gris claro de fondo:

```
body {
background-color: #F5F5F5;
}
```

CSS permite establecer de forma simultánea un color y una imagen de fondo. En este caso, la imagen se muestra delante del color, por lo que solamente si la imagen contiene zonas transparentes es posible ver el color de fondo.

El siguiente ejemplo muestra una imagen como fondo de toda la página:

```
body { background-image: url("imagenes/fondo.png") }
```

Las imágenes de fondo se indican a través de su URL, que puede ser absoluta o relativa. Suele ser recomendable crear una carpeta de imágenes que se encuentre en el mismo directorio que los archivos CSS y que almacene todas las imágenes utilizadas en el diseño de las páginas.

Así, las imágenes correspondientes al diseño de la página se mantienen separadas del resto de imágenes del sitio y el código CSS es más sencillo (por utilizar URL relativas) y más fácil de mantener (por no tener que actualizar URL absolutas en caso de que se cambie la estructura del sitio web).

Por otra parte, suele ser habitual indicar un color de fondo siempre que se muestra una imagen de fondo. En caso de que la imagen no se pueda mostrar o contenga errores, el navegador mostrará el color indicado (que debería ser, en lo posible, similar a la imagen) y la página no parecerá que contiene errores.

Si la imagen que se quiere mostrar es demasiado grande para el fondo del elemento, solamente se muestra la parte de imagen comprendida en el tamaño del elemento. Si la imagen es más pequeña que el elemento, CSS la repite horizontal y verticalmente hasta llenar el fondo del elemento.

En ocasiones, no es conveniente que la imagen de fondo se repita horizontal y verticalmente.

Para ello, CSS introduce la propiedad background-repeat que permite controlar la forma de repetición de las imágenes de fondo.

El valor repeat indica que la imagen se debe repetir en todas direcciones y por tanto, es el comportamiento por defecto. El valor no-repeat muestra una sola vez la imagen y no se repite en ninguna dirección. El valor repeat-x repite la imagen sólo horizontalmente y el valor repeat-y repite la imagen solamente de forma vertical.

Las reglas CSS definidas para la cabecera son:

```
#hdr {
background: url(/images/ds.gif) repeat-x;
width: 100%;
text-align: center;
}
```

La propiedad background-position permite indicar la distancia que se desplaza la imagen de fondo respecto de su posición original situada en la esquina superior izquierda.



Si se utilizan solamente palabras clave, el orden es indiferente y por tanto, es equivalente indicar top left y left top.

El siguiente ejemplo muestra una misma imagen de fondo posicionada de tres formas diferentes: Las reglas CSS del ejemplo anterior se muestran a continuación:

```
#caja1 {
background-image: url(images/help.png);
background-repeat: no-repeat;
background-position: bottom left;
}
#caja2 {
background-image: url(images/help.png);
background-repeat: no-repeat;
background-position: right top;
}
<div id="caja1"><h1>bottom left</h1></div>
<div id="caja2"><h1>right top</h1></div>
```

5. Texto

5.1 Tipografía

CSS define numerosas propiedades para modificar la apariencia del texto. A pesar de que no dispone de tantas posibilidades como los lenguajes y programas específicos para crear documentos impresos, CSS permite aplicar estilos complejos y muy variados al texto de las páginas web.

La propiedad básica que define CSS relacionada con la tipografía se denomina color y se utiliza para establecer el color de la letra.

Aunque el color por defecto del texto depende del navegador, todos los navegadores principales utilizan el color negro. Para establecer el color de letra de un texto, se puede utilizar cualquiera de las cinco formas que incluye CSS para definir un color.

A continuación se muestran varias reglas CSS que establecen el color del texto de diferentes formas:

```
h1 { color: #369; }
p { color: black; }
a, span { color: #B1251E; }
div { color: rgb(71, 98, 176); }
```

Normalmente, se utiliza la propiedad color del elemento body para establecer el color de letra de todos los elementos de la página:

```
body { color: #777; }
```

En el ejemplo anterior, todos los elementos de la página muestran el mismo color de letra salvo que establezcan de forma explícita otro color. La única excepción de este comportamiento son los enlaces que se crean con la etiqueta <a>. Aunque el color de la letra se hereda de los elementos



padre a los elementos hijo, con los enlaces no sucede lo mismo, por lo que es necesario indicar su color de forma explícita:

```
/* Establece el mismo color a todos los elementos de la página salvo los enlaces */
body { color: #777; }
/* Establece el mismo color a todos los elementos de la página, incluyendo los enlaces */
body, a { color: #777; }
```

La otra propiedad básica que define CSS relacionada con la tipografía se denomina font-family y se utiliza para indicar el tipo de letra con el que se muestra el texto.

