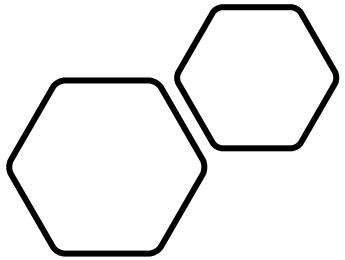
Curso 2023-2024



UD3: Hojas de estilo CSS

Lenguajes de marcas y sistemas de gestión de información

CSS

- CSS es el lenguaje que usamos para modificar el aspecto de las páginas web escritas en HTML.
- Esto permite separar los contenidos (HTML) de la presentación (CSS).
- En el proceso de creación de páginas web, se utiliza primero HTML para crear los contenidos, designando la función de cada elemento dentro de la página y una vez creado se usa CSS para definir el aspecto de cada elemento.
- Al igual que HTML, CSS está estandarizado por el W3C: https://www.w3.org/Style/CSS/
- CSS3 está dividido en varios documentos separados, llamados "módulos“.
- Usar CSS es una buena idea porque:
  - HTML no fue en inicio un lenguaje para definir el aspecto de la página.
  - Para cambiar el aspecto de todo un sitio web es suficiente con hacer cambios en un único archivo.

¿Cómo incluir CSS en nuestras páginas web?

1. Como CSS en línea:
   - Los estilos se incluyen en el atributo *style* **de cada elemento HTML**.
   - Esta solución **no es aconsejable**, ya que no separa la presentación de la estructura.
2. Como CSS incrustado:
   - El CSS se incluye en el propio documento HTML. Los estilos se definen en una zona concreta del documento .html, dentro de la etiqueta <style>, que tiene que ir obligatoriamente dentro del elemento <head>.
   - Únicamente usaremos este método cuando el número de estilos que tenemos es muy reducido, o bien cuando queremos incluir en una página concreta estilos que complementen a los estilos por defecto del resto de páginas de nuestro sitio web.

¿Cómo incluir CSS en nuestras páginas web?

```
<p style="color: red; font-size: 12pt; ">Primer ejemplo de CSS (En línea).</p>
```

```
<!DOCTYPE html>

<html>
    <head>
        <title>CSS incrustado en el HTML</title>
        <style>
            p {color: red; font-size: 12pt;}
        </style>
    </head>
    <body>
        <p>Segundo ejemplo de CSS (Incrustado).</p>
    </body>
</html>
```

¿Cómo incluir CSS en nuestras páginas web?

3. Como CSS vinculado:
   - Todos los estilos CSS se incluyen en un archivo independiente, de tipo CSS (.css) al que los documentos HTML hacen referencia mediante la etiqueta <link>, que debe estar situada en la cabecera <head>.
   - Este es el método **más recomendable** y tiene la ventaja de que una misma hoja de estilos puede emplearse en múltiples documentos HTML y un cambio a un solo archivo CSS permite modificar de forma instantánea todas las páginas HTML que lo enlazan.

¿Cómo incluir CSS en nuestras páginas web?

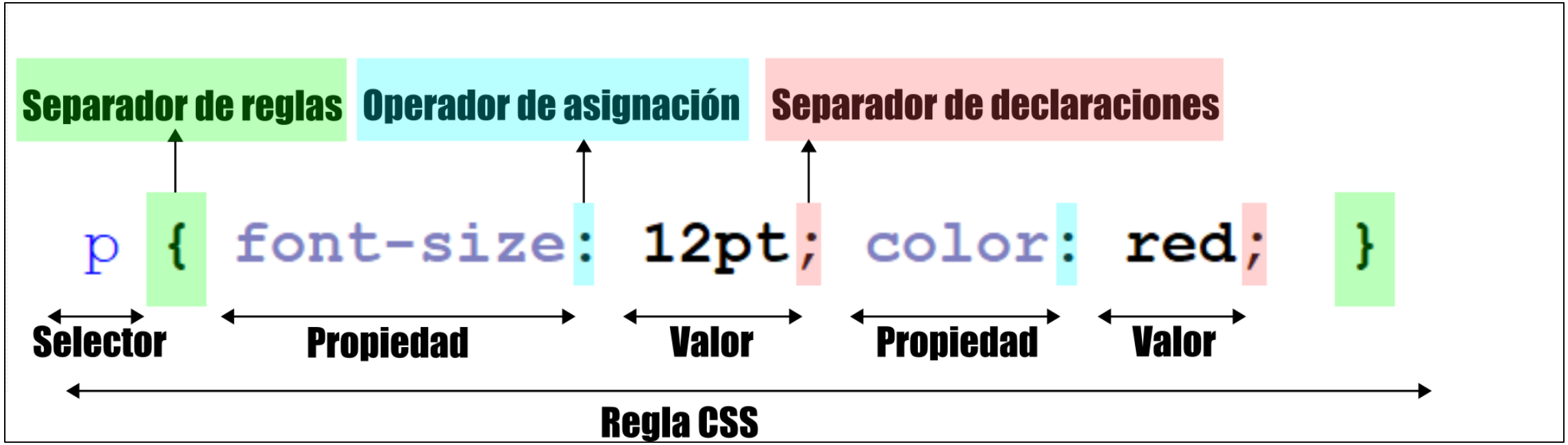
```html
<!DOCTYPE html>

<html>
    <head>
        <title>CSS vinculado</title>
        <link rel="stylesheet" type="text/css" href="css/ejemplo3.css"/>
    </head>
    <body>
        <p>Tercer ejemplo de CSS (Vinculado).</p>
    </body>
</html>
```

```css
p {color: red; font-size: 12pt;}
```

