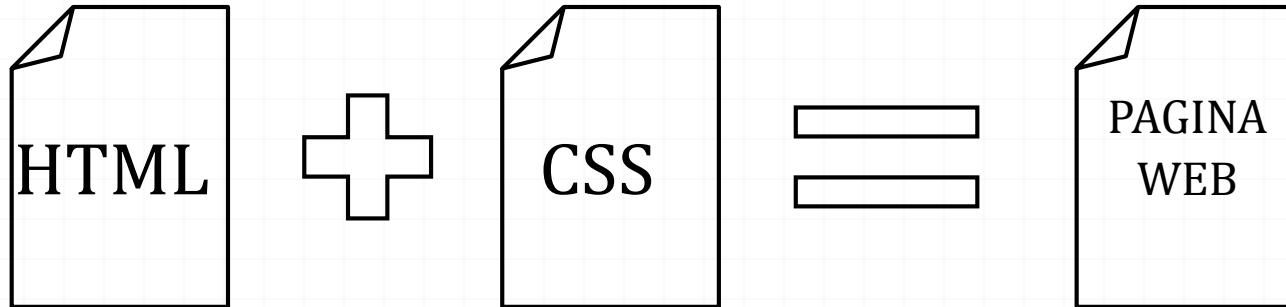


Hojas de estilo CSS

¿Qué es CSS?

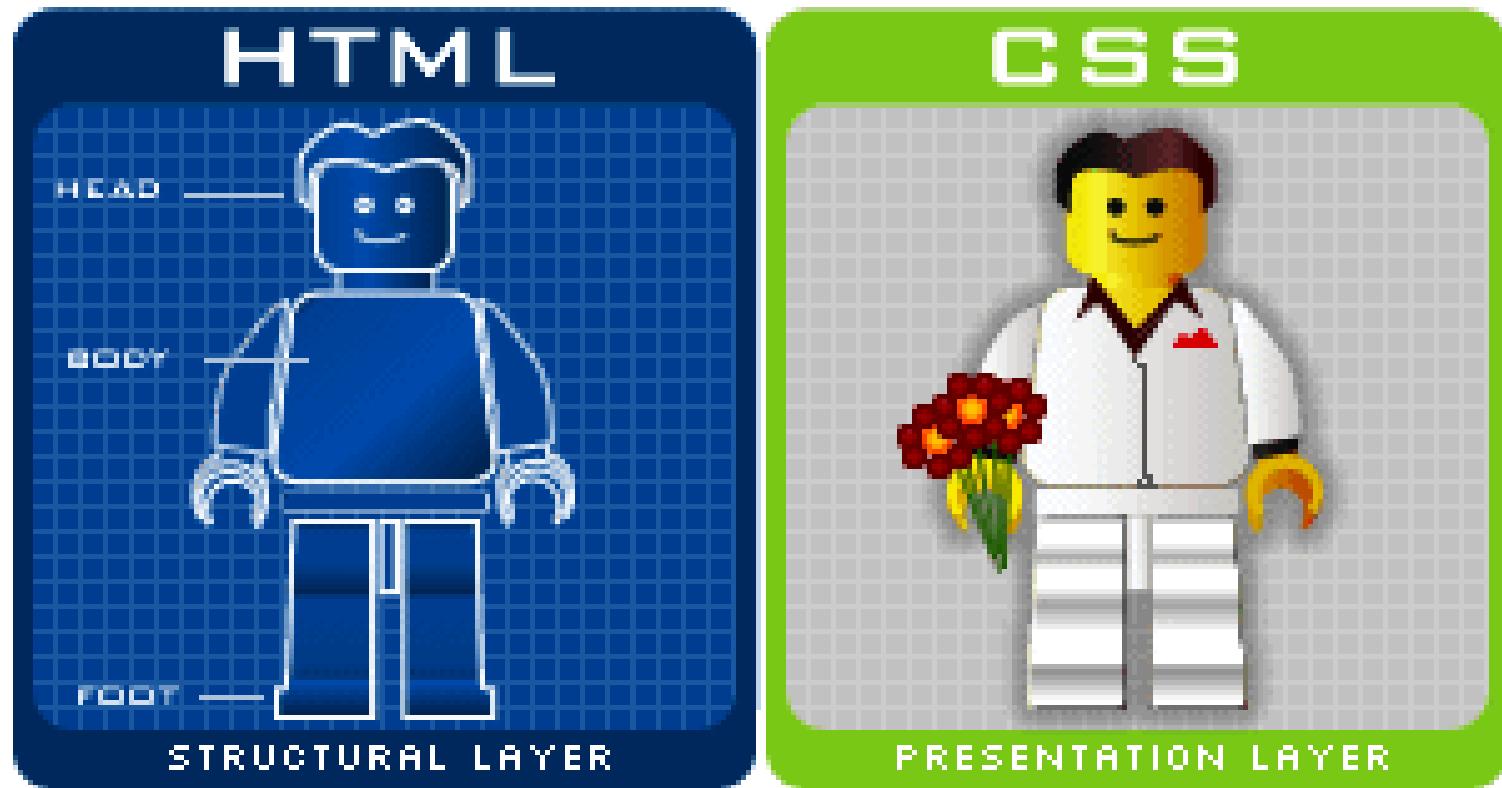
CSS = Cascading Style Sheet, Hojas de Estilo en Cascada

- Conjunto de reglas que determinan cómo se mostrarán los elementos incluidos en un documento HTML.
- Originalmente, los documentos HTML incluían el contenido que se debía de mostrar en las diferentes páginas web (texto, imágenes,...) así como información sobre el formato que debía de tener ese contenido (tipos de letra, tamaños, colores, etc.).
- Posteriormente, la aparición de CSS hizo posible separar los contenidos de la página web del aspecto que deben presentar esos contenidos.



HTML vs CSS

- HTML define los elementos que forman parte del documento
- CSS define cómo deben presentarse esos elementos



Tipos de hojas de estilos

- **Estáticas.** Solemos decir que una hoja de estilo es estática cuando ésta no varía. Es decir, el servidor enviará la misma hoja de estilo a todos los clientes (aunque habrá ciertos mecanismos que permitan adaptar dicha hoja de estilos al dispositivo desde el que se está visualizando la página).
- **Dinámicas.** Las hojas de estilo dinámicas son generadas por el servidor y en cada caso devolverá una hoja de estilo diferente (esto necesitará de la combinación de peticiones PHP y HTML).

Las hojas de estilo más habituales son las estáticas.

Tipos de estilos

- Estilos en línea
- Hojas de estilo internas
- Hojas de estilo externas

Estilos en línea

Atributo style

- Para aplicar estilos en línea debemos incluir el atributo style en la etiqueta HTML a la cual se quiere asociar el estilo.
- La sintaxis de este atributo es el siguiente:

```
<etiqueta_HTML style="propiedad1: valor; ...; propiedadx:valor;">
```

- etiqueta_HTML puede consistir en múltiples variaciones: <p>, <body>, <table>, <td>, etc.
- propiedadx es la propiedad CSS que queremos asignar.
- valor es el valor que queremos asignar a la propiedad.

```
<body style="background-color:lightgrey;">  
<h1 style="font-family:verdana;">This is a heading</h1>  
<p style="font-size:160%;">This is a paragraph.</p>
```

Principales propiedades de style

Propiedad	Resultado	Valores posibles (separados por comas)	Comentarios
font-family	Cambia el tipo de letra	verdana, arial, "Times New Roman"	Esta propiedad tiene muchas variaciones
font-size	Cambiar el tamaño de letra	10px, 80%	Puede ser en píxeles o en %
Color	Cambiar el color de letra	red, yellow, green, blue, grey, purple, #FFFFFF, #FFE022	Los colores pueden escribirse por su nombre en inglés o por su notación hexadecimal precedida de una almohadilla (#) o su valor rgb
background-color	Cambiar el color de fondo	red, yellow, green, blue, grey, purple, #FFFFFF, #FFE022, rgb(71, 98, 176);	Los colores pueden escribirse por su nombre en inglés o por su notación hexadecimal precedida de una almohadilla (#) o por su código rgb
text-align	Alinear el texto o algún otro elemento	left, center, right, justify	En el orden respectivo, los valores anteriores alinean: a la izquierda, al centro, a la derecha, alineación justificada

Propiedad	Resultado	Valores posibles (separados por comas)	Comentarios
width	Especificar el ancho de un elemento	100px, 60%	Con píxeles se obtiene un tamaño fijo, sin importar el tamaño de la ventana del navegador. Con porcentaje se obtiene un tamaño variable que depende del tamaño de la ventana del navegador.
height	Especificar el alto de un elemento	100px, 60%	Con píxeles se obtiene un tamaño fijo, sin importar el tamaño de la ventana del navegador. Con porcentaje se obtiene un tamaño variable que depende del tamaño de la ventana del navegador.
font-weight	Poner en negrita el texto	bold	También pueden usarse etiquetas para casos rápidos
font-style	Poner en cursiva un texto	italic	También pueden usarse etiquetas para casos rápidos
text-decoration	Subrayar un texto	underline	También pueden usarse etiquetas para casos rápidos



