

Nombre: Jose Antonio Sánchez López

# APUNTES

## Índice

APUNTES.....	1
<b>Capítulo 1. Introducción.....</b>	<b>3</b>
1.1. ¿Qué es CSS?.....	3
1.2. Breve historia de CSS. ....	3
1.3. Soporte de CSS en los navegadores.....	3
1.4. Especificación oficial.....	4
1.5. Funcionamiento básico de CSS.....	5
1.6. Cómo incluir CSS en un documento XHTML.....	6
1.6.1. Incluir CSS en el mismo documento HTML.....	6
1.6.2. Definir CSS en un archivo externo. ....	7
1.6.3. Incluir CSS en los elementos HTML. ....	8
1.7. Glosario básico.....	8
1.8. Medios CSS.....	9
1.8.1. Medios definidos con las reglas de tipo @media. ....	10
1.8.2. Medios definidos con las reglas de tipo @import.....	10
1.8.3. Medios definidos con la etiqueta <link>. ....	11
1.8.4. Medios definidos mezclando varios métodos.....	11
1.9. Comentarios. ....	11
1.10. Sintaxis de la definición de cada propiedad CSS.....	11
<b>Capítulo 2. Selectores. ....</b>	<b>12</b>
2.1. Selectores básicos. ....	13
2.1.1. Selector Universal.....	13
2.1.2. Selector de tipo o etiqueta. ....	13
2.1.3. Selector descendente. ....	13
2.1.4. Selector de clase. ....	13
2.1.5. Selectores de ID. ....	15
2.1.6. Combinación de selectores básicos. ....	15
2.2. Selectores avanzados. ....	15
2.2.1. Selector de hijos. ....	16
2.2.2. Selector adyacente.....	16
2.2.3. Selector de atributos.....	16
2.3. Agrupación de reglas.....	17
2.4. Herencia.....	17
2.5. Colisiones de estilos.....	17

Nombre: Jose Antonio Sánchez López

Fecha: 23/12/2020

<b>Capítulo 3. Unidades de medida y colores.</b>	18
3.1. Unidades de medida.	18
3.1.1 Unidades relativas.	18
3.1.2. Unidades absolutas.	19
3.1.3. Porcentajes.	19
3.1.4. Recomendaciones.	19
3.2. Colores.	20
3.2.1 Palabras clave.	20
3.2.2. RGB decimal.	20
3.2.3. RGB porcentual.	20
3.2.4. RGB hexadecimal.	20
3.2.5. Colores del sistema.	21
3.2.6. Colores web safe.	21

Nombre: Jose Antonio Sánchez López

## Capítulo 1. Introducción.

### 1.1. ¿Qué es CSS?

Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de documentos HTML y XHTML. Es imprescindible para crear páginas web complejas.

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. Breve historia de CSS.

Apareció poco después que el lenguaje de etiquetas SGML alrededor de 1970. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos con la misma apariencia en diferentes navegadores.

Las dos propuestas que se tuvieron en cuenta fueron la CHSS (Cascading HTML Style Sheets) y la SPP (Stream-based Style Sheet Proposal).

Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que cogía lo mejor de cada propuesta y lo llamaron CSS (Cascading Style Sheets).

En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML.

A finales de 1996 se publicó la primera recomendación oficial, llamada “**CSS nivel 1**”.

A principios de 1997 el W3C decide separar en tres secciones: el grupo de trabajo de HTML, el grupo de trabajo de DOM y el grupo de trabajo de CSS.

El 12 de mayo de 1998 aparece la segunda recomendación oficial, conocida como “**CSS nivel 2**”. La versión de CSS que utilizan todos los navegadores actualmente es CSS3, es la última evolución de CSS 2.1.

El primer navegador con soporte completo de CSS1 fue la versión para Mac de Internet Explorer 5, publicado en el año 2000. Actualmente todo los navegadores tienen un soporte completo para CSS 2.1.

### 1.3. Soporte de CSS en los navegadores.

Nombre: Jose Antonio Sánchez López

El trabajo de diseñador web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.

La parte del navegador se encarga de interpretar el código HTML y CSS para mostrar las páginas se denomina motor.