¿Cómo incluir CSS en nuestras páginas web?

- En caso de combinar varias formas de incluir estilos en un documento HTML, y estos estilos afectan a un mismo elemento… ¿Cuál tendrá prioridad?

Sintaxis CSS



Sintaxis CSS

- **Regla:** Cada uno de los estilos que componen la hoja de estilos.
- **Selector:** Indica el elemento o elementos HTML sobre los que se aplica la regla.
- **Declaración:** Especifica los estilos que se aplican a los elementos. Cada declaración termina siempre en **;**
- **Propiedad:** Característica que se modifica en el elemento seleccionado.
- **Valor:** Establece el valor de dicha característica.

Comentarios en CSS

- Para hacer comentarios en HTML usamos el formato

<!-- comentario -->

- Sin embargo, para hacer comentarios en las hojas de estilo CSS usamos el formato:

/* comentario */

Selectores

- **Selector universal**
  - Se utiliza para seleccionar **todos los elementos** de la página.
  - Usando el siguiente ejemplo pondremos en rojo todos los textos de la página:

```
* { color: red;}
```

  - El selector universal se indica mediante un asterisco (*). No se utiliza habitualmente, ya que **es difícil que un mismo estilo se pueda aplicar a todos los elementos** de una página.
  - No debemos confundir el selector universal con aplicar una regla a la etiqueta <body>.

Selectores

- **Selector de etiqueta**
  - Selecciona todos los elementos de la página cuyo nombre de 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.

  - Podemos **agrupar varios selectores** de etiqueta para los que vayamos a usar las mismas reglas separándolos con una coma:

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

Selectores

- Cuando las hojas de estilo van creciendo en tamaño, es práctica habitual definir las propiedades comunes de varios elementos agrupando sus selectores y posteriormente especificando las propiedades individuales de cada uno:

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

Selectores

- **Selector descendente**: selecciona a los elementos que son descendientes de otros elementos, es decir, están contenidos dentro de éstos (entre la etiqueta de apertura y de cierre).

```
p a {background-color: red;}
```

  - Con esta regla estamos indicando que todos los elementos <a> que son descendientes de un elemento (están contenidos entre la apertura y el cierre de una etiqueta p) tendrán un color de fondo rojo.
  - Esta regla se aplicaría tanto a:

 <a>

como a:

   <a>

Selectores

```
h1 p {background-color: yellow;}
h2 p {background-color: green;}
```

La primera regla se aplicará a los elementos <p> descendientes de una etiqueta h1, como por ejemplo:

```
<h1> <div> <p> </p> </div> </h1>
```

La segunda regla se aplicará a los elementos <p> descendientes de una etiqueta h2, como por ejemplo:

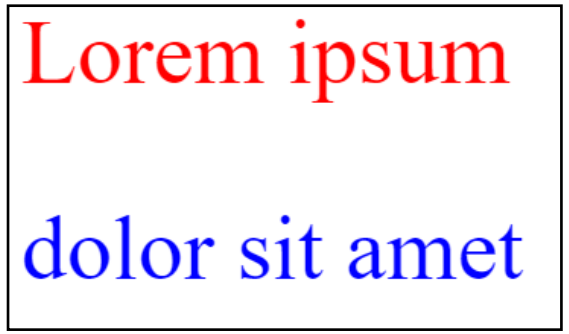
```
<h2> <div> <p> </p> </div> </h2>
```

- Los selectores descendentes siempre estarán formados por dos o más selectores separados entre sí por espacios en blanco. El último selector indica el elemento sobre el que se aplican los estilos y todos los anteriores determinan la posición en la que se debe encontrar dicho elemento.