Manos a la obra

- Dibujar una línea horizontal de color verde, de 2px de grosor, de la mitad del ancho de la ventana y alineada a la izquierda:

```
<hr style="border-color:green; background-color: green;  
height:2px; width:50%; text-align:left;" />
```

- Escribir un encabezado de primer nivel de color amarillo con fondo gris y centrado.

```
<h1 style="color:yellow; background-color:grey; text-  
align: center;">Encabezado 1</h1>
```

- Darle color de fondo azul a toda la página

```
<body style="background-color:blue;">
```

- Darle color de fondo azul y texto blanco a toda la página:

```
<body style="background-color:blue;color:white;">
```



Manos a la obra

- Sobre el documento de texto proporcionado hacer los cambios siguientes:
- Color de fondo para toda la página: #FFFF99
- Encabezado de primer nivel: Verdana, rojo, 18 pt y negrita
- Resto de párrafos: Letra Arial, tamaño 12 pt.

Inconvenientes de las hojas de estilo inline

- Imposibilidad de reutilizar estilos ya definidos. Por ejemplo, si queremos aplicar el mismo estilo a varios párrafos, tendremos que repetir el código con el atributo style para cada párrafo.
- Además, si más adelante decidimos cambiar alguno de los formatos aplicados, tendremos que modificar cada atributo style.
- Es por ello que resulta más conveniente utilizar hojas de estilo internas o externas.



Hojas de estilos
internas

Definición

- 0 Son aquellas que están integradas en el mismo documento en el cual se emplearán.

Etiqueta <style>

- En las hojas de estilo internas, en lugar de utilizar el atributo style, se utiliza las etiquetas <style> ... </style> entre las etiquetas <head> de apertura y cierre.

```
<style>  
    ...  
    Eiqueta_HTML{  
        propiedad1:valor;  
        ...;  
        propiedadx:valor;  
    }  
    ...  
</style>
```

Se puede usar <style type = "text/css"> pero en html5 no es necesario

- Type. Informa al navegador de que las sentencias que se encuentran entre las etiquetas <type> de apertura y </type> de cierre son sentencias CSS

- Etiqueta_HTML {propiedad1:valor; ...; propiedadx:valor}
 - Observar que el valor de las propiedades se asigna utilizando dos puntos (:) en lugar del signo igual
 - Si para una misma etiqueta se definen varias propiedades éstas se separarán por punto y coma (,)
 - Si queremos definir un mismo estilo para varias etiquetas, separaremos sus nombres por comas (,) y definimos el estilo una única vez.
- Entre las etiquetas <style> de apertura y cierre podremos incluir comentarios que ayuden a interpretar los estilos que se están definiendo. Dado que estamos entre las etiquetas <style>, la sintaxis adecuada para incluir el comentario no será la sintaxis establecida por HTML para los comentarios, sino la sintaxis CSS para los comentarios y que es la siguiente: /* comentario */

Consejos a la hora de crear una hoja de estilos

- *Llaves.* Generalmente, la llave de apertura se escribe en la misma línea del selector y la llave de cierre se escribirá en una nueva línea.

```
p {  
    font-family: verdana;  
}
```

- *Tabulaciones.* Para facilitar la lectura del código, las líneas donde se teclean las propiedades que se definen para un selector suelen empezar algo más a la derecha.

```
p {  
    font-family: verdana;  
}
```

- *Comentarios.* Es conveniente incluir comentarios que ayuden a interpretar el código. En CSS los comentarios se incluyen entre los signos /* comentario */

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

O Agrupación de reglas. Cuando trabajamos con archivos CSS complejos es muy habitual que se apliquen diferentes estilos a un mismo elemento. En este caso, podemos agrupar todas las declaraciones en una única sentencia:

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

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

O Igualmente, si un mismo estilo se aplica a diferentes elementos se pueden agrupar también en una misma sentencia.

```
h1 { color: red; }  
...  
h2 { color: red; }  
...  
h3 { color: red; }
```

```
h1, h2, h3 {  
    color: red;  
}
```



Manos a la obra

Sobre el documento de Mafalda anterior, definir los estilos en línea siguientes:

```
<head>
<style>
    body {background-color:silver;}
    h1 {
        font-family:"Times New Roman";
        color:red;
        font-weight:bold;
    }
    p {
        font-family: Tahoma;
        color:blue;
        font-size: 12px;
    }
</style>
</head>
```

Hojas de estilos externas

Definición

- Conjunto de estilos que se encuentran almacenados en un archivo diferente al archivo donde se almacena el código HTML de la página Web.
- Esta es forma habitual de definir estilos ya que nos permite separar por completo la definición de la estructura de la página de las reglas de formato que se van a aplicar a esa página.

Crear una hoja de estilos externa

- Abrir un editor de texto plano
- Comenzar a escribir las sentencias CSS
- Guardar el documento con la extensión .css
- Enlazar el documento html con el archivo .css

```
<link rel="stylesheet" type="text/css" href="url" media="medio" />
```

- **rel**: indica el tipo de relación que existe entre 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. Cuando enlazamos nuestro documento HTML con un archivo CSS en tipo siempre especificaremos **text/css**
- **href**: indica la URL del archivo CSS que contiene los estilos. Esta url puede ser absoluta o relativa.
- **media**: indica el medio para el que se va a aplicar los estilos. Este es un valor estandarizado y puede tomar valores como **screen** (este es el valor por defecto y se utiliza cuando la hoja de estilos se va a aplicar cuando el documento HTML se está presentando en pantalla), **print** (para salida a una impresora), **all** (para cualquier dispositivo de salida).

- En HTML5 para enlazar nuestro documento HTML con un archivo de hoja de estilos pensado para presentar el contenido por pantalla bastaría con la siguiente sentencia

```
<link rel="stylesheet" href="url" />
```

- Un mismo documento HTML puede contener varios enlaces a diferentes archivos CSS mediante varias etiquetas <link>.