En la siguiente tabla se muestra el soporte de CSS1, CSS 2.1, CSS3 de los navegadores más utilizados:

Navegador	Motor	CSS 1	CSS 2.1	CSS 3
Google Chrome	WebKit	Completo desde la versión 85 del motor	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Internet Explorer	Trident	Completo desde la versión 7.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades a partir de la versión 10.0 del navegador
Firefox	Gecko	Completo desde la versión 1.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Safari	WebKit	Completo desde la versión 85 del motor	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo desde la versión 1.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades

Los navegadores Safari y Opera son los más avanzados en el soporte de CSS, ya que incluyen muchos elementos de la versión de CSS3 y un soporte perfecto para la versión 2.1.

Por su parte, el navegador Internet Explorer sólo puede considerarse adecuado desde el punto de vista de CSS a partir de su versión 7. Internet Explorer tiene una versión actual v11.0.210 con el soporte de hasta CSS3.

#### 1.4. Especificación oficial.

Nombre: Jose Antonio Sánchez López

Fecha: 23/12/2020

La especificación o norma oficial que se utiliza actualmente para diseñar páginas web con CSS es la versión CSS 2.1, actualizada por última vez el 19 de julio de 2007.

La versión CSS3 ya está elaborada y es mucho más avanzada y más compleja que CSS2.

CSS3 ya está implementado en los navegadores actuales.

### 1.5. Funcionamiento básico de CSS.

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.

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 xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<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 <font> 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 <font>.

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 <font>, 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 xmlns="http://www.w3.org/1999/xhtml">
<head>
```

Nombre: Jose Antonio Sánchez López

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<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. 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, ejemplo: si ponemos que <h1> se debe ver rojo con tipo de letra Arial, y con tamaño de letra grande, todas las etiquetas <h1> aparecerán de esa manera.

#### 1.6. Cómo incluir CSS en un documento XHTML.

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.

##### 1.6.1. Incluir CSS en el mismo documento HTML.

Se emplea la etiqueta <style> de HTML y solamente se pueden incluir en la cabecera del documento (dentro de la sección de <head>).

Ejemplo:

```
<!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 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>
```

Nombre: Jose Antonio Sánchez López

Fecha: 23/12/2020

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 de este método 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.

#### 1.6.2. 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 se necesiten y cada pag HTML puede enlazar tantos archivos CSS como necesite.

Se crea un archivo de texto y se le añade solamente el siguiente contenido: p  
{ color: black; font-family: Verdana; }

- Se guarda el archivo de texto con el nombre de estilos.css Se debe poner especial atención a que el archivo tenga extensión `.css` y no `.txt`.
- En la página HTML se enlaza el archivo CSS externo mediante la etiqueta `<link>`.

Cuando el navegador carga la página HTML anterior, antes de mostrar sus contenidos también descarga los archivos CSS externos enlazados mediante la etiqueta `<link>` y aplica los estilos a los contenidos de la página.

La etiqueta `<link>` incluye cuatro atributos:

- rel
- type
- href
- media

De todas las formas de incluir CSS en las páginas HTML, esta es la más utilizada con mucha diferencia. Su principal ventaja es que se puede incluir un mismo archivo CSS en multitud de páginas HTML.

El mantenimiento se simplifica al máximo, ya que con un solo cambio en un solo archivo CSS permite variar de forma instantánea los estilos de todas las páginas.

Nombre: Jose Antonio Sánchez López

Fecha: 23/12/2020

Normalmente se usa la etiqueta `<link>` para enlazar los archivos CSS externos, también se puede utilizar `<style>`.

### 1.6.3. Incluir CSS en los elementos HTML.

El último método para incluir CSS es el peor y el menos utilizado ya que tiene los mismos problemas que usar la etiqueta `<font>`.

Ejemplo:

```
<!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 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.

### 1.7. Glosario básico.

CSS define una serie de términos que permiten describir cada una de las partes que componen los estilos CSS.

Los diferentes términos son:

- Regla
- Selector



Nombre: Jose Antonio Sánchez López

- Declaración
- Propiedad
- Valor

Un archivo CSS puede contener infinitas reglas CSS, cada regla puede contener infinitos selectores y cada declaración puede estar formada por un número infinito de pares propiedad/valor.