Selectores

- **Selector de clase**: en ocasiones no es suficiente con los selectores vistos hasta ahora para aplicar reglas a ciertos elementos, por lo que se hace necesario un selector más específico. En estos casos hacemos uso del atributo *class* de HTML, que en combinación con CSS permite indicar directamente a que elementos aplicar una regla.

```
<!DOCTYPE html>

<html>
    <head>
        <style>
            p {color: blue;}
            .destacado {color: red;}
        </style>
    </head>
    <body>
        <p class="destacado">Lorem ipsum</p>
        <p>dolor sit amet</p>
    <body>
</html>
```



Selectores

- En el código CSS indicamos el selector de clase con un punto (.):

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

- La regla anterior se aplicará a todos los elementos que tengan el atributo class="destacado" en su etiqueta HTML, por ejemplo:

```
<p class="destacado">Texto</p>
<h1 class="destacado">Texto</h1>
```

- La principal ventaja de usar clases es poder aplicar los mismos estilos a elementos diferentes de la página.

Selectores

- Podemos filtrar aún más usando las clases en conjunción con otros selectores:

```
p.destacado {}
```

- Afectaría a los elementos <p> de clase destacado, pero no a otros elementos con la clase destacado (por ejemplo, no se aplicaría a <a class="destacado">).
- De esta forma podemos seleccionar con gran precisión los elementos sobre los que vamos a aplicar los estilos.
- No obstante, no es buena idea abusar de las clases: complican la separación entre estructura y apariencia. Siempre que sea posible usar un selector convencional no usaremos una clase.

Selectores

- Es posible aplicar los estilos de más de una clase a un mismo elemento HTML, para ello usaremos la siguiente sintaxis:

```
<p class="clase1 clase2 clase3"> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
```

```
.clase1 {color: white;}
.clase2 {background-color: blue;}
.clase3 {border: 3px solid red;}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Selectores

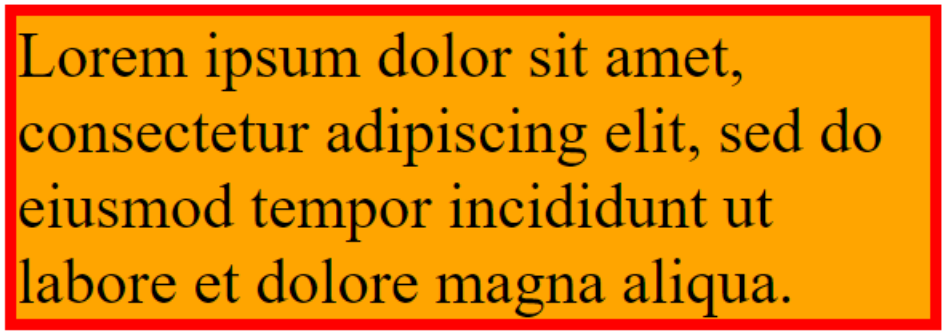
- Además, si usamos múltiples clases en un elemento HTML podemos usar un selector aún más específico. Si tomamos el ejemplo anterior y añadimos otra regla:

```
<p class=“clase1 clase2 clase3”> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
```