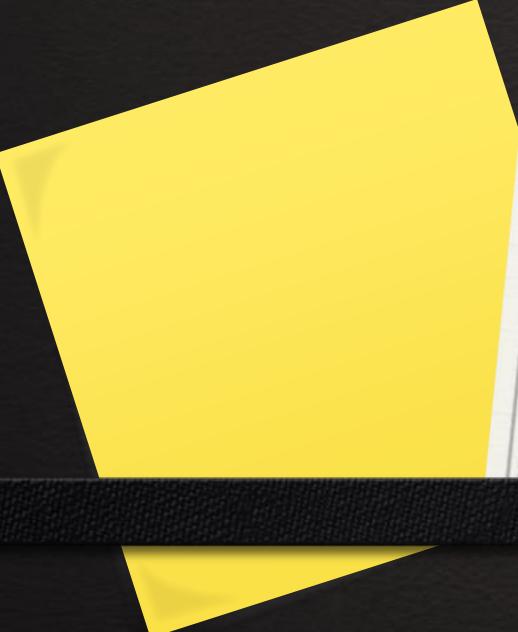
Manos a la obra

Sobre el documento Mafalda, definir los siguientes estilos utilizando una hoja de estilos externa

```
body {background-color:black; }

h1 {
    font-family:Arial;
    color:white;
    font-weight:bold;
}

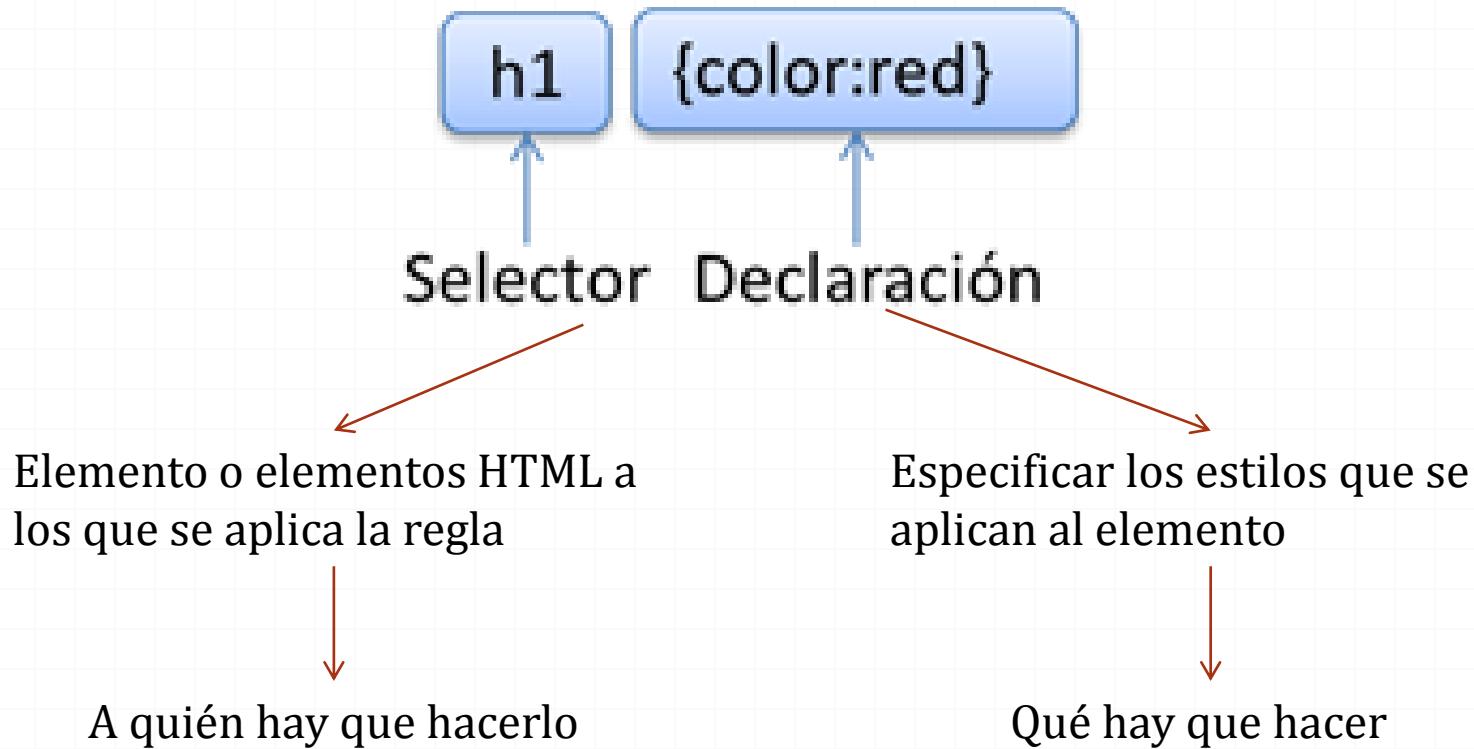
p {
    font-family: Verdana;
    color:white;
    font-size: 12px;
}
```



Definición de las
reglas de estilos

Reglas de estilos

- Regla = Cada uno de los estilos que aparecen en una hoja de estilos CSS
- Partes de una regla:



Partes de una declaración



Propiedad

Valor

Característica que se modifica del elemento (fuente, color, tamaño, etc.)

Permite establecer el nuevo valor de la característica que estamos modificando

- Un archivo CSS puede contener un número ilimitado de reglas CSS, cada regla se puede aplicar a varios selectores diferentes y cada declaración puede incluir tantos pares propiedad/valor como se deseé.

Tipos de selectores

Selector universal

- Se utiliza para referirnos a todos los elementos de la página
- Se indica mediante el asterisco (*)
- No se suele utilizar habitualmente ya que es difícil que un mismo estilo se aplique a todos los elementos de una página
- A veces se utiliza para borrar todos los estilos que pudieran tener cualquier elemento de la página para asegurarnos de que no se están arrastrando estilos
- El ejemplo siguiente elimina el margen de todos los elementos HTML:

```
* {  
    margin: 0;  
}
```



Manos a la obra

Sobre el documento de Mafalda en blanco añadir la estructura HTML adecuada. Luego ver la presentación de la página.

Luego crear un documento estilos.css con la siguiente sentencia:

```
* {  
    margin: 0;  
}
```

Enlazar el documento de Mafalda con esta hoja de estilos y comprobar cómo afecta a la página este estilo definido.

MAFALDA

Mafalda es el nombre de una tira de prensa argentina desarrollada por el historietista Quino desde 1964 a 1973, protagonizada por la niña homónima, "espejo de la clase media latinoamericana y de la juventud progresista" que se muestra preocupada por la humanidad y la paz mundial, y se rebela contra el mundo legado por sus mayores.



MAFALDA

Mafalda es el nombre de una tira de prensa argentina desarrollada por el historietista Quino desde 1964 a 1973, protagonizada por la niña homónima, "espejo de la clase media latinoamericana y de la juventud progresista" que se muestra preocupada por la humanidad y la paz mundial, y se rebela contra el mundo legado por sus mayores.

- Si se quiere aplicar los mismos estilos a varias etiquetas diferentes, se pueden encadenar los selectores. Para ello bastará con incluir todos los selectores separados por comas (,)

Sin encadenar selectores

```
h1 {  
    color: #8904B1;  
    font-family: Arial;  
}  
  
h2 {  
    color: #8904B1;  
    font-family: Arial;  
}
```

Encadenando selectores

```
h1, h2 {  
    color: #8904B1;  
    font-family: Arial;  
}
```

- Cuando trabajamos con hojas de estilo complejas, habitualmente se suelen agrupar las propiedades comunes de varios elementos en una única regla CSS y posteriormente se definen las propiedades específicas de cada uno de esos elementos.

```
h1, h2, h3 {  
    color: #8904B1;  
    font-family: Arial;  
}  
  
h1 { font-size: 2em; }  
h2 { font-size: 1.5em; }  
h3 { font-size: 1.2em; }
```



Modificar archivo
Mafalda como en
ejercicio 2

Selector descendente

- Estos selectores permiten aplicar estilos a un elemento, cuando éste se encuentre dentro de otro.
- Su sintaxis es la siguiente:

```
selector1 selector2 selector3 ... selectorN {  
    propiedad1:valor1; ... propiedad_n:valor_n  
}
```



Modificar archivo
Mafalda como en
ejercicio 3

- Los selectores descendentes siempre están formados por dos o más selectores separados entre sí por espacios en blanco. selectorN indica el elemento sobre el que se aplican los estilos y todos los selectores anteriores indican el lugar en el que se debe encontrar ese elemento.
- En el ejemplo siguiente, se aplicará una línea de subrayado a los elementos de tipo `` que se encuentren dentro de elementos de tipo `<a>` que, a su vez, se encuentren dentro de elementos de tipo `<p>`.

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

- Deberemos tener en cuenta que la única condición es que un elemento debe estar dentro de otro elemento, sin importar el nivel de profundidad en el que se encuentre.

- No es lo mismo utilizar el selector descendente que combinar selectores:

```
/* El estilo se aplica a todos los elementos "p", "a" y "em" */  
p, a, em { text-decoration: underline; }  
/* El estilo se aplica solo a los elementos "em" que se encuentran dentro de  
"a" que a su vez se encuentra dentro de "p" */  
p a em { text-decoration: underline; }
```

- En ocasiones, se delimita el alcance del selector descendente combinándolo con el selector universal

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

Y el siguiente código HTML:

```
<p>  
    <a href="#">Enlace_1</a>  
</p>  
<p>  
    <span>  
        <a href="#">Enlace_2</a>  
    </span>  
</p>
```

- Sólo se visualizará en rojo Enlace_2 ya que en el código CSS se le está indicando que muestre en rojo aquellos elementos `<a>` que se encuentren dentro de cualquier elemento que, a su vez, esté dentro de un elemento `<p>`. Enlace_1 el elemento `<a>` se encuentra directamente dentro de un elemento `<p>` y no se cumple la condición.

Selectores de clase

- 0 Supongamos que en nuestro documento tenemos el siguiente código:



```
<body>
```

```
    <p>Alguna de las novelas más famosas de Cervantes  
son:</p>
```

```
    <p>- El ingenioso hidalgo Don Quijote de la  
Mancha</p>
```

```
    <p>- Viaje al Parnaso</p>
```

```
    <p>- Los trabajos de Persiles y Segismunda</p>
```

```
</body>
```

- 0 Queremos aplicar un estilo CSS que destaque la línea que contiene el título del Quijote de la Mancha ya que es la obra más famosa de Cervantes ¿Cómo podríamos hacerlo?



No podemos utilizar el selector universal (*) porque seleccionaría todos los elementos de la página.



Tampoco podemos utilizar el selector (p) porque seleccionaría todos los párrafos.



Tampoco podemos utilizar el selector (body p) porque todos los elementos de tipo <p> se encuentran dentro del <body> y les estaríamos dando el formato a todos ellos.