### 1.8. Medios CSS.

Una de las características más importantes de las hojas de estilos CSS es que permiten definir diferentes estilos para diferentes medios o dispositivos: pantallas, impresoras, móviles, proyectores, etc.

Medio	Descripción
all	Todos los medios definidos
braille	Dispositivos táctiles que emplean el sistema braille
embosed	Impresoras braille
handheld	Dispositivos de mano: móviles, PDA, etc.

print	Impresoras y navegadores en el modo "Vista Previa para Imprimir"
projection	Proyectores y dispositivos para presentaciones
screen	Pantallas de ordenador
speech	Sintetizadores para navegadores de voz utilizados por personas discapacitadas
tty	Dispositivos textuales limitados como teletipos y terminales de texto
tv	Televisores y dispositivos con resolución baja

Los medios más utilizados actualmente son screen y print, seguidos de handheld.

Además CSS clasifica a los medios en diferentes grupos según sus características.

Medio	Continuo / Paginado	Visual / Auditivo / Táctil / Vocal	Mapa de bits / Caracteres	Interactivo / Estático
braille	continuo	táctil	caracteres	ambos
embossed	paginado	táctil	caracteres	estático
handheld	ambos	visual, auditivo, vocal	ambos	ambos
print	paginado	visual	mapa de bits	estático
projection	paginado	visual	mapa de bits	interactivo
screen	continuo	visual, auditivo	mapa de bits	ambos
speech	continuo	vocal	(no tiene sentido)	ambos
tty	continuo	visual	caracteres	ambos
tv	ambos	visual, auditivo	mapa de bits	ambos

La gran ventaja de CSS es que permite modificar los estilos de una página en función del medio en el que se visualiza. Existen cuatro formas diferentes de indicar el medio en que el que se deben aplicar los estilos CSS.

#### 1.8.1. Medios definidos con las reglas de tipo @media.

Las reglas @media son un tipo especial de reglas CSS que permiten indicar de forma indirecta el medio o medios en los que se aplicarán los estilos incluidos en la regla.

Ejemplo:

```
@media print {
  body { font-size: 10pt }
}
@media screen {
  body { font-size: 13px }
}
@media screen, print {
  body { line-height: 1.2 }
}
```

#### 1.8.2. Medios definidos con las reglas de tipo @import.

Cuando se utilizan reglas de tipo @import para enlazar archivos CSS externos, se puede especificar el medio en el que se aplican los estilos indicando el nombre del medio después de la URL del archivo CSS:

```
@import url("estilos_basicos.css") screen;
```

Nombre: Jose Antonio Sánchez López  
`@import url("estilos_impresora.css") print;`

Las reglas del ejemplo anterior establecen que cuando la página se visualiza por pantalla, se cargan los estilos definidos en el primer archivo CSS.

Si los estilos del archivo CSS externo deben aplicarse en varios medios, se indican los nombres de todos los medios separados por comas.

### 1.8.3. Medios definidos con la etiqueta <link>.

Si se utiliza la etiqueta <link> para enlazar los archivos CSS externos, se puede utilizar el atributo `media` para indicar el medio o medios en los que se aplican los estilos de cada archivo:

```
<link rel="stylesheet" type="text/css" media="screen" href="basico.css" />  
<link rel="stylesheet" type="text/css" media="print, handheld" href="especial.css" />
```

### 1.8.4. Medios definidos mezclando varios métodos.

CSS también permite mezclar los tres métodos.

```
<link rel="stylesheet" type="text/css" media="screen" href="basico.css" />  
@import url("estilos_seccion.css") screen;  
@media print {  
/* Estilos específicos para impresora */  
}
```

### 1.9. Comentarios.

CSS permite incluir comentarios entre sus reglas y estilos. Son contenidos de texto que el diseñador incluye en el archivo CSS para su propia información y utilidad, el navegador lo ignora.

```
/* Este es un comentario CSS */
```

```
/* Esto es un
```

```
Comentario en varias
```

```
Líneas */
```

`<!-- Este es un comentario de HTML -->` No hay que confundirlo con comentarios en CSS.