```
.clase1 {color: white;}
.clase2 {background-color: blue;}
.clase3 {border: 3px solid red;}
.clase1.clase2 {color: black; background-color: orange;}
```

Selectores

- Se aplicará la cuarta regla en lugar de la primera y la segunda, ya que la cuarta regla se aplica a los elementos que tienen el atributo clase1 y clase2 (y por tanto **es más específica**).



Selectores

- **Selector de ID**: Cuando necesitamos aplicar CSS a un único elemento de la página podemos usar un selector de clase para ese elemento o bien usar un selector más eficiente para estos casos: el selector de ID.
- El selector ID permite aplicar estilos a un elemento de la página por medio de su atributo id. Puesto que el atributo id no se puede repetir más de una vez en una página resulta más eficiente que crear una clase para un único elemento.
- La sintaxis en CSS es similar a la de los selectores de clase, cambiando el punto por el símbolo almohadilla (#).

Selectores

- **Selector de ID**: Cuando necesitamos aplicar CSS a un único elemento de la página podemos usar un selector de clase para ese elemento o bien usar un selector más eficiente para estos casos: el selector de ID.
- El selector ID permite aplicar estilos a un elemento de la página por medio de su atributo id. Puesto que el atributo id no se puede repetir más de una vez en una página resulta más eficiente que crear una clase para un único elemento.
- La sintaxis en CSS es similar a la de los selectores de clase, cambiando el punto por el símbolo almohadilla (#).

Selectores

```
<p id="especial"> Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua.</p>
```

```
#especial {color: white; background-color: violet;}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Selectores

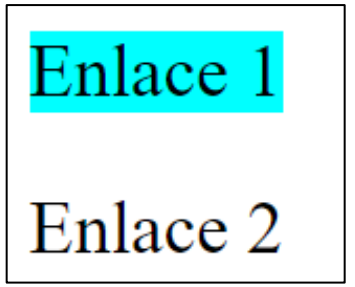
- **Selector de hijos**: cuando necesitamos delimitar los estilos al descendiente inmediato de una etiqueta, pero no a los siguientes descendientes usamos el selector de hijos, cuya sintaxis es la siguiente:

```
selector_padre > selector_hijo {…}
```

De esta forma aplicamos el estilo al hijo directo de otro selector.

```
<body>
    <p> <a> Enlace 1 </a> </p>
    <p> <span> <a> Enlace 2 </a> </span> </p>
</body>
```

```
p > a {background-color: cyan;}
```



Selectores

- **Selector adyacente**: el selector adyacente se usa para aplicar estilos a un elemento HTML que se encuentra inmediatamente a continuación de otro elemento. La sintaxis es la siguiente:

```
elemento1 + elemento2 {…}
```

  - Esta regla CSS se aplicará al elemento2 que venga después del cierre de la etiqueta de elemento1.

Selectores

- **Selector hermanos menores**: el selector de hermanos menores se usa para aplicar estilos a elementos HTML que se encuentran a continuación de otro elemento (en su mismo nivel, esto es, siendo hijos directos del mismo elemento padre). La sintaxis es la siguiente:

```
elemento1 ~ elemento2 {…}
```

  - Esta regla CSS se aplicará a los elemento2 que vengan después del cierre de la etiqueta de elemento1.

Selectores

- **Selector de atributos**: Existe una categoría de selectores que permite seleccionar elementos HTML en función de sus atributos y los valores asignados a éstos. Estos selectores de atributos son los siguientes:

    1. Selector[atributo]: permite seleccionar los elementos HTML que tienen definido ese atributo (independientemente del valor que le asignemos).
    2. Selector[atributo=valor]: permite seleccionar los elementos HTML que tienen definido ese atributo y el valor coincide con el que especificamos en el selector.

Selectores

3. Selector[atributo~=valor]: permite seleccionar los elementos HTML que tienen definido ese atributo y al menos uno de sus valores coincide con el que especificamos en el selector.
4. Selector[atributo|=valor]: permite seleccionar los elementos HTML que tienen definido ese atributo y el valor es una serie de caracteres separados por guiones, pero que empieza por los caracteres que definimos en la regla. En la práctica, este selector solo se usa con los atributos de tipo lang.

Selectores

- Si combinamos los selectores de atributos anteriores con *expresiones regulares* podemos crear tres nuevos tipos de selectores útiles:
  - Selector[atributo^=val]: nos permite seleccionar todos los elementos HTML que tengan definido ese atributo y cuyo valor comience por “val”.
  - Selector[atributo$=val]: nos permite seleccionar todos los elementos HTML que tengan definido ese atributo y cuyo valor termine en “val”.
  - Selector[atributo*=val]: nos permite seleccionar todos los elementos HTML que tengan definido ese atributo y cuyo valor contenga la cadena de texto “val”.