0 La solución más sencilla a este tipo de problemas, es la utilización del atributo class de HTML sobre ese elemento:

```
<body>
  <p>Alguna de las novelas más famosas de Cervantes son:</p>
  <p class="destacar">
    - El ingenioso hidalgo Don Quijote de la Mancha
  </p>
  <p>- Viaje al Parnaso </p>
  <p>- Los trabajos de Persiles y Segismunda</p>
</body>
```

0 A continuación, en el archivo CSS se especifica una regla a la que se llamará destacar. Para identificar perfectamente el elemento y que no haya confusión con otros tipos de selectores, se antepondrá un punto(.) al nombre de la regla:

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

0 Este tipo de selectores se llaman selectores de clase.

0 Podemos utilizar el mismo valor en el atributo class para varios elementos diferentes en un mismo documento HTML. Esto hace que sea posible reutilizar los mismos estilos para varios elementos diferentes.

```
<body>  
  <p>Alguna de las novelas más famosas de Cervantes son:</p>  
  <p class="destacar">  
    - El ingenioso hidalgo Don Quijote de la Mancha  
  </p>  
  <p>- Viaje al Parnaso </p>  
  <p class="destacar">- Los trabajos de Persiles y Segismunda</p>  
</body>
```



- A veces, necesitaremos restringir el alcance del selector de clase. Por ejemplo, en el código siguiente sólo queremos poner en rojo la línea correspondiente al título del Quijote y al de Persiles y Segismunda.

```
<body>
  <p>Alguna de las novelas más famosas de Cervantes son:</p>
  <p class="destacar">
    - El ingenioso hidalgo Don Quijote de la Mancha
  </p>
  <p>
    - Viaje al <a href="https://es.wikipedia.org/wiki/Parnaso"
      class="destacar">Parnaso</a>
  </p>
  <p class="destacar">
    - Los trabajos de Persiles y Segismunda
  </p>
</body>
```



- La solución en este caso, sería utilizar una combinación del selector de tipo y el selector de clase.

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

- En este caso, estamos indicando que se deben mostrar en rojo aquellos elementos de tipo `<p>` que disponga de un atributo `class` con el valor `destacar`.
- Podemos aplicar los estilos de varias clases a un mismo elemento. Para ello, escribiremos los diferentes valores del atributo `class` separados por espacios.
`<p class="especial destacar error">Párrafo de texto...</p>`
- En el ejemplo anterior, al párrafo se le aplicarían los estilos definidos en las reglas `.especial`, `.destacar` y `.error`.

0 No debemos confundir el selector de clase con los selectores vistos en los apartados 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 */  
p, .aviso { ... }
```

Selectores id

- En ocasiones, queremos aplicar estilos CSS a un único elemento de la página.
 - Aunque podríamos utilizar un selector de clase, en este caso es más eficiente utilizar el selector ID.
- Su sintaxis es muy parecida a la del selector de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

- Estilo CSS

```
#aviso { color: red; }
```



Ejercicio 7_1

- Contenido HTML

```
<p>Primer párrafo</p>
<p id="aviso">Segundo párrafo</p>
<p>Tercer párrafo</p>
```

- La principal diferencia entre el selector id y el selector de clase es que el valor del atributo id debe ser único en un documento HTML.
 - Es decir, dos elementos diferentes no pueden tener el mismo valor de id. Sin embargo, muchos elementos HTML diferentes pueden tener el mismo valor para su atributo class.
- En los selectores id también podemos restringir el alcance del selector mediante la combinación con otros selectores.

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

- En el ejemplo anterior sólo se aplica la regla CSS al elemento de tipo <p> que tenga un atributo id igual a aviso.
- Aunque nos pueda parecer que no tiene sentido ya que acabamos de decir que el id es único y si hay un p que tiene ese id no debe haber otro elemento en el documento HTML que tenga ese id, puede ocurrir que varios documentos HTML compartan el mismo archivo de estilos css y sí puede ocurrir que en diferentes documentos exista el mismo id asignado a diferentes elementos html.



Ejercicio 7_1 y 7_2 y comprobar cómo se ven en el navegador
Luego modificar estilos css y poner p#aviso y comprobar que
ahora no se ve en rojo el Encabezado de 7_2

0 No debe confundirse el selector de ID con los selectores anteriores:

```
/* Todos los elementos de tipo "p" con atributo id="aviso" */  
p#aviso { ... }
```

```
/* Todos los elementos con atributo id="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 id="aviso" de la página */  
p, #aviso { ... }
```



Manos a la obra

- En la carpeta ptm1 proporcionada tenemos dos archivos:
 - El primero se llama ptm1.html y contiene el código html de una página
 - El segundo se llama estilos.css y contiene el código css con los estilos que se deben de aplicar a la página anterior, pero faltan los selectores.
- Teniendo en cuenta los comentarios del propio archivos estilos.css, deben añadirse los estilos necesarios para que la página se vea con el formato adecuado.



Manos a la obra

- Abrir el archivo ptm2
- Como puedes ver es un archivo html con un enlace a un archivo css.
- Si miramos el archivo estilos.css solamente contiene una regla para mostrar las líneas de la tabla.
- Deberemos usar la propiedad color para cambiar el color de los elementos de la página.
 - El formato de esta propiedad es `color:valor`
 - Los valores que vamos a utilizar para los colores son:
teal, red, blue, orange, purple, olive, fuchsia y green

[Etiqueta span](#)

Etiqueta span

- La etiqueta **** se utiliza para añadir estructura a una parte de un texto.
- Esta etiqueta se usa comúnmente para añadir un estilo determinado dentro de una sección del texto.

```
<span>Texto </span>
```

- Esta etiqueta admite atributos como id y class, lo que hace que podamos dar formato a una parte de un párrafo

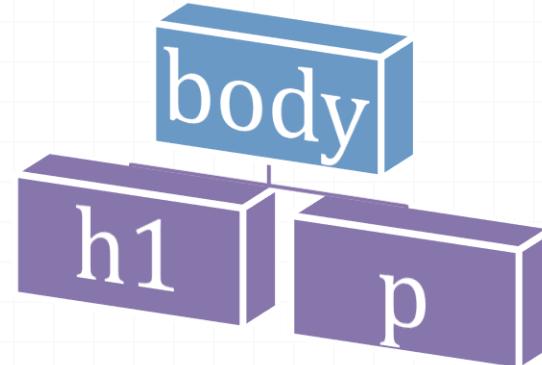
Texto con formato de párrafo `Texto a destacar ` Texto con formato de párrafo

Texto con formato de párrafo `Texto a destacar ` Texto con formato de párrafo

Herencia de estilos

- 0 Las etiquetas de un documento HTML están organizadas de manera que unas engloban a otras en una relación padre - hijo. Por ejemplo las etiquetas se encuentran dentro de la etiqueta <body> como <p>, <h1>, , etc. son etiquetas hijas y la etiqueta <body> será una etiqueta padre.
- 0 Esta organización permite una relación padre-hijo de manera que los estilos definidos para etiquetas padres serán heredados por los hijos.
- 0 El mecanismo de herencia de estilos siempre funciona a menos que un elemento hijo tenga definido su propio estilo, el cual prevalece sobre el heredado e incluso en algunos casos se complementan.

```
<!DOCTYPE html>
<html lang="es" >
<head>
    <meta charset="utf-8" />
    <title>Ejemplo de herencia de estilos</title>
    <style>
        body { color: blue; }
    </style>
</head>
<body>
    <h1>Titular de la página</h1>
    <p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```



- En el ejemplo anterior, el selector `body` solamente establece el color de la letra para el elemento `<body>`.
 - No obstante, la propiedad `color` es una de las que se heredan de forma automática, por lo que todos los elementos descendientes de `<body>` muestran ese mismo color de letra.
 - Por tanto, establecer el color de la letra en el elemento `<body>` de la página implica cambiar el color de letra de todos los elementos de la página.