### 1.10. Sintaxis de la definición de cada propiedad CSS.

Nombre: Jose Antonio Sánchez López

Fecha: 23/12/2020

La definición formal se basa en la información recogida en el estándar oficial y se muestra en forma de tabla.

Si el valor permitido se indica como una sucesión de palabras sin ningún carácter que las separe (paréntesis, comas, barras, etc) el valor de la propiedad se debe indicar tal y como se muestra y con esas palabras en el mismo orden.

Si el valor permitido se indica como sucesión de valores separados por una barra simple | el valor de la propiedad debe tomar uno y sólo uno de los valores indicados.

Si el valor permitido se indica como una sucesión de valores separados por una barra doble || el valor de la propiedad puede tomar uno o más valores de los indicados y en cualquier orden.

Por último, en cada valor o agrupación de valores se puede indicar el tipo de valor: opcional, obligatorio, múltiple o restringido.

El carácter \* indica que el valor ocurre cero o más veces; el carácter + indica que el valor ocurre una o más veces; el carácter ? indica que el valor es opcional y por último, el carácter {número\_1, número\_2} indica que el valor ocurre al menos tantas veces como el valor indicado en número\_1 y como máximo tantas veces como el valor indicado en número\_2.

No obstante, la mejor forma de entender la notación formal para las propiedades de CSS es observar la definición de cada propiedad y volver a esta sección siempre que sea necesario.

## Capítulo 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”.

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

CSS 2.1 incluye una docena de tipos diferentes de selectores.

Nombre: Jose Antonio Sánchez López

Fecha: 23/12/2020

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 Universal.

El selector universal se indica mediante un asterisco (\*).

### 2.1.2. Selector de tipo o etiqueta.

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector.

```
P {
```

```
...
```

```
}
```

### 2.1.3. Selector descendente.

El selector del siguiente ejemplo selecciona todos los elementos `<span>` de la página que se encuentren dentro de un elemento `<p>`:

```
p span { color: red; }
```

Si el código HTML de la página es el siguiente:

```
<p>
```

```
...
```

```
<span>texto1</span>
```

```
...
```

```
<a href="">...<span>texto2</span></a>
```

```
...
```

```
</p>
```

En el siguiente ejemplo, el selector descendente se compone de cuatro partes:

```
p a span em { text-decoration: underline; }
```

Los estilos de la regla anterior se aplican a los elementos de tipo `<em>` que se encuentren dentro de elementos de tipo `<span>`, que a su vez se encuentren dentro de elementos de tipo `<a>` que se encuentren dentro de elementos de tipo `<p>`.

### 2.1.4. Selector de clase.

Nombre: Jose Antonio Sánchez López

Fecha: 23/12/2020

Una de las soluciones más sencillas para aplicar estilos a un solo elemento de la página consiste en utilizar el atributo class de HTML sobre ese elemento para indicar directamente la regla CSS que se le debe aplicar:

```
<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>
```

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; }
```

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 utilizar 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>
```

¿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”.

No debe confundirse el selector de clase con los selectores anteriores:

```
/* Todos los elementos de tipo "p" con atributo class="aviso" */
p.aviso { ... }
/* Todos los elementos con atributo class="aviso" que estén dentro
  de cualquier elemento de tipo "p" */
p .aviso { ... }
/* Todos los elementos "p" de la página y todos los elementos con
  atributo class="aviso" de la página */
```

Nombre: Jose Antonio Sánchez López

p, .aviso { ... }

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

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

#### 2.1.5. Selectores de ID.

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 de 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>
```

Al igual que los selectores de clase, en este caso también se puede restringir el alcance del selector mediante la combinación con otros selectores. El siguiente ejemplo aplica la regla CSS solamente al elemento de tipo <p> que tenga un atributo id igual al indicado:

```
p#aviso { color: blue; }
```

#### 2.1.6. Combinación de selectores básicos.

CSS permite la combinación de uno o más tipos de selectores para restringir el alcance de las reglas CSS.

#### 2.2. Selectores avanzados.

Utilizando solamente los selectores básicos de la sección anterior, es posible diseñar prácticamente cualquier página web. No obstante, CSS define otros selectores más avanzados que permiten simplificar las hojas de estilos.

Nombre: Jose Antonio Sánchez López

### 2.2.1. Selector de hijos.

Se utilizan para seleccionar un elemento que es hijo directo de otro elemento y se indica mediante el “signo de mayor que” >:

```
p > span { color: blue; }  
<p><span>Texto1</span></p>  
<p><a href="#"><span>Texto2</span></a></p>
```

En el ejemplo anterior, el selector `p > span` se interpreta como “cualquier elemento `<span>` que se hijo directo de un elemento `<p>`”. por lo que el primer elemento `<span>` cumple la condición del selector. Sin embargo, el segundo elemento `<span>` no la cumple porque es descendiente pero no es un hijo directo de un elemento `<p>`.

### 2.2.2. Selector adyacente.

El selector adyacente utiliza el signo + y su sintaxis es:

Elemento1 + elemento 2 {.....}

La explicación del comportamiento de este selector no es sencilla, ya que selecciona todos los elementos de tipo elemento2 que cumplan las dos siguientes condiciones:

- Elemento1 y elemento2 deben ser hermanos, por lo que su elemento padre debe ser el mismo.
- Elemento2 debe aparecer inmediatamente después del elemento1 en el código HTML de la página.

### 2.2.3. Selector de atributos.

El último tipo de selectores avanzados lo forman los selectores de atributos, que permiten seleccionar elementos HTML en función de sus atributos y/o valores de esos atributos.

Hay cuatro tipos de selectores:

- `[nombre_atributo]`, selecciona los elementos que tienen establecido el atributo llamado `nombre_atributo`, independientemente de su valor.
- `[nombre_atributo=valor]`, selecciona los elementos que tienen establecido un atributo llamado `nombre_atributo` con un valor igual a `valor`.



Nombre: Jose Antonio Sánchez López

Fecha: 23/12/2020

- `[nombre_atributo~=valor]`, selecciona los elementos que tienen establecido un atributo llamado `nombre_atributo` y al menos uno de los valores del atributo es `valor`.
- `[nombre_atributo|=valor]`, selecciona los elementos que tienen establecido un atributo llamado `nombre_atributo` y cuyo valor es una serie de palabras separadas con guiones, pero que comienza con `valor`. Este tipo de selector sólo es útil para los atributos de tipo `lang` que indican el idioma del contenido del elemento.

### 2.3. 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;  
}
```

### 2.4. 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.

### 2.5. Colisiones de estilos.

En las hojas de estilos complejas, es habitual que varias reglas CSS se apliquen e aun mismo elemento HTML. El problema de estas reglas múltiples es que se pueden dar colisiones como la del siguiente ejemplo:

Nombre: Jose Antonio Sánchez López

p { color: red; }

p { color: blue; }

para evitar colisiones:

1. Determinar todas las declaraciones que se aplican al elemento para el medio CSS seleccionado.
2. Ordenar las declaraciones según su origen (CSS de navegador, de usuario o de diseñador) y su importancia (palabra clave !important).
3. Ordenar las declaraciones según lo específico que sea el selector. Cuanto más genérico es un selector, menos importancia tienen sus declaraciones.
4. Si después de aplicar las normas anteriores existen dos o más reglas con la misma prioridad, se aplica la que se indicó en último lugar.

De momento, la norma que se puede seguir es la de la “especificidad” del selector:

- Cuanto más específico sea un selector, más importancia tiene su regla asociada.
- A igual especificidad, se considera la última regla indicada.

## Capítulo 3. Unidades de medida y colores.

### 3.1. Unidades de medida.

Las medidas en CSS se emplean, entre otras cosas, para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto.

CSS divide todas las unidades de medida en dos grupos: absolutas y relativas.

#### 3.1.1 Unidades relativas.

Las unidades relativas son más flexibles que las unidades absolutas porque se adaptan más fácilmente a los diferentes medios.