Herencia

- Hay determinadas propiedades CSS que se heredan de forma automática. Por ejemplo, si tenemos el siguiente código vemos que el color rojo se aplica a todos los elementos descendientes de <body>, puesto que la propiedad color es de las que se hereda automáticamente.

```
<!DOCTYPE html>

<html>
    <head>
        <title>
            Ejemplo de herencia de estilos
        </title>
        <style type="text/css">
            body { color: red; }
        </style>
    </head>
    <body>
        <h1>Título de la página</h1>
        <p>Esto es un elemento &lt;p&gt;,
        que se usa para crear un párrafo
        de texto.</p>
    </body>
</html>
```

Herencia

- Puesto que la herencia de estilos se aplica automáticamente deberemos especificar los estilos de los elementos descendientes si no queremos que apliquen los estilos heredados. Si retomamos el ejemplo anterior:

```
<!DOCTYPE html>

<html>
    <head>
        <title>
            Ejemplo de herencia de estilos
        </title>
        <style type="text/css">
            body { color: red; }
            h1 {color: blue;}
            p {color: black;}
        </style>
    </head>
    <body>
        <h1>Título de la página</h1>
        <p>Esto es un elemento &lt;p&gt;, que se
        usa para crear un párrafo de texto.</p>
    </body>
</html>
```

Colisión de estilos

- En las dos diapositivas anteriores hemos visto que a un elemento se le pueden aplicar estilos desde diferentes reglas con diferentes selectores, pero en el caso de que dos reglas le apliquen la misma propiedad se tiene que decantar por una u otra.
- El mecanismo que sigue CSS para decidir que estilo aplicar en caso de “conflicto” es el siguiente:
  - Cuanto más específico es el selector, más prioridad tendrá la regla.
  - A igual especificidad, se selecciona la última regla indicada.

Colisión de estilos

En el siguiente ejemplo, ¿qué regla prevalece?

```
* {color: red;}
p {color: blue;}
p#especial {color:green;}
p.destacado {color: yellow;}
```

Colisión de estilos

- En el ejemplo anterior, la regla que prevalece es

```
p#especial {color:green;}
```

* {color: red;} Se aplica a todos los elementos de la página: es muy poco específico.

p {color: blue;} Se aplica a todos los párrafos de la página: sigue siendo muy genérico.

p.destacado {color: yellow;} Se aplica a todos los párrafos que tienen la clase destacado.

p#especial {color:green;} Se aplica únicamente al párrafo que tiene el identificador #especial. Es, por tanto, el selector más específico y el que se aplicará.

Color en CSS

- En CSS podemos indicar el color de varias maneras:
  - Palabras clave: (black, white, blue, red, yellow, orange, green, pink, violet, etc).
  - Códigos RGB: {color: rgb(255,255,255);}
  - Códigos RGB en hexadecimal: {color: #ff00ff;}
  - Códigos RGBA: {color: rgba(255,255,255,0.5);}
  - Códigos HSL: {color: hsl(9, 100%, 64%);}
  - Códigos HSLA: {color: hsla(9, 100%, 64%, 0.5);}

https://www.w3schools.com/colors/colors_picker.asp

Enlaces (<a>)

- El estilo de los enlaces puede ser modificado fácilmente con cualquier propiedad CSS:

a {color: green;}

Además, los enlaces pueden estar en cuatro estados diferentes, y por tanto podemos definir estilos diferentes para cada uno de estos estados:

a:link {…}: enlace normal, sin visitar.
a:visited {…}: enlace que ya hemos visitado.
a:hover {…}: enlace que tiene el puntero del ratón encima.
a:active {…}: enlace en el momento que hacemos clic sobre él.

Una propiedad útil para quitar el subrayado que tienen por defecto los enlaces es {text-decoration:none;}.

Unidades de medida absolutas en CSS

Unidad	Descripción
cm	Centímetros
mm	Milímetros
in	Pulgadas (2.54cm)
px	Píxeles (1px = 1/96 pulgadas)
pt	Puntos (1pt = 1/72 pulgadas)
pc	Picas (1pc = 12 puntos)

Unidades de medida relativas en CSS

Unidad	Descripción
em	Relativo al tamaño de la fuente actual (2em significa 2 veces el tamaño de la fuente actual)
ex	Relativo a la altura de la fuente actual (prácticamente no usado)
ch	Relativo a la anchura del “0” (cero)
rem	Relativo al tamaño de la fuente del elemento raíz
vw	Relativo al 1% de la anchura del *viewport* (parte visible del navegador)
vh	Relativo al 1% de la altura del *viewport* (parte visible del navegador)
vmin	Relativo al 1% de la dimensión más pequeña del *viewport* (tamaño de la ventana del navegador)
vmax	Relativo al 1% de la dimensión más grande del *viewport* (tamaño de la ventana del navegador)
%	Relativo al elemento padre