0 Aunque la herencia de estilos se aplica automáticamente, se puede anular su efecto estableciendo de forma explícita otro valor para la propiedad que se hereda

```
<!DOCTYPE html>
<html lang="es" >
  <head>
    <meta charset="utf-8" />
    <title>Ejemplo de herencia de estilos</title>
    <style>
      body {font-family: Arial; color: blue; }
      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 establece en primer lugar el color y tipo de letra del elemento <body>, por lo que todos los elementos de la página se mostrarían con ese mismo color y tipo de letra. No obstante, las otras reglas CSS modifican alguno de los estilos heredados.
 - Los elementos <h1> se muestran con el tipo de letra Verdana establecido por el selector h1 y se muestran de color negro que es el valor heredado del elemento <body>.
 - Los elementos <p> de la página se muestran del color rojo establecido por el selector p y con un tipo de letra Arial heredado del elemento <body> .



Jerarquía de los estilos

Concepto de cascada

- Como ya hemos visto los estilos pueden definirse en varios sitios:
 - En un fichero externo .css que luego referenciamos a través de la etiqueta <link> o @import.
 - A través de un bloque de estilos definidos en la etiqueta <style>.
 - Directamente sobre la etiqueta HTML con el atributo style.
- En una misma página web se pueden utilizar los tres tipos de estilos lo que puede dar lugar a que algunos estilos se solapen
 - En este caso, es necesario aplicar las reglas en cascada para determinar qué estilo prevalecerá.
- Las reglas en cascada determinan que los estilos prevalecerán en orden inverso a cómo aparecen en la lista superior. Es decir, prevalecen los estilos definidos mediante el atributo style frente a los definidos con la etiqueta <style> y éstos prevalecerán sobre los estilos definidos en un fichero externo.

Colisiones de estilos

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

```
p {color: red;}  
p {color: blue;}  
<p>Texto</p>
```

- Con las dos reglas anteriores, ¿De qué color se verá el texto?
 - El mecanismo simplificado que se puede aplicar es el siguiente:
 1. Cuanto más específico sea un selector, más importancia tiene su regla asociada.
 2. A igual especificidad, se considera la última regla indicada
 - Por lo que en el ejemplo anterior el texto se vería en azul

```
p {color: red;}  
p#especial {color: green;}  
* {color: blue;}  
  
<p id="especial">Texto</p>
```

- Al elemento <p> se le aplican las tres declaraciones.
 - El selector * es el menos específico, ya que se refiere a "todos los elementos de la página".
 - El selector p es poco específico porque se refiere a "todos los párrafos de la página".
 - Por último, el selector p#especial sólo hace referencia a "el párrafo de la página cuyo atributo id sea igual a especial".
- Como el selector p#especial es el más específico, su declaración es la que se tiene en cuenta y por tanto el párrafo se muestra de color verde.

Colores CSS

Colores CSS

- 0 Los colores en CSS se pueden indicar de cualquiera de las formas ya vistas: mediante un nombre de color válido, mediante un valor RGB o mediante un valor hexadecimal.

Nombre de color

○ CSS (al igual que HTML) permite 140 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

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

Valor RGB

- Los colores se pueden especificar mediante un valor RGB que se corresponde con la fórmula: `rgb (rojo, verde, azul)`.
- Cada parámetro (rojo, verde, azul) define la intensidad del color entre 0 y 255.

Valor RGB	Color
<code>rgb(255,0,0)</code>	Red
<code>rgb(0,255,0)</code>	Green
<code>rgb(0,0,255)</code>	Blue
<code>rgb(238,128,18)</code>	Orange

```
p {color: rgb(238, 128, 18) ; }
```

- Generalmente, los tonos grises se definen mediante valores iguales para los tres colores:

Valor RGB	Color
<code>rgb(0,0,0)</code>	Black
<code>rgb(128,128,128)</code>	Grey
<code>rgb(217,217,217)</code>	Light Grey
<code>rgb(255,255,255)</code>	White

Valor hexadecimal

- Los colores en CSS también se pueden especificar con su valor hexadecimal en la forma: #RRGGBB, donde RR (rojo), GG (verde) y BB (azul) son valores hexadecimales entre 00 y FF (igual que en decimal 0-255).
- Por ejemplo, # FF0000 se muestra como rojo, porque el rojo se establece en su valor más alto (FF) y los otros se establece en el valor más bajo (00).

Valor RGB	Color
#FF0000	
#00FF00	
#0000FF	
#EE8012	

h1, h2, h3, h4, h5, h6 {color: #C00; }

- Generalmente, los tonos grises se definen mediante valores iguales para los tres colores: #000000 #808080 #FFFFFF
- El formato hexadecimal permite comprimir sus valores cuando todas sus componentes son iguales dos a dos:
 - #AAA = #AAAAAA
 - #FFF = #FFFFFF
 - #A0F = #AA00FF
 - #369 = #336699

Unidades de medida

Unidades de medida

- Muchas de las propiedades de CSS permiten indicar medidas en sus valores.
 - Para definir la altura, anchura y márgenes de los elementos
 - Para establecer el tamaño de letra del texto.
 - Etc.
- CSS permite indicar las medidas de muchas formas diferentes
 - 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).
 - Algunas propiedades permiten indicar medidas negativas, aunque habitualmente sus valores son positivos.
 - Si el valor decimal de una medida es inferior a 1, se puede omitir el 0 de la izquierda (0.5em es equivalente a .5em).
- CSS divide las unidades de medida en dos grupos: absolutas y relativas.

Unidades absolutas

- Una medida indicada mediante unidades absolutas está completamente definida, ya que su valor no depende de otro valor de referencia.
- Unidades absolutas definidas por CSS y su significado:
 - in, pulgadas ("inches", en inglés). Una pulgada equivale a 2.54 centímetros.
 - cm, centímetros.
 - mm, milímetros.
 - pt, puntos. Un punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros.
 - pc, picas. Una pica equivale a 12 puntos, es decir, unos 4.23 milímetros.

Ejemplos de utilización

```
/* El cuerpo de la página debe mostrar un margen de media  
pulgada */  
body {margin: 0.5in;}  
  
/* Los elementos <h1> deben mostrar un interlineado de 2  
centímetros */  
h1 {line-height: 2cm;}  
  
/* Las palabras de todos los párrafos deben estar separadas 4  
milímetros entre si */  
p {word-spacing: 4mm;}  
  
/* Los enlaces se deben mostrar con un tamaño de letra de 12  
puntos */  
a {font-size: 12pt}  
  
/* Los elementos <span> deben tener un tamaño de letra de 1  
pica */  
span {font-size: 1pc}
```

- La principal ventaja de las unidades absolutas es que su valor es directamente el valor que se debe utilizar, sin necesidad de realizar cálculos intermedios.
- Su principal desventaja es que son muy poco flexibles y no se adaptan fácilmente a los diferentes medios.
- De todas las unidades absolutas, la única que suele utilizarse es el punto (pt).
 - Se trata de la unidad de medida preferida para establecer el tamaño del texto en los documentos que se van a imprimir.

Unidades relativas

- Las unidades relativas, a diferencia de las absolutas, no están completamente definidas, ya que su valor siempre está referenciado respecto a otro valor. A pesar de su aparente dificultad, son las más utilizadas en el diseño web por la flexibilidad con la que se adaptan a los diferentes medios.
- Unidades de medida relativas definidas por CSS:
 - em, (no confundir con la etiqueta de HTML) relativa respecto del tamaño de letra del elemento.
 - ex, relativa respecto de la altura de la letra x ("equis minúscula") del tipo y tamaño de letra del elemento.
 - px, (píxel) relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.

- Las unidades em y ex no han sido creadas por CSS, sino que llevan décadas utilizándose en el campo de la tipografía.
- La unidad em hace referencia al tamaño en puntos de la letra que se está utilizando. Si se utiliza una tipografía de 12 puntos, 1em equivale a 12 puntos. El valor de 1ex se puede aproximar por 0.5 em.

p {margin: 1em; }

- La regla CSS anterior indica que los párrafos deben mostrar un margen de anchura igual a 1em.
 - Como se trata de una unidad de medida relativa, es necesario realizar un cálculo matemático para determinar la anchura real de ese margen.
- La unidad de medida em siempre hace referencia al tamaño de letra del elemento. Por otra parte, todos los navegadores muestran por defecto el texto de los párrafos con un tamaño de letra de 16 píxel. Por tanto, en este caso el margen de 1em equivale a un margen de anchura 16px.

```
p { font-size: 32px; margin: 1em; }
```

- El valor del margen sigue siendo el mismo en unidades relativas (1em) pero su valor real ha variado porque el tamaño de letra de los párrafos ha variado. En este caso, el margen tendrá una anchura de 32px, ya que 1em siempre equivale al tamaño de letra del elemento.
- Si se quiere reducir la anchura del margen a 16px pero manteniendo el tamaño de letra de los párrafos en 32px, se debe utilizar la siguiente regla CSS:

```
p { font-size: 32px; margin: 0.5em; }
```

- El valor 0.5em se interpreta como "la mitad del tamaño de letra del elemento", ya que se debe multiplicar por 0.5 su tamaño de letra ($32\text{px} \times 0.5 = 16\text{px}$).
- De la misma forma, si se quiere mostrar un margen de 8px de anchura, se debería utilizar el valor 0.25em, ya que $32\text{px} \times 0.25 = 8\text{px}$.

- La gran ventaja de las unidades relativas es que siempre mantienen las proporciones del diseño de la página.
 - Establecer el margen de un elemento con el valor 1em equivale a indicar que "el margen del elemento debe ser del mismo tamaño que su letra y debe cambiar proporcionalmente".
 - Si el tamaño de letra de un elemento aumenta hasta un valor enorme, su margen de 1em también será enorme.
 - Si su tamaño de letra se reduce hasta un valor diminuto, el margen de 1em también será diminuto.
 - El uso de unidades relativas permite mantener las proporciones del diseño cuando se modifica el tamaño de letra de la página.

Unidad ex

- 0 El funcionamiento de la unidad ex es idéntico a em, salvo que en este caso, su valor es aproximadamente la mitad que el de la unidad em.

píxeles

- Las medidas indicadas en píxel también se consideran relativas, ya que el aspecto de los elementos dependerá de la resolución del dispositivo en el que se visualiza la página HTML.
 - Si un elemento tiene una anchura de 400px, ocupará la mitad de una pantalla con una resolución de 800x600, pero ocupará menos de la tercera parte en una pantalla con resolución de 1440x900.

Uso de unidades absolutas y relativas conjuntamente

- Las unidades de medida se pueden mezclar en los diferentes elementos de una misma página, como en el siguiente ejemplo:

```
body {font-size: 10px;}
```

```
h1 {font-size: 2.5em;}
```

- En primer lugar, se establece un tamaño de letra base de 10 píxel para toda la página. A continuación, se asigna un tamaño de 2.5em al elemento <h1>, por lo que su tamaño de letra real será de $2.5 \times 10\text{px} = 25\text{px}$.

- Como ya vimos en la herencia de estilos, el valor de la mayoría de propiedades CSS se hereda de padres a hijos. Así por ejemplo, si se establece el tamaño de letra al elemento <body>, todos los elementos de la página tendrán el mismo tamaño de letra, salvo que indiquen otro valor.
- Sin embargo, el valor de las medidas relativas no se hereda directamente, sino que se hereda su valor real una vez calculado. El siguiente ejemplo muestra este comportamiento:

```
body { font-size: 12px; text-indent: 3em; }  
h1 { font-size: 15px }
```

- El elemento <body> define un valor para esta propiedad, pero el elemento <h1> no lo hace, por lo que heredará el valor de su elemento padre. Sin embargo, el valor heredado no es 3em, sino 36px
 - En primer lugar se calcula el valor real de 3em para el elemento <body>:
 $3em \times 12px = 36px$.
 - Una vez calculado el valor real, este es el valor que se hereda para el resto de elementos
 - Para calcular el tamaño en em partiendo de px se divide el tamaño en píxeles entre 16 (pixeles/16=em)

Porcentajes

- El porcentaje también es una unidad de medida relativa, aunque por su importancia CSS la trata de forma separada a em, ex y px.
- Un porcentaje está formado por un valor numérico seguido del símbolo % y siempre está referenciado a otra medida.

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

- Los tamaños establecidos para los elementos <h1> y <h2> mediante las reglas anteriores, son equivalentes a 2em y 1.5em respectivamente, por lo que es más habitual definirlos mediante em.

```
/*Estilos CSS*/  
div#contenido { width: 600px; }  
div.principal { width: 80%; }
```

```
<!-- Código HTML -->  
<div id="contenido">  
  <div class="principal">  
    ...  
  </div>  
</div>
```

- 0 En el ejemplo anterior, la referencia del valor 80% es la anchura de su elemento padre. Por tanto, el elemento `<div>` cuyo atributo class vale principal tiene una anchura de 80% x 600px = 480px

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.
- El documento Recomendaciones sobre técnicas CSS para la mejora de la accesibilidad de los contenidos HTML, elaborado por el organismo W3C, recomienda el uso de la unidad em para indicar el tamaño del texto y para todas las medidas que sean posibles.



Formatos de
fuente

Introducción

- 0 CSS permite aplicar estilos complejos y muy variados al texto de las páginas web.

font-family

- El tipo de letra del texto se puede indicar de dos formas diferentes:
 - Mediante el **nombre de una familia tipográfica**:
 - "Arial"
 - "Verdana"
 - "Garamond"
 - etc.
 - Mediante el **nombre genérico** de una familia tipográfica: los nombres genéricos no se refieren a ninguna fuente en concreto, sino que hacen referencia al estilo del tipo de letra.
 - Las familias genéricas definidas son
 - serif (tipo de letra similar a Times New Roman),
 - sans-serif (tipo Arial),
 - cursive (tipo Comic Sans),
 - fantasy (tipo Impact)
 - monospace (tipo Courier New)

- Los navegadores muestran el texto de las páginas web utilizando los tipos de letra instalados en el ordenador o dispositivo del propio usuario.
 - 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.
 - Como es imposible especificar todos los tipos de letras, para eso se utilizan las familias tipográficas genéricas. Cuando la propiedad font-family toma un valor igual a sans-serif, el diseñador no indica al navegador que debe utilizar la fuente Arial, sino que debe utilizar "la fuente que más se parezca a Arial de todas las que tiene instaladas el usuario".

0 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.

0 Las listas de tipos de letra más utilizadas son las siguientes:

`font-family: Arial, Helvetica, sans-serif;`

`font-family: "Times New Roman", Times, serif;`

`font-family: "Courier New", Courier, monospace;`

`font-family: Georgia, "Times New Roman", Times, serif;`

`font-family: Verdana, Arial, Helvetica, sans-serif;`

Cuando el nombre de la fuente tiene más de una palabra, va entre comillas

Ya que las fuentes que se utilizan en la página deben estar instaladas en el ordenador del usuario, cuando se quiere disponer de un diseño complejo con fuentes muy especiales, se debe recurrir a soluciones alternativas.

font-size

- Una vez seleccionado el tipo de letra, se puede modificar su tamaño mediante la propiedad **font-size**.
- CSS recomienda indicar el tamaño del texto en la unidad em o en porcentaje (%).
- Además, es habitual indicar el tamaño del texto en puntos (pt) cuando el documento está específicamente diseñado para imprimirlo.

```
h1 { font-size: 1.5em; }
```

color

- Aunque el color por defecto del texto depende del navegador, todos los navegadores principales utilizan el color negro. Para establecer un color diferente al color por defecto, utilizaremos la propiedad color.

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

- Como el valor de la propiedad color se hereda, normalmente se establece la propiedad color en el elemento <body> para establecer el color de letra de todos los elementos de la página:

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

- 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:

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

font-style

- La propiedad font-style se utiliza sobre todo para especificar el texto en cursiva.
- Esta propiedad tiene tres valores:
 - normal - El texto se muestra normalmente
 - italic - El texto se muestra en cursiva
 - oblique - El texto está "inclinado" (oblicua es muy similar a cursiva, pero menos compatible)

```
p {font-style: italic;}
```

font-weight

- Esta propiedad permite establecer el grosor con el que se muestran las letras del texto.
- Aunque admite varios valores, el más normal es bold para poner el texto en negrita.

```
p { font-weight: bold; }
```

font-variant

- La propiedad font-variant permite elegir la variante con la que se muestra el tipo de letra seleccionado mediante la propiedad font-family. Las variantes utilizadas suelen ser:
 - normal (normal)
 - small-caps (versalitas).

```
p { font-variant: small-caps; }
```

- En una fuente small-caps (versalita), todas las letras minúsculas se convierten en mayúsculas. Sin embargo, las letras mayúsculas convertidas aparecerán en un tamaño de letra más pequeño que las letras mayúsculas originales en el texto.

font

- La propiedad font permite establecer algunas o todas las propiedades tipográficas del texto de forma simultánea. Por este motivo, font es una de las propiedades abreviadas que se conocen con el nombre de "*propiedades shorthand*".
- Las propiedades que se pueden establecer, son (en orden): "font-style font-variant font-weight font-size / line-height font-family"
 - Los atributos font-size y font-family son valores obligatorios. El resto de los atributos son opcionales y en caso de no ponerlos, el navegador tomará los valores por defecto.

```
p {font: italic bold 12px/30px Georgia, serif;}
```



Manos a la obra

○ En el documento pagina, definir los siguientes estilos:

- h1: Fuente verdana, arial, helvetica, sans-serif, tamaño de fuente 2 em, color navy y negrita.
- H2: Fuente Arial, Helvetica, sans-serif, tamaño de fuente 1,5em, color blue y negrita.
- H3: Igual que h2 solo que con un tamaño de 1em.
- H4: igual que h2 y h3 solo que con un tamaño de 0,8em y cursiva
- Lista numerada. Fuente times new roman, times, serif de color rojo
- Elemento con clase etiqueta: negrita y cursiva

@font-face

- La regla @font-face permite al autor proporcionar sus propias fuentes.
 - Esto hace que no dependamos de las fuentes que los usuarios tengan instaladas en sus computadoras.
- En la nueva regla @font-face, primero debe definir un nombre para la fuente (por ejemplo, miFuente) y, a continuación, apuntar mediante una url al archivo de fuente.