Los navegadores muestran el texto de las páginas web utilizando los tipos de letra instalados en el ordenador del propio usuario. De esta forma, si el diseñador indica en la propiedad font-family que el texto debe mostrarse con un tipo de letra especialmente raro o rebuscado, casi ningún usuario dispondrá de ese tipo de letra.

Para evitar el problema común de que el usuario no tenga instalada la fuente que quiere utilizar el diseñador, CSS permite indicar en la propiedad font-family más de un tipo de letra. El navegador probará en primer lugar con el primer tipo de letra indicado. Si el usuario la tiene instalada, el texto se muestra con ese tipo de letra.

Si el usuario no dispone del primer tipo de letra indicado, el navegador irá probando con el resto de tipos de letra hasta que encuentre alguna fuente que esté instalada en el ordenador del usuario. Evidentemente, el diseñador no puede indicar para cada propiedad font-family tantos tipos de letra como posibles fuentes parecidas existan.

Por todo ello, el valor de font-family suele definirse como una lista de tipos de letra alternativos separados por comas. El último valor de la lista es el nombre de la familia tipográfica genérica que más se parece al tipo de letra que se quiere utilizar.

Una vez seleccionado el tipo de letra, se puede modificar su tamaño mediante la propiedad font-size.

Además de todas las unidades de medida relativas y absolutas y el uso de porcentajes, CSS permite utilizar una serie de palabras clave para indicar el tamaño de letra del texto:

- tamaño_absoluto: indica el tamaño de letra de forma absoluta mediante alguna de las siguientes palabras clave: xx-small, x-small, small, medium, large, x-large, xx-large.
- tamaño_relativo: indica de forma relativa el tamaño de letra del texto mediante dos palabras clave (larger, smaller) que toman como referencia el tamaño de letra del elemento padre.

Una vez indicado el tipo y el tamaño de letra, es habitual modificar otras características como su anchura (texto en negrita) y su estilo (texto en cursiva). La propiedad que controla la anchura de la letra es font-weight.

Los valores que normalmente se utilizan son normal (el valor por defecto) y bold para los textos en negrita. El valor normal equivale al valor numérico 400 y el valor bold al valor numérico 700.

El siguiente ejemplo muestra una aplicación práctica de la propiedad font-weight:

Por defecto, los navegadores muestran el texto de los elementos en cursiva y el texto de los elementos en negrita. La propiedad font-weight permite alterar ese aspecto por defecto y mostrar por ejemplo los elementos como cursiva y negrita y los elementos destacados mediante un color de fondo y sin negrita.

Las reglas CSS del ejemplo anterior se muestran a continuación:

```
#especial em {
font-weight: bold;
}
#especial strong {
font-weight: normal;
background-color: #FFFF66;
```



```
padding: 2px;
```

```
}
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut in  
purus ac <em>libero nonummy vestibulum</em>. Nullam molestie, nunc id  
nonummy laoreet, <strong>tortor diam mollis elit</strong>, quis hendrerit  
libero lorem vitae nunc.</p>
```

```
<p id="especial">Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

```
Ut in purus ac <em>libero nonummy vestibulum</em>. Nullam molestie, nunc id  
nonummy laoreet, <strong>tortor diam mollis elit</strong>, quis hendrerit  
libero lorem vitae nunc.</p>
```

Además de la anchura de la letra, CSS permite variar su estilo mediante la propiedad `font-style`.

Normalmente la propiedad `font-style` se emplea para mostrar un texto en cursiva mediante el valor `italic`.

El único cambio necesario en las reglas CSS anteriores es el de añadir la propiedad `font-style`:

```
#especial em {
```

```
font-weight: bold;
```

```
font-style: normal;
```

```
}
```

```
#especial strong {
```

```
font-weight: normal;
```

```
font-style: italic;
```

```
background-color:#FFFF66;
```

```
padding: 2px;
```

```
}
```

5.2 Texto

Además de las propiedades relativas a la tipografía del texto, CSS define numerosas propiedades que determinan la apariencia del texto en su conjunto. Estas propiedades adicionales permiten controlar la alineación del texto, el interlineado, la separación entre palabras, etc.

La propiedad que define la alineación del texto se denomina `text-align`.

Los valores definidos por CSS permiten alinear el texto según los valores tradicionales: a la izquierda (`left`), a la derecha (`right`), centrado (`center`) y justificado (`justify`).

CSS también permite controlar la separación entre las letras que forman las palabras y la separación entre las palabras que forman los textos. La propiedad que controla la separación entre letras se llama `letter-spacing` y la separación entre palabras se controla mediante `word-spacing`.