Propiedades CSS más usadas

Propiedad	Descripción	Ejemplo
margin	Margen externo (en las cuatro direcciones)	margin: 10px;
padding	Margen interno (en las cuatro direcciones)	padding: 20px;
color	Indica el color de un elemento	color: #4EFCC8;
background-color	Establece el color de fondo de un elemento	background-color: #E356D8;
background-image	Establece una imagen como fondo	background-image: url(“imagen.jpg")
width	Establece el ancho de un elemento	width: 60px;
height	Establece la altura de un elemento	height: 30px;

Propiedades CSS más usadas

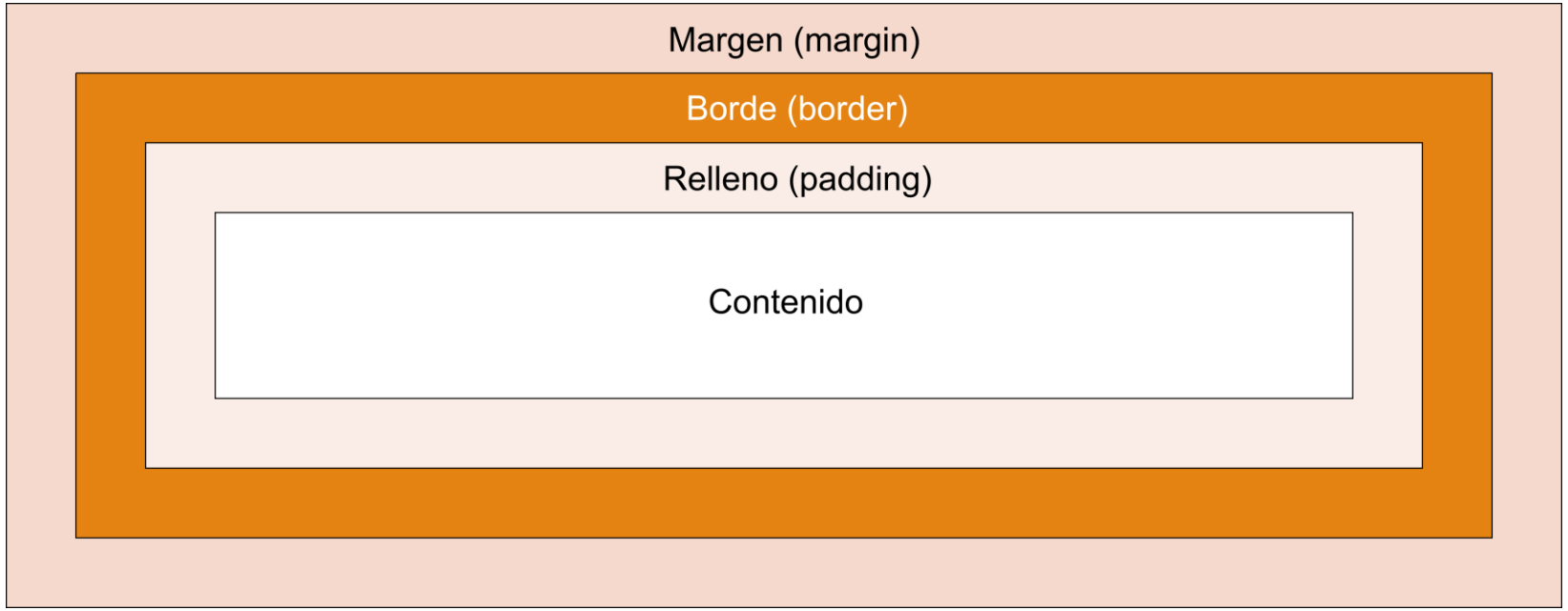
Propiedad	Descripción	Ejemplo
font-family	Define la fuente	font-family: arial;
font-size	Define el tamaño de fuente	font-size: 10pt;
font-style	Estilo de la fuente {normal, italic, oblique}	font-style: italic;
font-weight	Grosor de la fuente {normal, bold, lighter, bolder}	font-weight: bold;
line-height	Interlineado	line-height: 12px;
text-align	Alineación: left, right, center, justify.	text-align: right;

El modelo de caja

- Todo en CSS tiene una caja alrededor, visible o no, y entender este modelo de cajas es clave para poder crear diseños con CSS o para alinear elementos unos con otros.
- El modelo de caja implica que todos los elementos en CSS están representados por el **contenido**, rodeado por una serie de **marcos o cajas que lo contienen**. Las cajas se crean automáticamente con cada etiqueta HTML que escribimos.

El modelo de caja

- El modelo de cajas especifica los siguientes elementos



El modelo de caja

- No todos los elementos se comportan igual en el modelo de caja. Fundamentalmente podemos encontrar dos tipos de elementos en función de su caja:
  - Elementos en bloque (block):
    - Fuerza un salto de línea al llegar al final de la línea.
    - Ocupan todo el espacio horizontal disponible en su contenedor.
    - Se respetan las propiedades **width** y **height**.
    - padding, margin y border mantienen al resto de elementos alejados de la caja.
  - Elementos en línea (inline):
    - No se fuerza un salto de línea al llegar al final de la línea.
    - No se pueden aplicar las propiedades **width** y **height**.
    - Se aplican relleno, margen y bordes **verticales**…, pero **no mantienen alejados a otros elementos de tipo inline**.
    - Se aplican relleno, margen y bordes horizontales y mantienen alejados a otros elementos de tipo inline.

El modelo de caja

Ejemplos de elementos con caja en bloque (block):

`<div>, <h1> - <h6>, <p>, <form>, <header>, <footer>`

Ejemplos de elementos con caja en línea (inline):

`<span>, <a>, <img>`

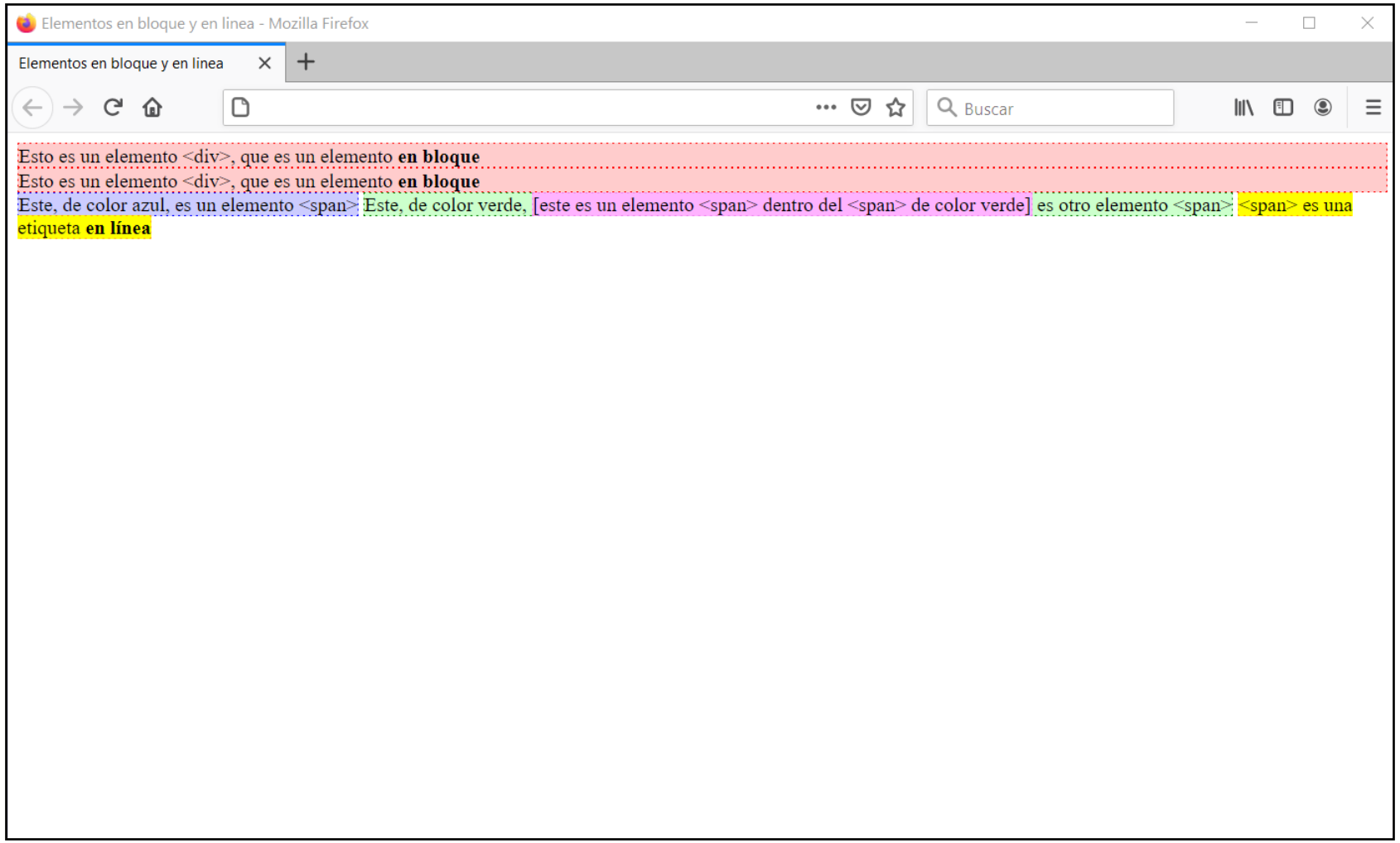
Podemos cambiar el comportamiento por defecto de los elementos con la propiedad display:

```
a {display: block;}
p {display: inline}
h1{display: inline-block}
```