```
@font-face {  
    font-family: miFuente;  
    src: url(cappella.ttf);  
}
```

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

Consejo: Es mejor utilizar letras minúsculas para la url de la fuente ya que las letras mayúsculas pueden dar resultados inesperados en IE

@font-face

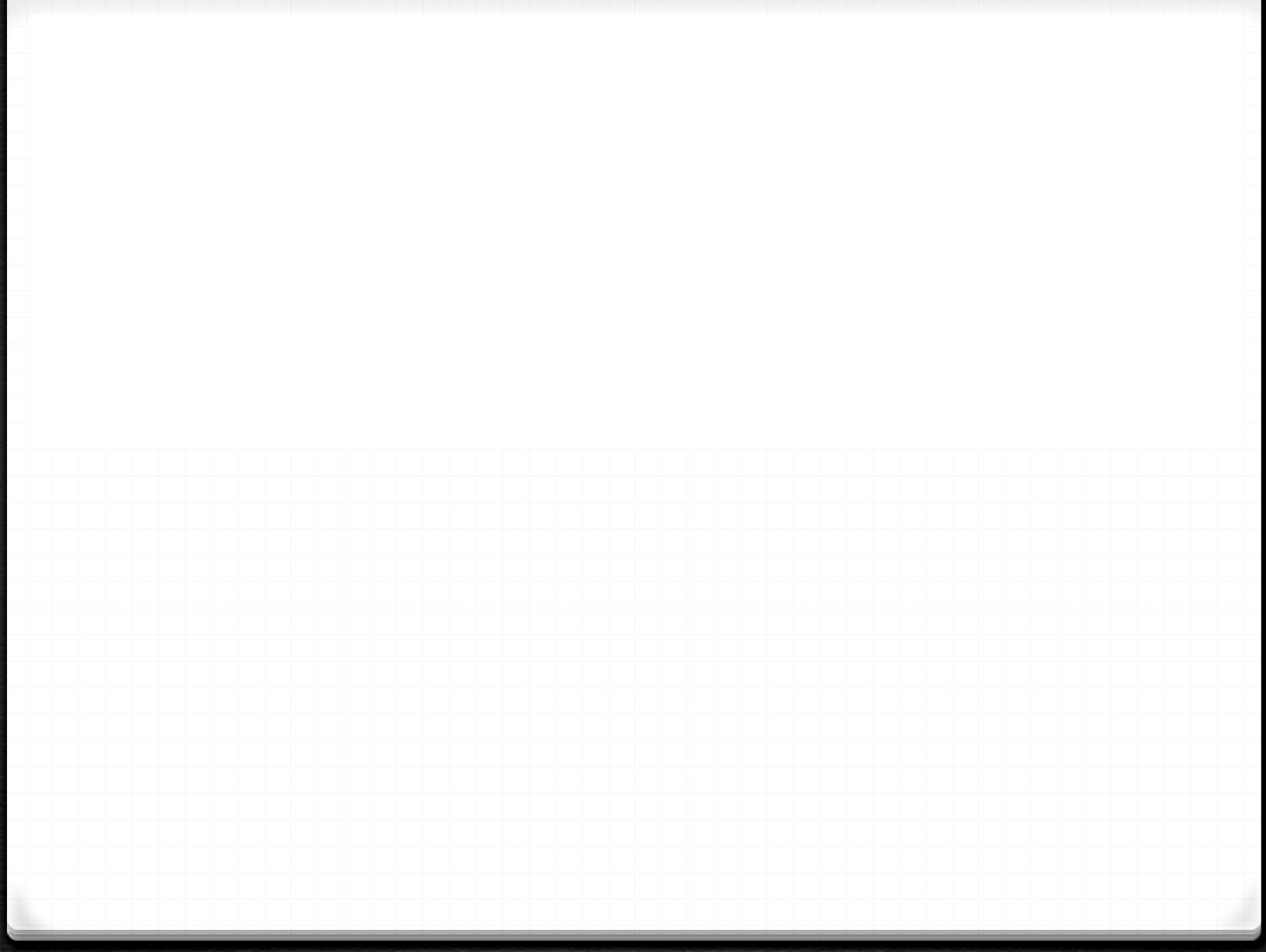
- El problema de @font-face es que no todos los navegadores utilizan el mismo formato de fuente:
 - **TTF**: Formato muy estándar.
 - **EOT**: El formato Embedded Open Type (.eot) fue creado por Microsoft como pionero en Internet Explorer 6 para agregar fuentes a través de font-face.
 - **WOFF**: Web Open Font Format es el formato oficial de la W3C, creado por Mozilla y ahora apoyado por Chrome, Opera e IE9 y 10.
 - **SVG**: Scalable Vector Graphics es un formato de dibujo vectorial soportado por todos los navegadores y por herramientas como Adobe Illustrator. SVG también puede contener fuentes por dentro, porque una fuente a fin de cuentas es un vector. SVG es obligatorio para darle soporte a Safari Mobile (iPhone, iPad).

@font-face

- Por lo tanto, la sintaxis correcta de @font-face es:

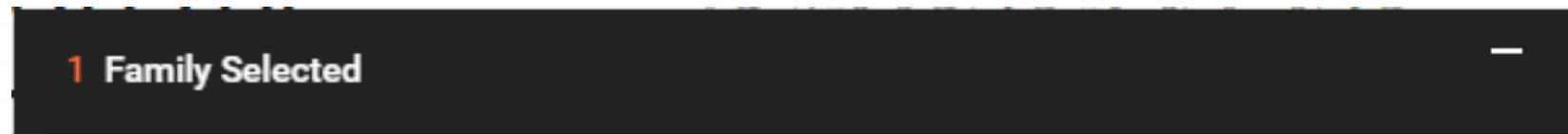
```
@font-face {  
    font-family: miFuente;  
    src: url("fonts/cappella.eot"); /* IE9 */  
    src: url('fonts/cappella.eot?#iefix') format('embedded-opentype'),  
        url("fonts/cappella.woff") format("woff"),  
        url("fonts/cappella.ttf") format("truetype"),  
        url("fonts/cappella.svg#IDdelafuente") format("svg");  
}
```

- .eot siempre debe ir el primero porque IE siempre quiere que le pongamos el primero frente al resto
- Las fuentes svg tienen un id que debe de ir después del nombre del archivo que contiene la fuente. Este id lo puedes encontrar abriendo el fichero de la fuente con cualquier editor de texto y buscando el atributo id.



Google Fonts

- Navega hasta el sitio de google fonts (www.google.com/fonts) y busca la fuente que te guste.
- Elige la fuente siguiendo uno de los dos métodos siguientes:
 - Haz clic sobre el botón + que aparece a la derecha del nombre de la fuente.
 - Haz clic sobre la fuente elegida y luego clic sobre el botón Select this font que se encuentra en la parte superior derecha.
- De cualquiera de las dos formas anteriores, el navegador mostrará una barra negra con el número de fuentes seleccionadas:



- Haremos clic sobre dicha barra y se abrirá una nueva ventana

Google Fonts

Si elegimos el método standard en el documento html dentro del head deberemos poner la línea link que aparece en esta pantalla

The screenshot shows the 'Embed' tab selected on the Google Fonts website. It displays two options: 'STANDARD' and '@IMPORT'. The 'STANDARD' option is highlighted with a red border. Below it is a code snippet for a `<link>` tag:

```
<link href="https://fonts.googleapis.com/css?family=UnifrakturCook:700" rel="stylesheet">
```

En el archivo CSS utilizaremos la fuente dentro de la propiedad font-family como si fuera cualquier otra fuente

Specify in CSS
Use the following CSS rules to specify these families:

```
font-family: 'UnifrakturCook', cursive;
```

Google Fonts

Si elegimos
@import
pondremos la línea
import al principio
del archivo css y
luego utilizaremos
la fuente dentro de
la propiedad font-
family como si fuera
cualquier otra
fuente

EMBED CUSTOMIZE Load Time: Fast

Embed Font

To embed your selected fonts into a webpage, copy this code into the <head> of your HTML document.

STANDARD @IMPORT

```
<style>
@import url('https://fonts.googleapis.com/css?family=UnifrakturCook:700');
</style>
```

Specify in CSS

Use the following CSS rules to specify these families:

```
font-family: 'UnifrakturCook', cursive;
```

For examples of how fonts can be added to webpages, see the [getting started guide](#).

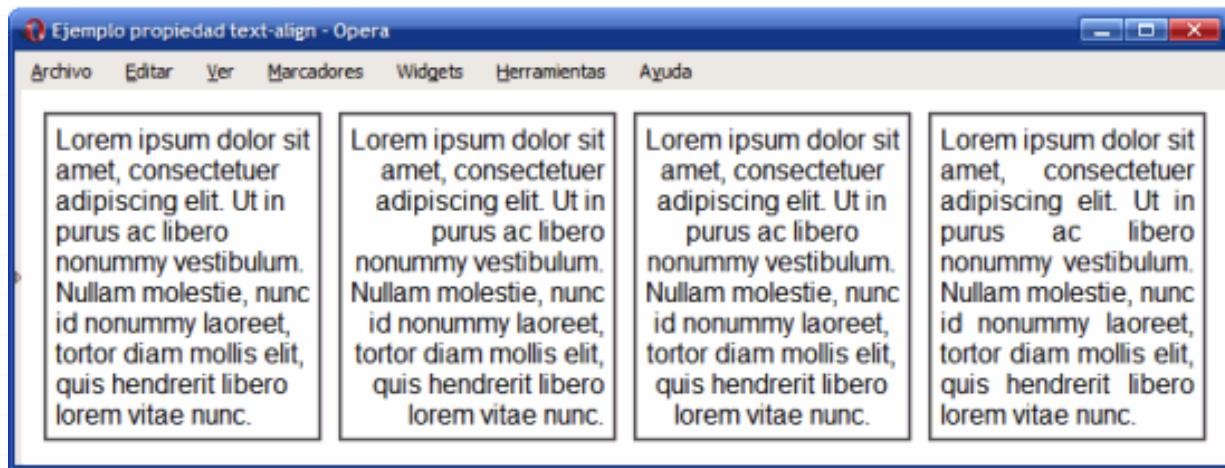
Nota. La línea import si la incluimos en el CSS, no lleva las etiquetas style externas. Si lo incluimos dentro del head en el documento html entonces sí deberíamos incluir las etiquetas style.

Formatos de texto

Text-align

- Esta propiedad nos permite definir la alineación del texto.
- 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).

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



Text-decoration

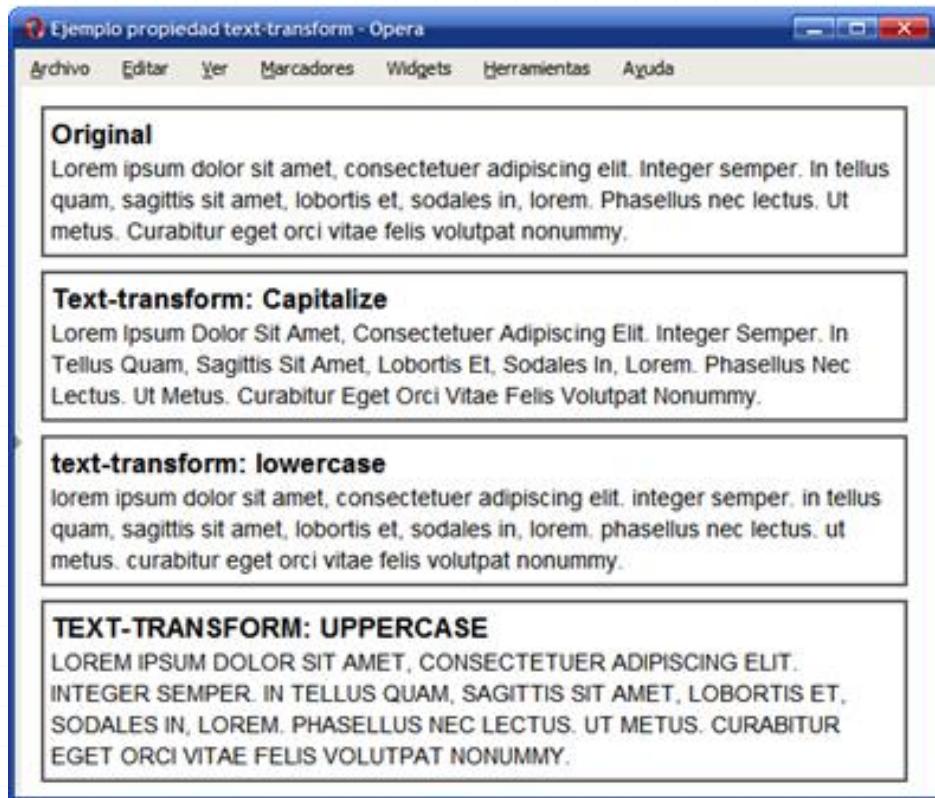
- Esta propiedad nos permite añadir o eliminar una decoración del texto.
- Los posibles valores de esta propiedad son:

Valor	Efecto	Ejemplo
overline	Añade una línea en la parte superior del texto	Texto con linea por encima
line-through	Muestra el texto tachado con una línea continua	Texto tachado
underline	Subraya el texto. Se aconseja no utilizarlo porque puede llevar a creer que el texto es un enlace.	Texto subrayado
none	Quita cualquier tipo de línea que apareciera con el texto. Esta opción se utiliza cuando se quiere quitar los subrayados de los enlaces	

```
h1 { text-decoration: overline; }
```

text-transform

- 0 La propiedad text-transform permite mostrar el texto original transformado en un texto completamente en mayúsculas (uppercase), en minúsculas (lowercase) o con la primera letra de cada palabra en mayúscula (capitalize).



```
p {  
    text-transform: uppercase;  
}
```

text-indent

- Esta propiedad permite definir una sangría de primera línea en un párrafo.
- La siguiente imagen muestra la comparación entre un texto largo formado por varios párrafos sin tabular y el mismo texto con la primera línea de cada párrafo tabulada:



p { text-indent: 2em; }

letter-spacing

- La propiedad letter-spacing nos permite establecer la separación entre las letras que forman las palabras.

Valor	Ejemplo
Sin propiedad letter-spacing	Interletraje normal
<code>h1 { letter-spacing: 3px; }</code>	Interletraje mayor
<code>h1 { letter-spacing: -3px; }</code>	Interletraje menor

- La propiedad word-spacing nos permite establecer la separación entre las palabras que forman los textos.

Valor	Ejemplo
Sin propiedad word-spacing	Separacion entre palabras normal
<code>h1 { word-spacing: 10px; }</code>	Separacion entre palabras mayor
<code>h1 { letter-spacing: -5px; }</code>	Separacion entre palabras menor

Line-height

- Esta propiedad define la altura que ocupa cada línea de texto lo que nos permite controlar el interlineado de un texto.

p { line-height: 0.8; }

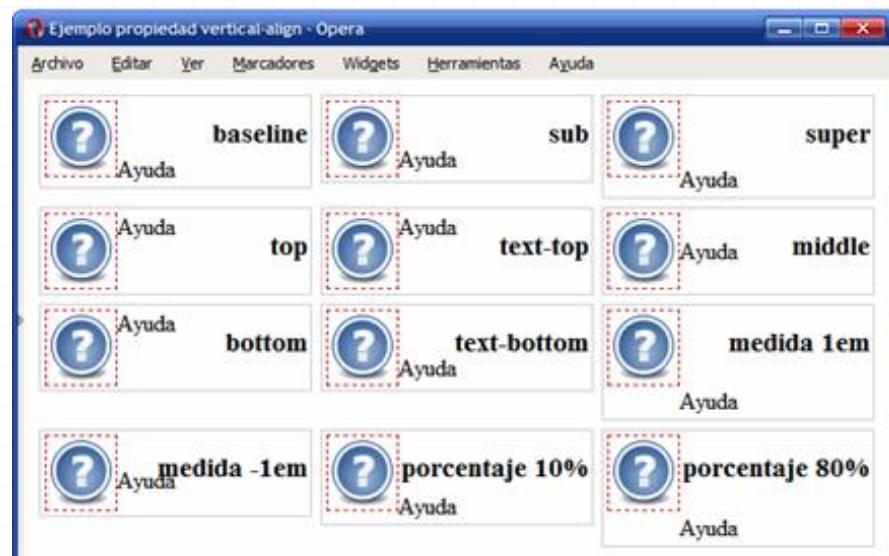
- Siempre que se utilice de forma moderada, el interlineado mejora notablemente la legibilidad de un texto:



vertical-align

- Uno de los principales problemas del diseño de documentos y páginas mediante CSS consiste en la alineación vertical en una misma línea de varios elementos diferentes como imágenes y texto.
- Para controlar esta alineación, CSS define la propiedad **vertical-align**.
- Los valores que puede tomar esta propiedad son:
 - `baseline | length | sub | super | top | text-top | middle | bottom | text-bottom | initial | inherit;`

Length puede ser una medida absoluta, una medida relativa o un porcentaje





Manos a la obra

- En el documento página del tema anterior, añadir los siguientes formatos de párrafo a nuestros elementos html:
- h1: alineación: centrada, espaciado entre palabras: 10px;
- h2: alineación: centrada, espaciado entre letras: 3px;
- h3: alineación: centrada
- H4: subrayado
- Párrafos: alineación justificada, sangría primera línea 2 em, altura de línea 1 em
- Lista numerada: alineación justificada
- Elementos de clase etiqueta: mayúsculas