```
.especial h1 { letter-spacing: .2em; }
```

```
.especial p { word-spacing: .5em; }
```

```
<div><h1>Lorem ipsum dolor sit amet</h1>
```



```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum dictum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Etiam pharetra blandit purus.</p>
```

```
...
```

```
</div>
```

```
<div class="especial"><h1>Lorem ipsum dolor sit amet</h1>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum dictum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Etiam pharetra blandit purus.</p>
```

```
...
```

```
</div>
```

Como ya se sabe, el tratamiento que hace HTML de los espacios en blanco es uno de los aspectos más difíciles de comprender cuando se empiezan a crear las primeras páginas web. Básicamente, HTML elimina todos los espacios en blanco sobrantes, es decir, todos salvo un espacio en blanco entre cada palabra.

Para forzar los espacios en blanco adicionales se debe utilizar la entidad HTML ` `; y para forzar nuevas líneas, se utiliza el elemento `
`. Además, HTML proporciona el elemento `<pre>` que muestra el contenido tal y como se escribe, respetando todos los espacios en blanco y todas las nuevas líneas.

CSS permite aplicar estilos a la primera letra del texto mediante el pseudo-elemento `:first-letter`. La siguiente imagen muestra el uso del pseudo-elemento `:first-letter` para crear una letra capital: El código HTML y CSS se muestra a continuación:

```
#normal p:first-letter {
```

```
font-size: 2em;
```

```
}
```

```
#avanzado p:first-letter {
```

```
font-size: 2.5em;
```

```
font-weight: bold;
```

```
line-height: .9em;
```

```
float: left;
```

```
margin: .1em;
```

```
}
```

```
#avanzado p:first-line {
```

```
font-weight: bold;
```

```
}
```

```
<div id="normal">
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum dictum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Etiam pharetra blandit purus.</p>
```

```
</div>
```




```
<div id="avanzado">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum
dictum. Class aptent taciti sociosqu ad litora torquent per conubia nostra,
per inceptos hymenaeos. Etiam pharetra blandit purus.</p>
</div>
```

6. Enlaces

6.1 Estilos básicos

6.1.1 Tamaño, color y decoración

Los estilos más sencillos que se pueden aplicar a los enlaces son los que modifican su tamaño de letra, su color y la decoración del texto del enlace.

A continuación se muestra un ejemplo :

```
a { margin: 1em 0; display: block;}
.alternativo {color: #CC0000;}
.simple {text-decoration: none;}
.importante {font-weight: bold; font-size: 1.3em;}
.raro {text-decoration:overline;}
<a href="#">Enlace con el estilo por defecto</a>
<a class="alternativo" href="#">Enlace de color rojo</a>
<a class="simple" href="#">Enlace sin subrayado</a>
<a class="importante" href="#">Enlace muy importante</a>
<a class="raro" href="#">Enlace con un estilo raro</a>
```

6.1.2 Pseudo-clases

CSS define cuatro pseudo-clases que permiten aplicar estilos avanzados para los enlaces de los documentos. Las pseudo-clases permiten aplicar diferentes estilos a un mismo enlace en función de su estado: enlace no visitado, enlace visitado, enlace en el que se pasa el puntero del ratón por encima y enlace activo en ese momento.

Cada una de las pseudo-clases definidas se muestra a continuación:

- `:link`, permite aplicar estilos para los enlaces que aún no han sido pinchados.
- `:visited`, aplica estilos a los enlaces que han sido pinchados anteriormente (el navegador elimina automáticamente el historial de enlaces visitados cada cierto tiempo).
- `:hover`, estilos que muestra el enlace cuando el usuario posiciona el puntero del ratón sobre el enlace.
- `:active`, estilos que se aplican al enlace cuando el usuario está pinchando sobre el enlace (el tiempo durante el que se aplica este estilo es muy breve).

El orden recomendado para la definición de las pseudo-clases de los enlaces es `:link`, `:visited`, `:hover`, `:active`.

Las pseudo-clases `:link` y `:visited` solamente están definidas para los enlaces, pero las pseudo-clases `:hover` y `:active` se definen para todos los elementos HTML.

Desafortunadamente, Internet Explorer 6 y sus versiones anteriores solamente las soportan para los enlaces.



Las pseudo-classes de los enlaces permiten variar el comportamiento por defecto que los navegadores aplican a los enlaces. El siguiente ejemplo oculta el subrayado a los enlaces cuando el usuario pasa el ratón por encima de un enlace:

```
a { }
a:hover { text-decoration: none; }
```

6.2 Estilos avanzados

6.2.1 Decoración personalizada

La propiedad `text-decoration` permite añadir o eliminar el subrayado de los enlaces. Sin embargo, el aspecto del subrayado lo controla automáticamente el navegador, por lo que su color siempre es el mismo que el del texto del enlace y su anchura es proporcional al tamaño de letra.

No obstante, utilizando la propiedad `border-bottom` es posible añadir cualquier tipo de subrayado para los enlaces de las páginas. El siguiente ejemplo muestra algunos enlaces con el subrayado personalizado:

```
a { margin: 1em 0; float: left; clear: left; text-decoration: none;}
.simple {text-decoration: underline;}
.color { border-bottom: medium solid #CC0000;}
.ancho {border-bottom: thick solid;}
.separado {border-bottom: 1px solid; padding-bottom: .6em;}
.discontinuo {border-bottom: thin dashed;}
<a class="simple" href="#">Enlace con el estilo por defecto</a>
<a class="color" href="#">Enlace un subrayado de otro color</a>
<a class="ancho" href="#">Enlace con un subrayado muy ancho</a>
<a class="separado" href="#">Enlace con un subrayado muy separado</a>
<a class="discontinuo" href="#">Enlace con un subrayado discontinuo</a>
```

6.2.2 Imágenes según el tipo de enlace

En ocasiones, puede resultar útil incluir un pequeño icono al lado de un enlace para indicar el tipo de contenido que enlaza, como por ejemplo un archivo comprimido o un documento con formato PDF.

Este tipo de imágenes son puramente decorativas en vez de imágenes de contenido, por lo que se deberían añadir con CSS y no con elementos de tipo ``. Utilizando la propiedad `background` (y `background-image`) se puede personalizar el aspecto de los enlaces para que incluyan un pequeño icono a su lado.

La técnica consiste en mostrar una imagen de fondo sin repetición en el enlace y añadir el `padding` necesario al texto del enlace para que no se solape con la imagen de fondo.

El siguiente ejemplo personaliza el aspecto de dos enlaces añadiéndoles una imagen de fondo:

```
a { margin: 1em 0; float: left; clear: left; }
.rss {
color: #E37529;
padding: 0 0 0 18px;
background: #FFF url(imagenes/rss.gif) no-repeat left center;
}
```



```
.pdf {
padding: 0 0 0 22px;
background: #FFF url(imagenes/pdf.png) no-repeat left center;
}
<a href="#">Enlace con el estilo por defecto</a>
<a class="rss" href="#">Enlace a un archivo RSS</a>
<a class="pdf" href="#">Enlace a un documento PDF</a>
```

6.2.3 Mostrar los enlaces como si fueran botones

Las propiedades definidas por CSS permiten mostrar los enlaces con el aspecto de un botón, lo que puede ser útil en formularios en los que no se utilizan elementos específicos para crear botones.

El siguiente ejemplo muestra un enlace simple formateado como un botón:

```
a { margin: 1em 0; float: left; clear: left; }
a.boton {
text-decoration: none;
background: #EEE;
color: #222;
border: 1px outset #CCC;
padding: .1em .5em;
}
a.boton:hover {
background: #CCB;
}
a.boton:active {
border: 1px inset #000;
}
<a class="boton" href="#">Guardar</a>
<a class="boton" href="#">Enviar</a>
```

7. Imágenes

7.1 Estilos básicos

7.1.1 Establecer la anchura y altura de las imágenes

Utilizando las propiedades width y height, es posible mostrar las imágenes con cualquier altura/anchura, independientemente de su altura/anchura real:

```
#destacada {
width: 120px;
height: 250px;
```



```
}

```

No obstante, si se utilizan alturas/anchuras diferentes de las reales, el navegador deforma las imágenes y el resultado estético es muy desagradable.

Por otra parte, establecer la altura/anchura de todas las imágenes mediante CSS es una práctica poco recomendable. El motivo es que normalmente dos imágenes diferentes no comparten la misma altura/anchura. Por lo tanto, se debe crear una nueva regla CSS (y un nuevo selector) para cada una de las diferentes imágenes del sitio web.

En la práctica, esta forma de trabajo produce una sobrecarga de estilos que la hace inviable. Por este motivo, aunque es una solución que no respeta la separación completa entre contenidos (XHTML) y presentación (CSS), se recomienda establecer la altura/anchura de las imágenes mediante los atributos `width` y `height` de la etiqueta ``:

```

```

7.1.2 Eliminar el borde de las imágenes con enlaces

Cuando una imagen forma parte de un enlace, los navegadores muestran por defecto un borde azul grueso alrededor de las imágenes. Por tanto, una de las reglas más utilizadas en los archivos CSS es la que elimina los bordes de las imágenes con enlaces:

```
img {
border: none;
}
```

8. Listas

8.1 Estilos básicos

8.1.1 Viñetas personalizadas

Por defecto, los navegadores muestran los elementos de las listas no ordenadas con una viñeta formada por un pequeño círculo de color negro. Los elementos de las listas ordenadas se muestran por defecto con la numeración decimal utilizada en la mayoría de países.

No obstante, CSS define varias propiedades para controlar el tipo de viñeta que muestran las listas, además de poder controlar la posición de la propia viñeta. La propiedad básica es la que controla el tipo de viñeta que se muestra y que se denomina `list-style-type`.

En primer lugar, el valor `none` permite mostrar una lista en la que sus elementos no contienen viñetas, números o letras. Se trata de un valor muy utilizado, ya que es imprescindible para los menús de navegación creados con listas.

El resto de valores de la propiedad `list-style-type` se dividen en tres tipos: gráficos, numéricos y alfabéticos.

- Los valores gráficos son `disc`, `circle` y `square` y muestran como viñeta un círculo relleno, un círculo vacío y un cuadrado relleno respectivamente.
- Los valores numéricos están formados por `decimal`, `decimal-leading-zero`, `lower-roman`, `upper-roman`, `armenian` y `georgian`.
- Por último, los valores alfanuméricos se controlan mediante `lower-latin`, `lower-alpha`, `upper-latin`, `upper-alpha` y `lower-greek`.

El código CSS de algunas de las listas se muestra a continuación:



```

<ul style="list-style-type: square">
<li>list-style-type: square</li>
<li>Elemento</li>
<li>Elemento</li>
</ul>
<ol style="list-style-type: lower-roman">
<li>list-style-type: lower-roman</li>
<li>Elemento</li>
<li>Elemento</li>
</ol>
<ol style="list-style-type: decimal-leading-zero; padding-left: 2em;">
<li>list-style-type: decimal-leading-zero</li>
<li>Elemento</li>
<li>Elemento</li>
</ol>

```

La propiedad `list-style-position` permite controlar la colocación de las viñetas.

Cuando se requiere personalizar el aspecto de las viñetas, se debe emplear la propiedad `list-style-image`, que permite mostrar una imagen propia en vez de una viñeta automática.

Las imágenes personalizadas se indican mediante la URL de la imagen. Si no se encuentra la imagen o no se puede cargar, se muestra la viñeta automática correspondiente (salvo que explícitamente se haya eliminado mediante la propiedad `list-style-type`).

El siguiente ejemplo muestra el uso de la propiedad `list-style-image`:

```

ul {
margin:0;
padding-left: 1.5em;
line-height: 1.5em;
}
ul li { padding-left: .2em; }
ul.ok { list-style-image: url(imagenes/ok.png); }
ul.go { list-style-image: url(imagenes/bullet_go.png); }
ul.redondo { list-style-image: url(imagenes/bullet_red.png); }
<div>
<ul class="ok">
<li>Elemento</li>
<li>Elemento</li>
<li>Elemento</li>
<li>Elemento</li>
</ul>
</div>

```