Ejemplo con y

```
<!DOCTYPE html>

<html>
    <head>
        <meta charset="utf-8">
        <title>
            Elementos en bloque y en linea
        </title>
        <style>
            div {border: 1px dashed red; background-color: rgb(255, 204, 204);}
            span {border: 1px dashed blue; background-color: rgb(204, 204, 255);}
            span span {border: 1px dashed violet; background-color: rgb(255, 179, 255);}
            span + span {border: 1px dashed green; background-color: rgb(204, 255, 204);}
            span + span + span {border: 1px dashed orange; background-color: yellow;}
        </style>
    </head>
    <body>
        <div> Esto es un elemento &lt;div&gt;, que es un elemento <strong>en bloque</strong></div>
        <div> Esto es un elemento &lt;div&gt;, que es un elemento <strong>en bloque</strong></div>
        <span> Este, de color azul, es un elemento &lt;span&gt;</span> <span>Este, de color verde, <span>[este es un
        elemento &lt;span&gt; dentro del &lt;span&gt; de color verde]</span> es otro elemento &lt;span&gt;</span>
        <span>&lt;span&gt; es una etiqueta <strong>en línea</strong></span>
    </body>
</html>
```

Ejemplo con y



El modelo de caja: propiedades

Propiedad	Descripción	Ejemplo
`margin-left`	Margen izquierdo	`margin-left: 50px;`
`margin-right`	Margen derecho	`margin-right: 50px;`
`margin-top`	Margen superior	`margin-top: 50px;`
`margin-bottom`	Margen inferior	`margin-bottom: 50px;`
`padding-left`	Relleno izquierdo	`padding-left: 50px;`
`padding-right`	Relleno derecho	`padding-right: 50px;`
`padding-top`	Relleno superior	`padding-top: 50px;`
`padding-bottom`	Relleno inferior	`padding-bottom: 50px;`

El modelo de caja: propiedades

Propiedad	Descripción	Ejemplo
`border-width`	Grosor del borde	`border-width: 5px;`
`border-color`	Color del borde	`border-color: blue;`
`border-style`	Tipo de borde {none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset}	`border-style: solid;`
`border-collapse`	Usado en tablas. Permite fusionar los bordes de la tabla y las celdas	`border-collapse: collapse;`

Propiedades para alineamiento de texto

Existen diferentes propiedades CSS de aplicación al texto plano. Alguna de ellas ya las hemos visto (por ejemplo, la propiedad color).

Algo importante a tener en cuenta cuando realizamos el posicionamiento de los elementos de la página es el posicionamiento del texto. Al igual que en los procesadores de texto de las *suites* ofimáticas, podemos establecer que el texto se muestre alineado a la izquierda (comportamiento por defecto), a la derecha, centrado o justificado. Para establecer este comportamiento hacemos uso de la propiedad text-align:

```
p {text-align: center;}

a {text-align: left;}

h1 {text-align: right;}
```