- em. Aunque no es una definición exacta, el valor de 1em se puede aproximar por la anchura de la letra M del tipo de letra que se esté utilizando.
- ex, relativa respecto de la altura de la letra x.
- px

Nombre: Jose Antonio Sánchez López

### 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.

- 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)

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 otra referencias.

### 3.1.3. Porcentajes.

CSS define otra unidad de medida relativa basada en los porcentajes. Un porcentaje está formado por un valor numérico seguido del símbolo % y siempre está referenciado a otra medida.

Los porcentajes se pueden utilizar por ejemplo para establecer el valor del tamaño de los elementos:

```
body { font-size: 1em; }  
h1 { font-size: 200%; }  
h2 { font-size: 150%; }
```

### 3.1.4. Recomendaciones.

En general, se recomienda el uso de unidades relativas siempre que sea posible, ya que mejora la accesibilidad de la página y permite que los documentos se adapten fácilmente a cualquier medio y dispositivo.

Normalmente se utilizan píxel y porcentajes para definir el layout del documento (básicamente, la anchura de las columnas y elementos de las páginas) y em y porcentajes para el tamaño de letra de los textos.

Nombre: Jose Antonio Sánchez López

Fecha: 23/12/2020

Por otra parte uno de los problemas habituales cuando se utilizan unidades relativas es el problema de “el texto cada vez se ve mas pequeño” o “el texto cada vez se ve mas grande”. El siguiente ejemplo muestra el primer caso:

```
div { font-size: 0.9em; }  
<div>  
  <p>Texto 1</p>  
  <div>  
    <p>Texto 2</p>  
    <div>  
      <p>Texto 3</p>  
    </div>  
  </div>  
</div>
```

### 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.

#### 3.2.1 Palabras clave.

CSS define 17 palabras clave para referirse a los colores básicos.

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

#### 3.2.2. RGB decimal.

Los dos modelos más conocidos son RGB y CMYK.

En el modelo RGB un color se define indicando sus tres componentes R (rojo), G (verde) y B (azul).

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

#### 3.2.3. RGB porcentual.

Otra forma de indicar las componentes RGB de un color es mediante un porcentaje.

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

#### 3.2.4. RGB hexadecimal.

Nombre: Jose Antonio Sánchez López

Fecha: 23/12/2020

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

Por tanto, para definir un color en CSS con RGB hexadecimal se realizan los siguientes pasos:

1. Se determinan las componentes RGB del color original, por ejemplo: R = 71, G = 98, B = 176.
2. El valor numérico de cada componente se transforma al sistema numérico hexadecimal. Se trata de una operación exclusivamente matemática, por lo que puedes utilizar una calculadora. En el ejemplo anterior, el valor hexadecimal de cada componente es: R = 47, G = 62, B = B0.
3. Para obtener el color completo en formato RGB hexadecimal, se concatenan los valores de las componentes RGB en ese orden y se les añade el prefijo #. De esta forma, el color del ejemplo anterior es #4762B0 en formato RGB hexadecimal.

Una de las ventajas del formato RGB hexadecimal es que se pueden comprimir sus valores cuando todas sus componentes son iguales dos a dos:

```
#AAA = #AAAAAA  
#FFF = #FFFFFF  
#A0F = #AA00FF  
#369 = #336699
```

### 3.2.5. Colores del sistema.

Los colores del sistema son similares a los colores indicados mediante su nombre, pero en este caso hacen referencia al color que muestran algunos elementos del sistema operativo del usuario.

### 3.2.6. Colores web safe.

Como cada componente RGB de los colores puede tomar un valor entre 0 y 255, el número total de colores que se pueden representar con este formato es de  $256 \times 256 \times 256 = 16.777.216$  colores. Sin embargo, en la década de los 90 los monitores de los usuarios no eran capaces de mostrar más de 256 colores diferentes. A partir de todos los colores disponibles, se eligieron 216 colores que formaron la paleta de colores "web safe".

Nombre: Jose Antonio Sánchez López

Fecha: 23/12/2020

Hoy en día, su importancia ha descendido notablemente, no obstante, el auge en el uso de los dispositivos móviles hace que siga siendo un tema a considerar, ya que las pantallas de muchos móviles sólo pueden representar un número reducido de colores.