```

<div>
<ul class="go">
<li>Elemento</li>
<li>Elemento</li>
<li>Elemento</li>
<li>Elemento</li>
</ul>
</div>
<div>
<ul class="redondo">
<li>Elemento</li>
<li>Elemento</li>
<li>Elemento</li>
<li>Elemento</li>
</ul>
</div>

```

Como es habitual, CSS define una propiedad de tipo "shorthand" que permite establecer todas las propiedades de una lista de forma directa. El siguiente ejemplo indica que no se debe mostrar ni viñetas automáticas ni viñetas personalizadas:

```
ul { list-style: none }
```

Cuando se utiliza una viñeta personalizada, es conveniente indicar la viñeta automática que se mostrará cuando no se pueda cargar la imagen:

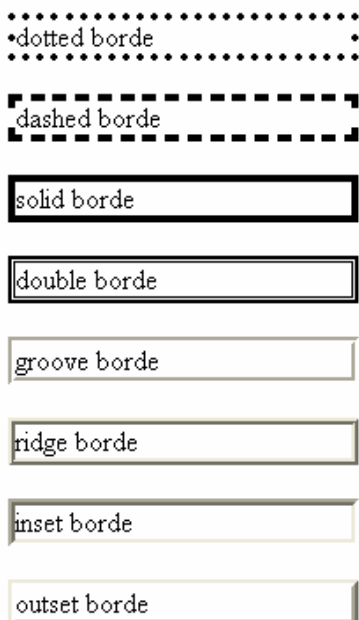
```
ul { list-style: url(imagenes/cuadrado_rojo.gif) square; }
```

9. Ejercicios

1. Realiza una página web que muestre la imagen de fondo repetida verticalmente.
2. Realiza una página web que muestre la imagen de fondo centrada.
3. Realiza una página web en la que la cabecera 1 sea verde, la 2 roja y el párrafo azul, utilizando la notación hexadecimal y la decimal.
4. Realiza una página web en la que el fondo del párrafo es amarillo y la letra azul.
5. Realiza una página web en la que "Texto azul y" sea la letra azul y "fondo amarillo" la letra sea azul y el fondo amarillo.
6. Realiza una página web en la que el espacio entre palabras sea de 30px.
7. Realiza una página web en la que la cabecera 1 esté alineada al centro, la 2 a la izquierda y la 3 a la derecha.
8. Realiza una página web en la que el tipo de letra sea comic sans ms y si no existe este tipo de letra que la muestre en verdana.



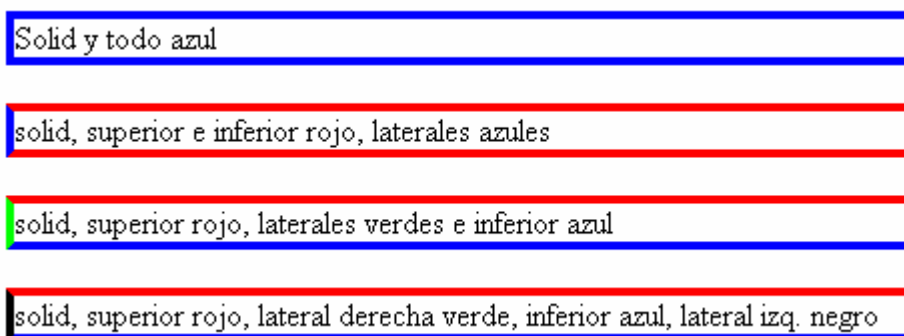
9. Realiza una página web en la que el tamaño de la letra sea el doble de la normal utilizando porcentajes.
10. Realiza una página web en la que la letra del párrafo sea arial, tenga un tamaño de 12 píxels, sólo en una declaración.
11. Utilizando la definición de bordes, realiza la siguiente página.



12. Realiza una página web como la que se muestra en la imagen, el borde debe ser de color rojo.



13. Realiza la siguiente página web.



14. Realiza una página en la que el texto tengo borde y sólo el borde derecho tenga color rojo.
15. Realiza una página en la que el texto este a una distancia respecto al margen o respecto a otros textos de: 5cm en la parte superior, 4cm en la parte izquierda, 3cm en la parte inferior y 2cm en la parte derecha.



16. Realiza una página web en la que los iconos de la lista sean imágenes elegida por ti mismo.
17. Realiza una página web en la que el espacio entre líneas sea 40px.
18. Realiza la siguiente página web:

