



XUNTA DE GALICIA

CONSELLERÍA DE CULTURA, EDUCACIÓN
E ORDENACIÓN UNIVERSITARIA

Unidad 2

Hojas de estilo

CSS

Módulo: Diseño de interfaces Web
2º Curso Desarrollo de Aplicaciones Web

Contenido

- 2.1. Introducción
- 2.2. Incluir CSS en un documento HTML
- 2.3. Definir reglas de estilo CSS
- 2.4. Selectores
- 2.5. Comentarios
- 2.6. Unidades de medida y colores
- 2.7. Medios CSS
- 2.8. Modelo de cajas
- 2.9. Propiedades de Colores y fondos
- 2.10. Propiedades de Textos y fuentes
- 2.11. Propiedades de Posicionamiento y visualización
- 2.12. Propiedades de listas
- 2.13. Propiedades de tablas
- 2.14. Otras propiedades
- 2.15. GRID (cuadrícula)
- 2.16. Diseño web adaptativo (Responsive Web Design)

2.1. Introducción



- **¿Qué es CSS?**
 - **CSS** es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML.
 - CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.
 - Ventajas:
 - Obliga a crear documentos HTML bien definidos y con significado completo (documentos semánticos)
 - Mejora la accesibilidad del documento
 - Reduce la complejidad del mantenimiento
 - Permite visualizar el documento en múltiples dispositivos

Introducción. Evolución

- Las hojas de estilo aparecieron alrededor de 1970.
- El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos.
- 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.

Introducción. Evolución

- El organismo [W3C](#) (World Wide Web Consortium), encargado de crear todos los estándares relacionados con la web, propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML.
- En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML.
- 1996: primera recomendación oficial: CSS nivel 1
- 1998: CSS nivel 2.
- CSS 2.1: Revisión de CSS nivel 2
- 1999: CSS 3, dividida en módulos (fondos y colores, diseños multicolumna, etc.)

Introducción.

Soporte de CSS en los navegadores

Navegador	Motor	CSS 1	CSS 2.1	CSS 3
Microsoft Edge	Trident	Completo	Completo	Todos los selectores, pseudo-clases y algunas propiedades
Firefox	Gecko	Completo	Casi completo	Selectores, pseudo-clases y algunas propiedades
Safari	WebKit	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades
Google Chrome	WebKit	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades

Comprobar compatibilidad en: <http://caniuse.com/>

2.2. Incluir CSS en un documento HTML

- Una de las principales características de CSS es su flexibilidad y las diferentes opciones que ofrece para realizar una misma tarea. De hecho, existen tres opciones para incluir CSS en un documento HTML:
 1. Incluir CSS en los elementos HTML
 2. En el mismo documento HTML
 3. Definir CSS en un archivo externo

Incluir CSS en un documento HTML

1. Incluir CSS en los elementos HTML

- Es el peor método y el menos utilizado, ya que tiene los mismos problemas que la utilización de las etiquetas ``.
- Solamente se utiliza en determinadas situaciones en las que se debe incluir un estilo muy específico para un solo elemento concreto.

```
<body>  
<pstyle="color: black; font-family: Verdana;">Un párrafo de texto.</p>  
</body>  
</html>
```


Incluir CSS en un documento HTML

2. Incluir CSS en el mismo documento HTML

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

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

Incluir CSS en un documento HTML

- **Incluir CSS en el mismo documento HTML.** Se emplea:
 - Cuando se define un número pequeño de estilos.
 - 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.
 - *Inconveniente:* 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.

Incluir CSS en un documento HTML

3. Definir CSS en un archivo externo

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

Incluir CSS en un documento HTML

- **Definir CSS en un archivo externo. Pasos**

- 1) Se crea un archivo de texto y se le añaden como contenido las reglas de estilo CSS.
- 2) Se guarda el archivo con extensión .css
- 3) En la página HTML se enlaza el archivo CSS externo mediante la etiqueta <link>:

```
<!DOCTYPE html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos CSS en un archivo externo</title>
<link rel="stylesheet" type="text/css" href="estilos.css" media="screen" />
</head>
<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

Incluir CSS en un documento HTML

```
<head>  
  <link rel="stylesheet" type="text/css" href="styles.css" media="screen">  
</head>
```

- **Etiqueta <link>**

- Normalmente, la etiqueta <link> incluye cuatro atributos cuando enlaza un archivo CSS:
- **rel**: relación 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. Sus valores están estandarizados y para los archivos CSS su valor siempre es `text/css`
- **href**: indica la URL del archivo CSS que contiene los estilos. La URL indicada puede ser relativa o absoluta y puede apuntar a un recurso interno o externo al sitio web.
- **media**: indica el medio en el que se van a aplicar los estilos del archivo CSS. *Más adelante se explican en detalle los medios CSS y su funcionamiento.*

Incluir CSS en un documento HTML

- Utilizar la etiqueta `<link>`. **Ventajas:**
 - Se puede incluir un mismo archivo CSS en multitud de páginas HTML: estilo homogéneo en todas las páginas del sitio.
 - El mantenimiento del sitio web se simplifica al máximo.
- Alternativa: Utilizar la etiqueta `<style>` para archivo externo:

```
<!DOCTYPE html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos CSS en un archivo externo</title>
<style type="text/css" media="screen">
    @import 'estilos.css';
</style>
</head>
<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

2.3. Definir reglas de estilo CSS



- **Regla:** cada uno de los estilos que componen una hoja de estilos CSS.
- Cada regla está compuesta de una parte de "*selectores*", un símbolo de "*llave de apertura*" (`{`), otra parte denominada "*declaración*" y por último, un símbolo de "*llave de cierre*" (`}`).



```
selector {  
    propiedad : valor ;  
    propiedad : valor  
}
```

Definir reglas de estilo CSS

- Las reglas de estilos CSS se componen de diferentes partes:
 - **Selector:** elemento/s HTML a los que se aplica la regla CSS.
 - **Declaración:** estilos que se aplican a los elementos. Una o más propiedades CSS.
 - **Propiedad:** característica que se modifica en el elemento seleccionado.
 - **Valor:** establece el nuevo valor de la característica modificada en el elemento.
- 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.

Definir reglas de estilo CSS

- Por ejemplo, una regla de estilo básica sería la siguiente:

```
footer{  
    background-color: black;  
    color:white;  
}
```

- En este ejemplo se crea una regla que dota de un diseño específico al pie de página (**footer**). En concreto la regla especifica que el color de fondo (**background-color**) del **footer** será negro (**black**) y el color del texto (**color**) será blanco (white).

2.4 Selectores

```
selector #id .clase :pseudoclase ::pseudoelemento [atributo] {  
    propiedad : valor ;  
    propiedad : valor  
}
```

- Son los elementos HTML a los que se aplica la regla CSS.

- **Tipos:**

1. Selector universal
2. De tipo o de etiqueta
3. Descendiente
4. De clase
5. De ID
6. Avanzados
7. Pseudo-elementos
8. Pseudo-clases



REGLA

```
Selector      Declaración  
h1 { color: #990000; }
```

Propiedad Valor

Selectores

1. Selector Universal

- Se utiliza para seleccionar todos los elementos de la página.
- El selector universal se indica mediante un asterisco (*).
- A pesar de su sencillez, no se utiliza habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página.
- Ejemplo:

```
/* Aplicar un borde de 1 pixel  
punteado a todos los elementos  
de la pagina web */  
* {  
    border: 1px;  
    border-style: dotted;  
}
```

Selectores

2. Selector de tipo o de etiqueta

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

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

Selectores

- **Selector de tipo o de etiqueta**

- Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, se pueden encadenar los selectores.
- Para ello, se incluyen todos los selectores separados por una coma (,).

- Ejemplo:

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

- En las hojas de estilo complejas, es habitual agrupar las propiedades comunes de varios elementos en una única regla CSS y posteriormente definir las propiedades específicas de esos mismos elementos.

```
strong {  
  border-color: red;  
  background: white;  
}  
  
span {  
  border-color: red;  
  background: white;  
}  
  
img {  
  border-color: red;  
  background: white;  
}
```



```
strong, span, img {  
  border-color: red;  
  background: white;  
}
```

Selectores

3. Selector descendiente

- Selecciona los elementos que se encuentran dentro de otros elementos.
- Un elemento es **descendiente** de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.
- Permiten aumentar la precisión del selector de tipo o etiqueta.
- Ejemplo:

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

- *En el ejemplo, selecciona todos (hijos, nietos, etc.) los elementos de la página que se encuentren dentro de un elemento <p>*

Selectores

- **Selector descendiente**

- Los selectores descendientes siempre están 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 selectores anteriores indican el lugar en el que se debe encontrar ese elemento.
- Ejemplo:
p a span em{text-decoration: underline; }
- La regla de estilo se aplica únicamente a los elementos em

Selectores

4. Selector de clase

- Permite asociar un estilo a aquellos elementos HTML que han sido clasificados dentro una clase específica.
- Esto se realiza utilizando el atributo ***class*** dentro de los elementos HTML.
- En una misma página HTML varios elementos diferentes pueden utilizar el mismo valor en el atributo class.
- Imprescindibles para diseñar páginas web complejas, ya que permiten disponer de una precisión total al seleccionar los elementos.
- Permiten reutilizar estilos.

Selectores

- **Selector de clase. Ejemplo**

```
<h1 class="main">Titulo</h1>
```

En el archivo HTML
(uso)

```
h1.main {  
    font-weight: bolder;  
    font-size: 18pt;  
}  
.main {  
    font-weight: bolder;  
    font-size: 18pt;  
}
```

En el archivo CSS
(definición)

Selectores

- **Selector de clase**

- En ocasiones, es necesario restringir el alcance del selector de clase.
- Se pueden combinar los selectores, por ejemplo: el selector de tipo y el selector de clase. Ejemplo:

```
<body>
  <p class="destacado"> Loremipsum dolor sitamet...</p>
  <p>Nunc sed lacus et <a href="#" class="destacado">
    estadipiscing</a> accumsan...</p>
  <p>Classaptenttaciti <em class="destacado"> sociosquad</em>
    litora...</p>
</body>

p.destacado {color: red}
```

Selectores

- **Selector de clase**

- Es posible aplicar los estilos de varias clases CSS sobre un mismo elemento.
- Los diferentes valores del atributo class se separan con espacios en blanco.
- Se aplica en cascada desde el CSS
- Ejemplo:

```
.error { color: red; }  
.destacado { font-size: 15px; }  
.especial { font-weight: bold; }
```

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

Selectores

- No confundir los selectores:

```
/* 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

5. Selectores de ID

- Se emplean para los casos que es necesario aplicar estilos CSS a un único elemento de la página.
- Permite seleccionar un elemento de la página a través del valor de su atributo id.
- 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:

Selectores

- **Selectores de ID. Ejemplo**

```
#saludo {  
  background-color: blue;  
  color: white;  
}
```

En el archivo CSS
(definición)

```
<div id="saludo">  
  <p>¡Hola, visitante! ¡Bienvenido a esta página!</p>  
</div>
```

En el archivo HTML
(uso)

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

Selectores

- **Selector de ID**

- Al igual que los selectores de clase, en este caso también se puede restringir el alcance del selector mediante la combinación con otros selectores.

- Ejemplo:

p#aviso {color: blue; }

- En este ejemplo aplica la regla CSS solamente al elemento de tipo <p> que tenga un atributo id igual al indicado.

Selectores

- **Selector de ID**

- 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{ ...}
```


Selectores

- **Combinación de selectores**

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

```
.aviso .especial { ...}  
div.aviso span.especial { ...}
```

- La combinación de selectores puede llegar a ser todo lo compleja que sea necesario:

```
ul#menuPrincipal li.destacado a#inicio{ ... }
```

Selectores

6. Selectores avanzados

- CSS define otros selectores más avanzados que permiten simplificar las hojas de estilos.
- Tipos:
 - Selector de hijos
 - Selector adyacente
 - selector general de hermanos
 - Selector de atributos

Selectores avanzados

- **Selector de hijos**

- Se utiliza para seleccionar un elemento que es hijo directo de otro elemento y se indica con 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 `p > span` indica “cualquier elemento span que sea hijo directo de un elemento p, por lo cual, se aplica la regla al primer elemento span que aparece, pero no al segundo, ya que éste no es hijo directo.*

Selectores avanzados

- **Selector de hijos**

- No confundir con el selector descendente:

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

```
<p><a href="#">Enlace1</a></p>  
<p><span><a href="#">Enlace2</a></span></p>
```

- *El primer selector es descendente y por tanto se aplica a todos los elementos <a> que se encuentran dentro de elementos <p>, es decir, a todos los enlaces que están en un párrafo.*
- *Sin embargo el segundo tipo de selector es de hijos, y obliga a que el elemento <a> sea hijo directo de <p>, por tanto no se aplican los estilos al segundo enlace.*

Selectores avanzados

- **Selector adyacente**

- Se utiliza para seleccionar elementos que en HTML de la página se encuentran justo a continuación de otros elementos. Se emplea el signo “+” para separar los elementos:

```
elemento1 + elemento2 { ... }
```

```
<body>
```

```
<h1>Titulo1</h1>
```

```
<h2>Subtítulo</h2>
```

```
...
```

```
<h2>Otro subtítulo</h2>
```

```
...
```

```
</body>
```


```
h2 { color: green; }
```

```
h1 + h2 { color: red }
```

- Todos los <h2> se verán en color verde excepto aquellos que estén justo después de los <h1>, que se verán en color rojo.

Selectores avanzados

- **Selector general de hermanos**

- Se utiliza para seleccionar el elemento2 que es hermano de elemento1 y se encuentra detrás en el código HTML.
- Generaliza el selector adyacente.
- Se emplea el signo “” para separar los elementos:

```
1 p ~ span {
2   color: red;
3 }
```

elemento ~ elemento { *estilos* }

Este span no es rojo.

Aquí hay un párrafo.

Aquí hay algo de código. Aquí hay un span.
Es rojo porque va precedido de un párrafo y
ambos comparten el mismo padre.

```
1 <span>Este span no es rojo.</span>
2 <p>Aquí hay un párrafo.</p>
3 <code>Aquí hay algo de código.</code>
4 <span>Aquí hay un span. Es rojo porque va precedido de un párrafo y amb
```

Selectores avanzados

- **Selector de atributos**

- Éstos permiten seleccionar elementos HTML en función de sus atributos y/o valores de esos atributos.
- Tipos:
 - **elemento[atributo]**, selecciona los elementos que tienen establecido el atributo llamado atributo, independientemente de su valor.
 - **elemento[atributo=valor]**, selecciona los elementos que tienen establecido un atributo llamado atributo con un valor igual a valor.
 - **elemento[atributo~=valor]**, selecciona los elementos que tienen establecido un atributo llamado atributo y al menos uno de los valores del atributo es valor.
 - **elemento[atributo|=valor]**, selecciona los elementos que tienen establecido un atributo llamado 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.*

Selectores avanzados

- **Selector de atributo**

- **Más tipos:**

- **elemento[atributo^=valor]**, selecciona los elementos con ese atributo y que su valor comienza por la cadena de texto indicada en valor.
 - **elemento[atributo\$=valor]**, selecciona los elementos con ese atributo y que su valor termina por la cadena de texto indicada en valor.
 - **elemento[atributo*=valor]**, selecciona los elementos con ese atributo y que su valor contiene la cadena de texto indicada en valor.

Ejemplos

```
/* Se muestran de color azul todos los enlaces que tengan  
un atributo "class", independientemente de su valor */  
a[class] { color: blue; }
```

```
/* Se muestran de color azul todos los enlaces que tengan  
un atributo "class" con el valor "externo" */  
a[class="externo"] { color: blue; }
```

```
/* Se muestran de color azul todos los enlaces que apunten  
al sitio "http://www.ejemplo.com" */  
a[href="http://www.ejemplo.com"] { color: blue; }
```

```
/* Se muestran de color azul todos los enlaces que tengan  
un atributo "class" en el que al menos uno de sus valores  
sea "externo" */  
a[class~="externo"] { color: blue; }
```

```
/* Selecciona todos los elementos de la página cuyo atributo  
"lang" sea igual a "en", es decir, todos los elementos en inglés */  
*[lang=en] { ... }
```

```
/* Selecciona todos los elementos de la página cuyo atributo  
"lang" empiece por "es", es decir, "es", "es-ES", "es-AR", etc. */  
*[lang|= "es"] { color : red }
```

Selectores. Tabla resumen

Selector	Afectará a:
*	Cualquier elemento
E	Un elemento (etiqueta HTML) específico, como por ejemplo div o article.
E1 E2	Un elemento (E2) dentro de otro elemento (E1), anidado. Funciona en toda la estructura hereditaria.
E1 > E2	Un elemento (E2) que sea hijo de otro (E1).
E1 + E2	Un elemento (E2) que va detrás y es adyacente a otro elemento (E1)
.clase	Afectará a todo los elementos a los que se le haya asignado esa clase. Ejemplo, <div class="caja lateral">
E.clase	Únicamente al elemento E al que se le haya especificado esa clase.
#ID	Al elemento con el identificador #ID
E#ID	Al elemento E con el identificador ID
E[attr="valor"]	A todos los elementos E cuando el atributo attr sea igual al valor especificado

Selectores

7. Pseudo-elementos

- Se refiere a una parte de un elemento que, aunque actúa como éste, no es un elemento en sí.
- Ejemplo: la primera letra de un párrafo, o la primera línea....
- Hay 5 tipos (*En CSS3 se añaden los :: delante para diferenciarlos de los pseudo-elementos de CSS2.1*)
 - **:first-letter**
 - **:first-line**
 - **:first-child**
 - **:before**
 - **:after**

Selectores. Pseudo-elementos

- **:first-letter**

- Permite seleccionar la primera letra de la primera línea de texto de un elemento.

- Ejemplo:

```
p{  
    font-size: 14px;  
}  
  
p:first-letter {  
    font-size: 2em;  
}
```

- *Esta regla muestra la primera letra de cada párrafo con el doble de tamaño.*

Selectores. Pseudo-elementos

- **:first-line**

- Permite seleccionar la primera línea de texto de un elemento.
- Ejemplo:

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

- *Esta regla muestra en mayúscula la primera línea de cada párrafo.*

Selectores. Pseudo-elementos

- **:first-child**

- Permite seleccionar el primer elemento hijo de un elemento.
- Ejemplo:

```
div p:first-child {  
    color: red;  
}
```

- *Esta regla muestra como identificar al primer párrafo dentro de un bloque <div> sin necesidad de asignar ninguna clase al párrafo.*

Selectores. Pseudo-elementos

- **:before**

- Permite insertar contenido antes de un elemento.
- Ejemplo:

```
1 p.errores:before {  
2   content: "ERRORES COMUNES";  
3 }
```

```
1 <p class="errores">Contenido del párrafo..</p>
```

- *Esta regla permite* que cuando encuentre un párrafo de clase *errores*, inserte justo antes del mismo “ERRORES COMUNES”.

Selectores. Pseudo-elementos

- **:after**

- Hace justo lo contrario a :before, es decir, permite insertar contenido después de un elemento.

- Ejemplo:

```
1 p.ejemplo:after {  
2   content: "";  
3   display: block;  
4   width: 200px;  
5   height: 50px;  
6   border: 1px solid red;  
7 }
```

- Siempre tienen que llevar el atributo content, sino no funcionarán.

Selectores. Pseudo-clases

- **Pseudo-clases**

- Con las pseudo-clases es posible aplicar efectos especiales a determinados elementos de una página HTML, como a enlaces o imágenes, cuando se realizan algunas acciones específicas con el ratón, como pasar sobre una imagen o enlace, o hacer clic en ellos...
- Tipos:

:link	:nth-last-child (n)	:only-of-type
:visited	:nth-of-type(n)	: first-of-type
:hover	:nth-last-of-type(n)	:not(selector)
:active	:only-child	:empty
:focus	:fist-child	:enabled /:disabled
:nth-child(n)	:last-child	:checked

Selectores. Pseudo-clases

- **:link**

- Usamos la pseudo-clase :link para seleccionar todas las etiquetas de enlaces a las que aún no se les ha dado clic.

- **:visited**

- Se aplica a todos los enlaces que han sido visitado al menos una vez por el usuario.

```
a:link {  
    color: red;  
}  
  
a:visited {  
    color: green;  
}
```

Selectores. Pseudo-clases

- **:hover**

- Se aplica a cualquier elemento HTML, no sólo a enlaces.
- Se activa cuando el usuario pasa el ratón por encima de cualquier elemento.

- **:active**

- Se produce cuando el usuario activa un elemento, por ejemplo cuando se pulsa el ratón sobre un elemento. Se aplica durante un espacio de tiempo muy pequeño.

- **:focus**

- Permite aplicar un estilo a un elemento cuando éste recibe el foco, es decir, la atención del usuario.

Selectores. Pseudo-clases

- **:nth-child (n)**

- Acepta un número entero como parámetro (n).
- Permite aplicar un estilo a los elementos que se indique, en base a sus posiciones en una lista de elementos hijos dentro de un elemento padre.

- Ejemplo:

```
1  li:nth-child(3) {  
2    color: red;  
3  }
```

- *Selecciona el tercer elemento de la lista que se muestra y aplica el estilo indicado (en color rojo).*

Selectores. Pseudo-clases

- **:nth-last-child (n)**

- Funciona igual que :nth-child, excepto que selecciona los elementos empezando a contar por el final, en lugar de por el principio.
- Ejemplo:

```
1  li:nth-last-child(2) {  
2    color: red;  
3  }
```

Selecciona el segundo elemento de la lista pero empezando por el final.

- **:nth-of-type(n)**

- Selecciona los elementos HTML de un tipo concreto dentro de los hijos de un elemento padre.

```
1  ul:nth-of-type(3) {  
2    border: 1px solid black;  
3  }
```

Selecciona la tercera lista de elementos desordenada (ul) y le aplica el estilo indicado

Selectores. Pseudo-clases

- **:nth-last-of-type(n)**

- Selecciona los elementos HTML de un tipo concreto dentro de los hijos de un elemento padre, pero contando desde el último hijo.

- Ejemplo

```
p:nth-last-of-type(2) {  
    background: red;  
}
```

Selecciona el segundo párrafo empezando por el final.

- **:only-child**

- Permite aplicar un estilo a un elemento que es hijo único de otro elemento padre.

```
1  div p:only-child {  
2    color: red;  
3  }
```

```
1  <div><p> Mi Párrafo. </p></div>  
2  
3  <div>  
4    <p> Dos Párrafos en Total. </p>  
5    <p> Dos Párrafos en Total. </p>  
6  </div>
```



Selectores. Pseudo-clases

- **:first-child**

- Permite aplicar un estilo al primer hijo del padre del elemento indicado.
- Equivalente a `:nth-child(1)`
- Ejemplo:

```
1  ul li:first-child {  
2    border-top: none;  
3  }
```

Selecciona el primer elemento de la lista y le aplica el estilo (sin borde).

- **:last-child**

- Permite aplicar un estilo al último elemento del padre del elemento HTML indicado.
- Equivalente a `:nth-last-child(1)`
- Ejemplo:

```
1  ul > li:last-child {  
2    color: green;  
3  }
```

Selecciona el último elemento de la lista y le aplica el estilo (color verde).

Selectores. Pseudo-clases

- **:only-of-type**

- Se aplicará un estilo definido a elementos que no tienen hermanos dentro de su elemento padre.
- Ejemplo:

```
1  ul > li:only-of-type {  
2    font-weight: bold;  
3  }
```

Aplica el estilo (negrita) a los elementos de lista que son únicos, es decir, lista de un solo elemento.

- **:first-of-type**

- Permite seleccionar los primeros elementos hermanos de su tipo.
- Ejemplo:

```
1  ul:first-of-type > li:nth-child(2) {  
2    font-weight: bold;  
3  }
```

```
01  <div>  
02    <p> My paragraph here. </p>  
03    <ul>  
04      <li> List Item 1 </li>  
05      <li> List Item 2 </li>  
06    </ul>  
07  
08    <ul>  
09      <li> List Item 3 </li>  
10      <li> List Item 4 </li>  
11    </ul>  
12  </div>
```

Aplica el estilo (negrita) únicamente al segundo elemento de la primera lista.

Selectores. Pseudo-clases

- **:not(selector)**

- Selecciona los elementos que no coincidan con el selector indicado y les aplica un estilo.
- Ejemplo:

```
/*Selecciona los h3 que no pertenezcan a  
la clase .rojo*/  
h3:not(.rojo){  
    font-style:italic;  
}
```

- **:empty**

- Selecciona los elementos que no tienen hijos, es decir, que no hay ninguna otra etiqueta HTML dentro.

Selectores. Pseudo-clases

- **:enabled**

- Selecciona los elementos de la página que están habilitados.

- **:disabled**

- Selecciona los elementos HTML que están deshabilitados.
- Ejemplo:

```
1  input:enabled {  
2    color: #22AA22;  
3  }  
4  input:disabled {  
5    color: #D9D9D9;  
6  }
```

Aplica diferentes colores de texto a los elementos <input> dependiendo de si están habilitados o no

Selectores. Pseudo-clases

- **:target**

- Representa el elemento único, si existe alguno, con un **id** coincidente con el identificador de fragmentos de la URI del documento.

- Ejemplo:

<https://developer.mozilla.org/es/docs/Web/CSS/:target>

- **:lang**

- Se utiliza para modificar elementos basándose en el idioma seleccionado.

- Ejemplo: <https://developer.mozilla.org/es/docs/Web/CSS/:lang>

- **:root**

- Selecciona el elemento raíz de un documento (<html>)

Selectores. Pseudo-clases

- Nuevas pseudo-clases para utilizar en **formularios**:
 - **:valid** - campo válido (dependerá del tipo de campo).
 - **:invalid** - campo inválido (dependerá del tipo de campo).
 - **:required** - campo requerido (marcado con el atributo “required”).
 - **:optional** - campo opcional (campo no marcado con el atributo “required”).
 - **:checked** - elemento marcado (o checked, válido para radio button o checkbox).
 - **:in-range** - valor dentro del rango indicado (para campos numéricos o de rango).
 - **:out-of-range** - valor fuera de rango (para campos numéricos o de rango).
 - **:read-only** - campo de solo lectura.
 - **:read-write** - campo de lectura / escritura.

Selectores.

Soporte en los navegadores

Comprobar el soporte de los selectores CSS3 en los diferentes navegadores:

- <http://tools.css3.info/selectors-test/test.html>
- <https://dev.l-c-n.com/CSS3-selectors/browser-support.php>



2.5 Comentarios en CSS

- Permitirán entender mejor la estructura del código que hemos creado, que nos dé una pista de qué reglas se definen y qué alcance tienen.
- Los navegadores los ignoran.
- El comienzo de un comentario se indica mediante los caracteres `/*` y el final del comentario se indica mediante `*/`

```
| /* Este es un comentario en CSS */  
|  
| /* Este es un  
| comentario CSS de varias  
| líneas */
```

2.6 Unidades de medida y colores

- Muchas de las propiedades CSS permiten indicar medidas y colores en sus valores.
- Además permite indicarlos de formas diferentes.
- **Unidades de medida:**
 - Para altura, anchura, márgenes, tamaño letra, etc.
 - Valores numéricos enteros o decimales seguidos de la unidad de medida.
 - Tipos:
 - **Absolutas:** establecen el valor de forma completa. Valor real es el indicado.
 - **Relativas:** definen el valor en relación a otra medida.

Unidades de medida y colores

- **Unidades Relativas:**

- **em:** relativa al tamaño de letra empleado. Aunque no es exacto, el valor 1em se aproxima a la anchura de la letra “M” del tipo y tamaño de letra utilizado.

Tamaño en puntos. 1em = 12 puntos.

- **ex:** relativo a la altura de la letra “x” del tipo y tamaño de la letra utilizada.

1ex = 0.5em

- **px:** relativa a la resolución de la pantalla del usuario.

- Ejemplo:

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

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

El tamaño de la letra de la página será el 90% del tamaño por defecto (que es el tamaño de referencia)

Unidades de medida y colores

- **Unidades absolutas:**

- Definen las medidas de forma completa. Sus valores reales no se calculan a partir de otro valor de referencia, sino que son los indicados.
- **in** (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.

Unidades de medida y colores

- Ejemplos de utilización de unidades absolutas:

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

- De las unidades de medida absolutas, la única que se utiliza con frecuencia es la de puntos (pt), para el tamaño de letra.

Unidades de medida y colores

- **Porcentajes**

- Es otra medida relativa, ya que referencia a otra medida.
- Ejemplos de uso:

- Tamaño de letra:

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

Equivalente a 2em y
1.5em

- Ancho de elementos:

```
div#contenido{width: 600px; }  
div.principal{width: 80%; }
```

80% ancho del padre
div.principal =
80% x 600px = 480px

```
<div id="contenido">  
<div class="principal">  
...  
</div>  
</div>
```

Unidades de medida y colores

- **Recomendaciones**

- 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.
- Se recomienda utilizar la medida “em” y porcentajes para tamaños de letras y pixel y porcentajes para definir el layout (estructura) del documento: anchura de columnas y elementos.

Unidades de medida y colores

- **Colores**

- Los colores en CSS se pueden indicar de cinco formas diferentes:

1. Palabras clave
2. RGB decimal
3. RGB Porcentual
4. RGB Hexadecimal
5. RGBA (CSS3)
6. HSL
7. HSLA



















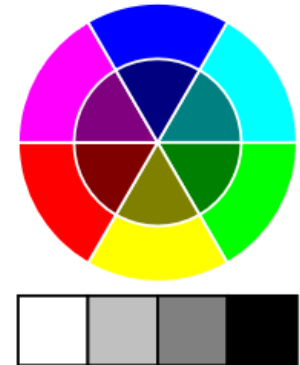
Unidades de medida y colores

1. Colores mediante Palabras clave

- En la recomendación HTML 3.2 (1997) se incluyeron dieciséis nombres de colores: **aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white y yellow.**
- Estos colores son los colores de la paleta VGA de Windows.

Casi no se utilizan en CSS. Muy limitados

Color	Nombre	Código RGB	Color	Nombre	Código RGB
	blue	#0000FF		navy	#000080
	fuchsia	#FF00FF		purple	#800080
	red	#FF0000		maroon	#800000
	yellow	#FFFF00		olive	#808000
	lime	#00FF00		green	#008000
	aqua	#00FFFF		teal	#008080
	gray	#808080		black	#000000
	white	#FFFFFF		silver	#C0C0C0



- Además los navegadores modernos soportan muchos otros nombres: [Tabla de colores](#)

Unidades de medida y colores

2. RGB decimal

- RGB: define un color indicando la cantidad de color rojo, verde y azul para obtener el color.
- En CSS los componentes de los colores RGB decimal pueden tomar los valores entre 0 y 255.
- Ejemplo:

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

Unidades de medida y colores

3. RGB Porcentual

- Muy similar al RGB decimal.
- El valor de los componentes RGB puede tomar valores entre 0% y 100%.
- Ejemplo:

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

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

Unidades de medida y colores

4. RGB Hexadecimal

- Es el método más utilizado de indicar los colores.
- Casi todos los sitios web reales utilizan este método.
- Determinar las componentes RGB decimales del color original, por ejemplo: R = 71, G = 98, B = 176.
- Transformar a hexadecimal cada componente. En el ejemplo anterior: R=47, G=62, B=B0
- El color se indica con # y a continuación cada uno de los componentes: #4762B0
- Ejemplo: `p {color: #4762B0; }`
- Se pueden comprimir sus valores cuando son iguales dos a dos:

```
body{background-color: #FFF; color: #000; }  
h1, h2, h3, h4, h5, h6 {color: #C00; }
```

Unidades de medida y colores

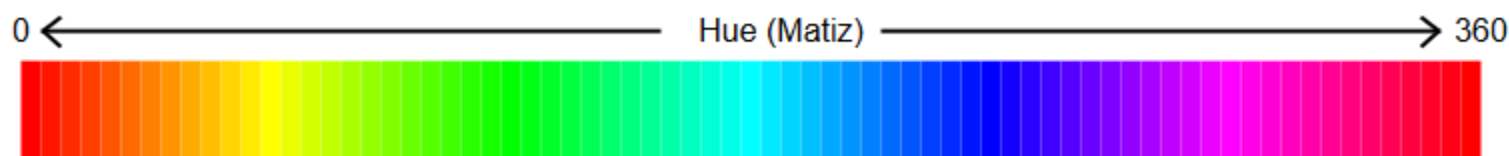
5. Colores RGBA (CSS3)

- Incorporan la transparencia alpha a los [códigos RGB](#) mediante un cuarto valor que indica la transparencia del elemento.
- La transparencia se expresa como un número decimal entre 0 y 1, en el que el 0 significa completamente transparente y el 1 completamente opaco.
- Los códigos RGBA se pueden expresar de distintas formas:
 - `rgba(rojo, verde, azul, transparencia)`
 - Colores: entre 0 y 255 o bien en porcentaje (0%-100%)
 - Transparencia: entre 0 y 1

Unidades de medida y colores

6. Colores HSL (CSS3)

- El color se define mediante tres valores:
 - **Hue** (Matiz) es un entero entre 0 y 360 y recorre todos los colores



- **Saturation** (Saturación) es un porcentaje que define la intensidad del color.
 - **Lightness** (Luminosidad) es un porcentaje que indica la claridad u oscuridad del color.
- Ejemplo:

```
p {  
  background-color: hsl(153, 80%, 40%)  
}
```

Si algo puede ir mal, irá mal.

Unidades de medida y colores

7. HSLA (CSS3)

- Incorporan la *transparencia alpha* a los códigos HSL mediante un cuarto valor que indica la transparencia del elemento.
- La transparencia se expresa como en el caso de los códigos RGBA con un número decimal entre 0 y 1, en el que el 0 significa completamente transparente y el 1 completamente opaco.
- [Colores web](#)
- <http://hslpicker.com/>
- <https://lenguajecss.com/css/colores-y-fondos/colores-css/>

2.7 Medios CSS

- Una de las características más importantes de las hojas de estilos CSS es que permiten definir diferentes estilos para distintos medios o dispositivos: monitores de ordenador, impresoras, móviles, tabletas, etc.
- CSS define algunas propiedades específicamente para determinados medios, como por ejemplo la paginación y los saltos de página para los medios impresos o el volumen y tipo de voz para los medios de audio.

Medios CSS

- La siguiente tabla muestra el nombre que CSS utiliza para identificar a cada medio y su descripción:

Medio	Descripción
all	Todos los medios definidos. Es el que se utiliza por defecto
braille	Dispositivos táctiles que emplean el sistema braille
embossed	Impresoras braille
handheld	Dispositivos móviles tradicionales (no smartphones).
print	Impresoras y navegadores en el modo <i>"Vista Previa para Imprimir"</i>
projection	Proyectores y dispositivos para presentaciones
screen	Monitores o 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

Medios CSS

- CSS permite modificar los estilos de una página en función del medio en que se visualiza.
- Formas de indicar el medio en el que aplicar los estilos CSS:
 1. Reglas de tipo @media
 2. Reglas de tipo @import
 3. Etiqueta <link>
 4. Mezcla de medios

Medios CSS

1. Reglas de tipo @media

- Permiten indicar de forma directa el medio o medios en los que se aplicarán los estilos incluidos en la regla.
- Si los estilos se aplican a varios medios, se incluyen los nombres de todos los medios separados por comas.

- Ejemplo:

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


Medios CSS

2. Reglas de tipo **@import**

- En este caso se puede especificar el medio en el que se aplican los estilos indicando el nombre del medio después de la URL del archivo CSS.

- Ejemplo:

```
1 @import url("fineprint.css") print;
2 @import url("bluish.css") projection, tv;
3 @import 'custom.css';
4 @import "common.css" screen, projection;
5 @import url('landscape.css') screen and (orientation:landscape);
```

- En el ejemplo anterior, cuando la página se visualiza por pantalla, se cargan los estilos definidos en el primer archivo CSS. Por otra parte, cuando la página se imprime, se tienen en cuenta los estilos que define el segundo archivo CSS.
- Si no se indican medios, los estilos se aplican a todos los medios (all)

Medios CSS

3. Medios mediante 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"/>
```

Medios CSS

4. Mezcla de medios

- CSS también permite mezclar los tres métodos anteriores para indicar los medios en los que se aplica cada archivo CSS externo:

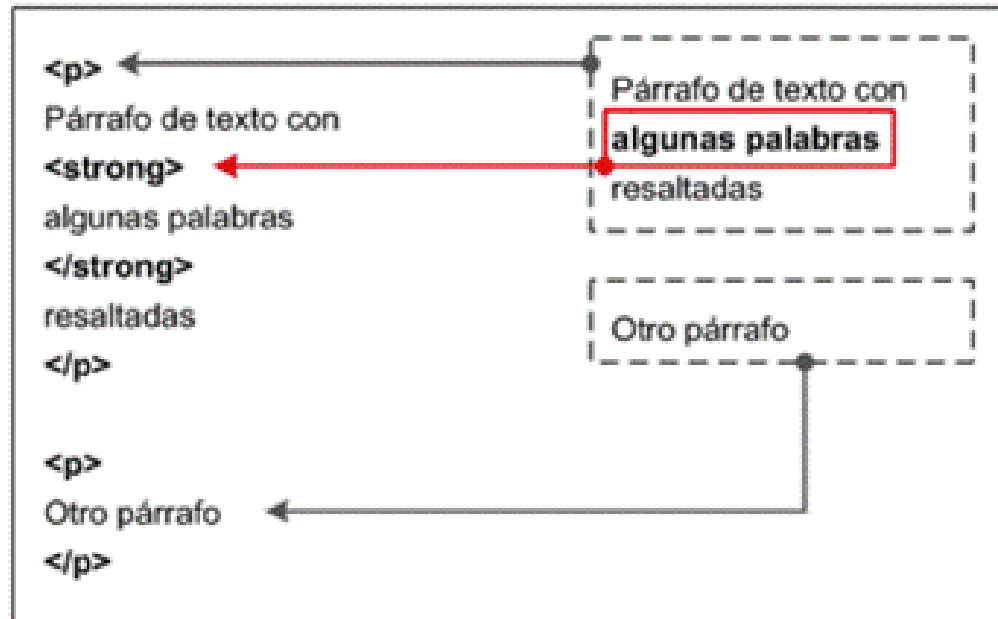
```
<link rel="stylesheet" type="text/css" media="screen" href="basico.css" />
<style>
@import url ("estilos_seccion.css") screen;
@media print {
/* Estilos específicos para impresora */
}
</style>
```

2.8 Modelo de cajas

- El modelo de cajas o "box model" es seguramente la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web.
- El modelo de cajas es el comportamiento de CSS que hace que *todos los elementos de las páginas se representen mediante cajas rectangulares*.
- Las cajas de una página se crean automáticamente, cada vez que se inserta una etiqueta
- HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento.

Modelo de cajas

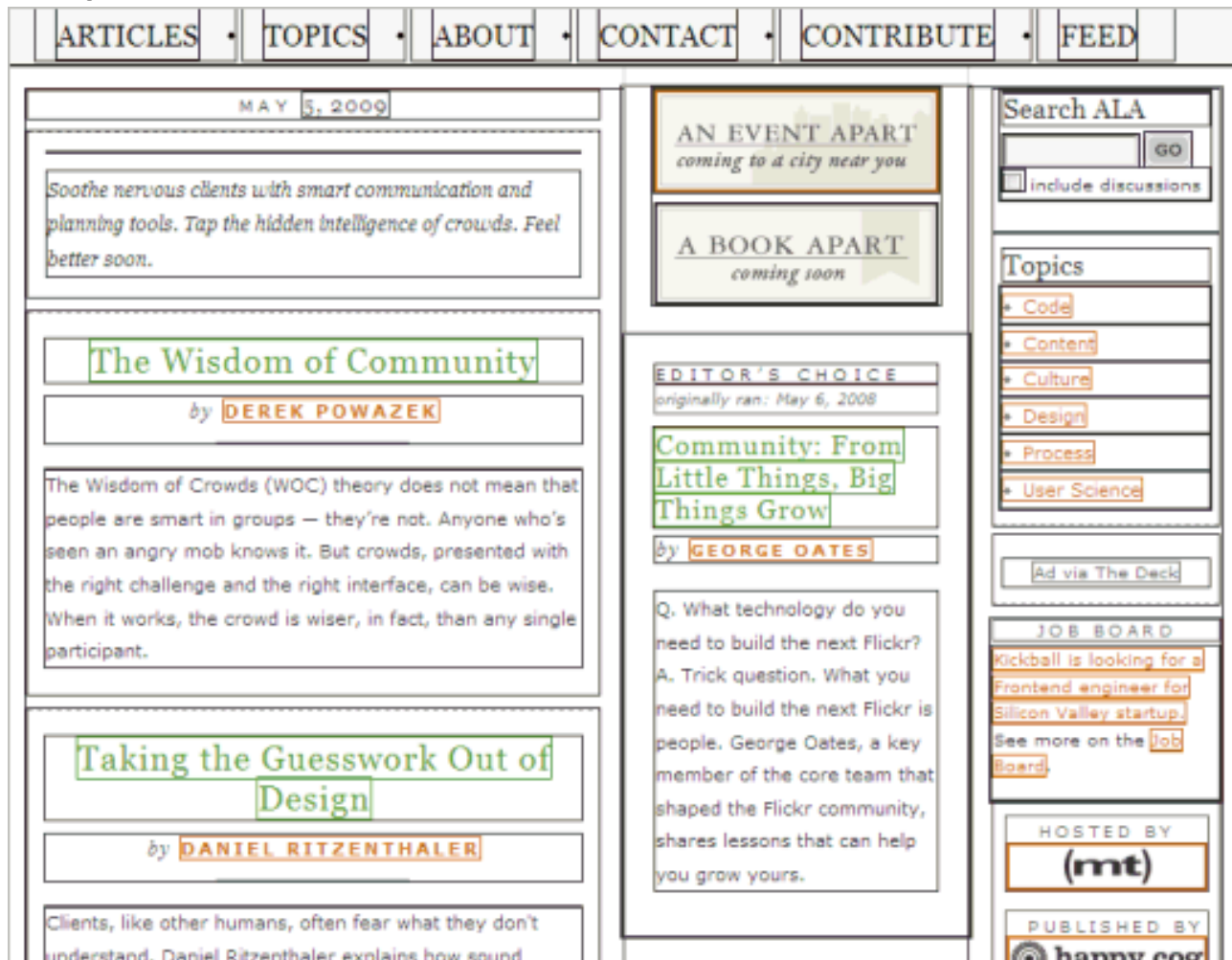
- Ejemplo:



- Cada caja se crea automáticamente al definir cada elemento HTML. No son visibles a simple vista.

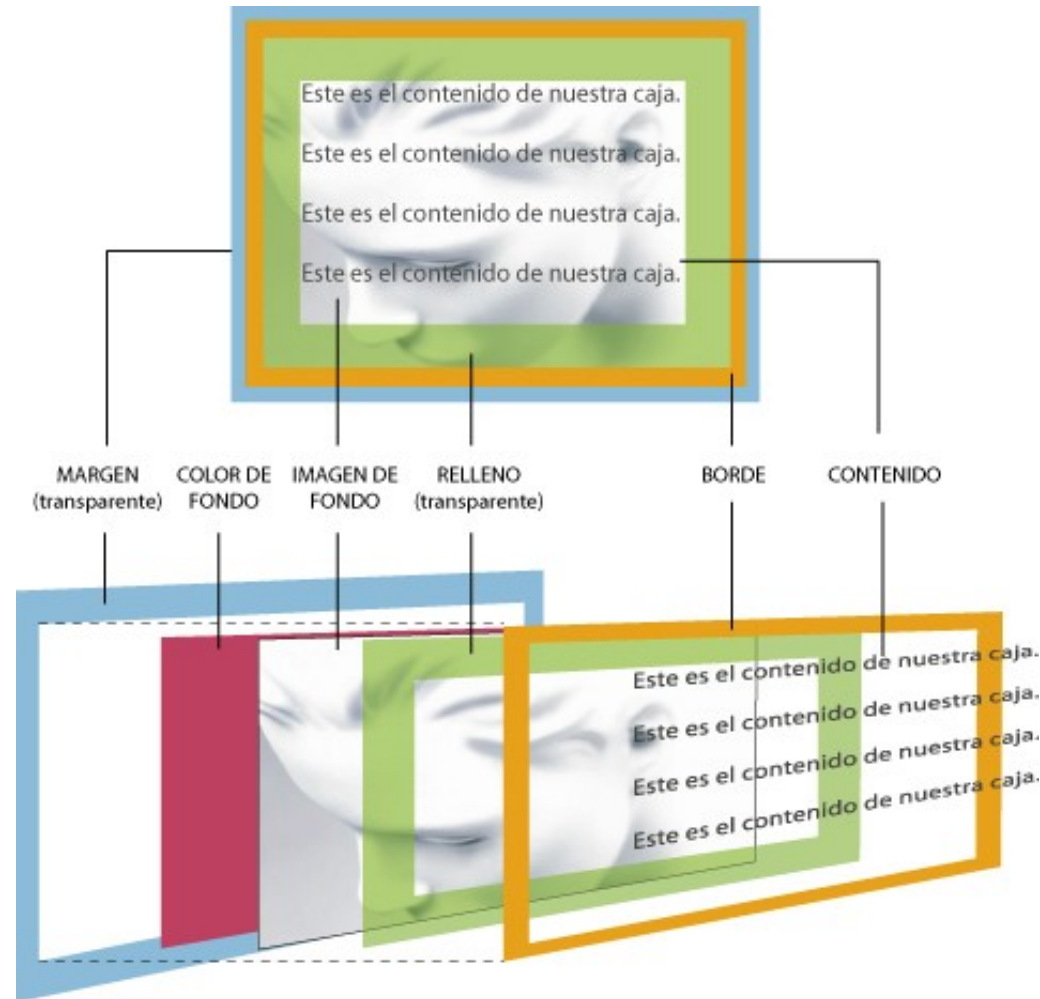
Modelo de cajas.

Ejemplo de cómo se vería



Modelo de cajas

- Los navegadores crean y colocan las cajas de forma automática, pero modificar todas sus características.
- Cada una de las cajas está formada por seis **partes**:

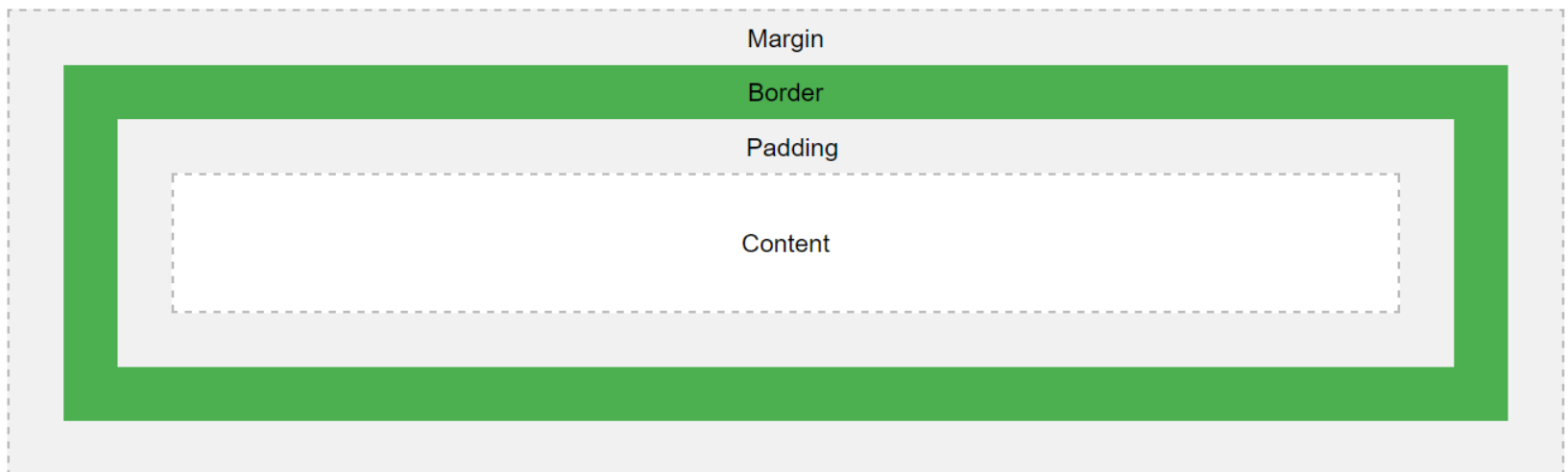


Modelo de cajas

- Las partes que componen cada caja y su orden de visualización desde el punto de vista del usuario son las siguientes:
 - **Contenido** (content): se trata del contenido HTML del elemento (párrafo, una imagen, el texto de una lista de elementos, etc.) las palabras de un
 - **Borde** (border): línea que encierra completamente el contenido y su relleno.
 - **Relleno** (padding): espacio libre opcional existente entre el contenido y el borde.
 - **Imagen de fondo** (background Image): imagen que se muestra detrás del contenido y el relleno.
 - **Color de fondo** (background color): color que muestra por detrás del contenido y el relleno.
 - **Margen** (margin): separación opcional entre la caja y el resto de elementos adyacentes.

Modelo de cajas (Box Model)

- Content (contenido)
- Padding (relleno)
- Border (borde)
- Margin (margen)



Modelo de cajas. Propiedades

- Al trabajar con un modelo de caja, debemos tener en cuenta que existen dos propiedades principales para definirla, **height** y **width**, que definirán el ancho y alto de nuestra caja.
- Los valores que tomarán pueden ser en cualquiera de las unidades de medida vistas (px, pt, %, etc).
- Ejemplo:

```
div{  
  background: #5DEAA8;  
  height: 100px;  
  width: 200px;  
}
```

Modelo de cajas. Propiedades

- Adicionalmente tenemos las propiedades **margin**, **padding** y **border** que nos brindarán mayor control al momento de organizar los elementos.

```
div{  
  border: 10px solid #3EC483;  
  background: #5DEAA8;  
  height: 100px;  
  width: 200px;  
  padding: 10px;  
  margin: 10px;  
}
```

- Los valores que pueden tomar margin y padding son: en las unidades de medida vistas, en %, inherit (valor heredado del elemento padre) o bien auto (solo margin)

Modelo de cajas

- Margen y relleno
 - Si queremos establecer valores más específicos a los márgenes, relleno y borde, podemos utilizar estas propiedades de modo más restrictivo:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

```
div {  
    width: 300px;  
    border: 25px solid green;  
    padding: 25px;  
    margin: 25px;  
}
```

- http://www.w3schools.com/css/css_margin.asp
- http://www.w3schools.com/css/css_padding.asp

Modelo de cajas

- **Bordes**

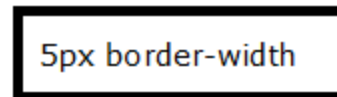
- Se establecen varias características para el borde de los elementos:
 - **Border-style** (estilo)
 - **Border-width** (ancho)
 - **Border-color** (color del trazo)
 - **Border-radius** (Bordes redondeados. CSS3)
- Se pueden definir los cuatro bordes de modo individual, o bien hacerlo en una sola regla de estilo.
- Ejemplos: http://www.w3schools.com/css/css_border.asp

Modelo de cajas. Bordes

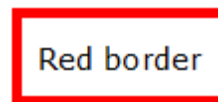
- Border-style

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

- Border-width



- Border-color



A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

```
p.one {
  border-style: solid;
  border-width: 5px;
}
```

```
p.one {
  border-style: solid;
  border-color: red;
}
```

Modelo de cajas. Bordos

- Border-radius (bordes redondeados)

Normal border

Round border

Rounder border

Roudest border

```
p.normal {  
    border: 2px solid red;  
}
```

```
p.round1 {  
    border: 2px solid red;  
    border-radius: 5px;  
}
```

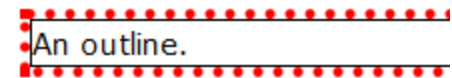
```
p.round2 {  
    border: 2px solid red;  
    border-radius: 8px;  
}
```

```
p.round3 {  
    border: 2px solid red;  
    border-radius: 12px;  
}
```

Modelo de cajas.

- Contorno (**outline**)
 - Línea exterior. Perfil.
 - Los contornos difieren de los bordes de la siguiente manera:
 - Los contornos no ocupan espacio, son dibujados por encima del contenido.
 - Los contornos pueden no ser rectangulares.
 - Propiedades:
 - **outline-style**
 - **outline-color**
 - **outline-width**

```
p {  
  border: 1px solid black;  
  outline: 5px dotted red;  
}
```



2.9 Colores y Fondos

- **Colores**

- Propiedades
 - **color**: cambia el color del texto de un elemento.
 - **background-color**: Cambia el color de fondo de un elemento.
- Todos se pueden indicar de las formas vistas.

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

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

Colores y fondos

- **Fondos**

- Puede ser un color simple o una imagen.
- El fondo solamente se visualiza en el área ocupada por el contenido y su relleno, ya que el color de los bordes se controla directamente desde los bordes y las zonas de los márgenes siempre son transparentes.
- Para establecer un color o imagen de fondo en la página entera, se debe establecer un fondo al elemento `<body>`.
- *Propiedades:*
 - **background:** propiedad compuesta para definir fondos.
 - **background-color:** establece el color de fondo de un elemento.

Colores y fondos

Pueden añadirse varios fondos separados por comas

- Fondos:

- **background-image:** establece cualquier imagen como fondo de un elemento.

```
body {  
    background-image: url("verde.jpg");  
}
```

- **background-repeat:** Por omisión, los navegadores repiten la imagen de fondo tanto en vertical como en horizontal. La propiedad background-repeat permite controlar esa repetición.

- Valores:

- **no-repeat:** la imagen no se repite
- **repeat-x:** la imagen se repite únicamente en horizontal
- **repeat-y:** la imagen se repite únicamente en vertical
- **repeat:** la imagen se repite en horizontal y vertical

CSS3

- **space:** la imagen se repite en horizontal y vertical, pero se añaden espacios entre las imágenes para mostrar las imágenes completas y que no se corten en los bordes derecho o inferior.
- **round:** la imagen se repite en horizontal y vertical, pero las imágenes se deforman para mostrarlas completas y que no se corten en los bordes derecho e inferior.

Colores y fondos

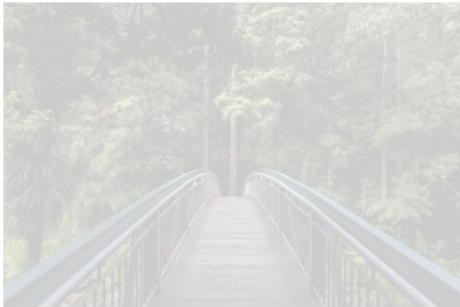
- Fondos
 - **background-position:** permite establecer la posición de la imagen de fondo.
 - A esta propiedad hay que darle dos valores: la posición horizontal y la posición vertical.
 - **Valores:**
 - valores numéricos (porcentajes o distancias)
 - **left**, **center** y **right**: establecen la posición horizontal (izquierda, centro y derecha, respectivamente)
 - **top**, **center**, **bottom**: establece la posición vertical (arriba, en medio, abajo, respectivamente)
 - **background-attachment:** establece el comportamiento de la imagen de fondo cuando se desplaza el elemento (al utilizar las barras de desplazamiento del navegador).
 - **Valores:**
 - **scroll**: la imagen acompaña al elemento cuando este se desplaza
 - **fixed**: la imagen permanece fija

Colores y fondos

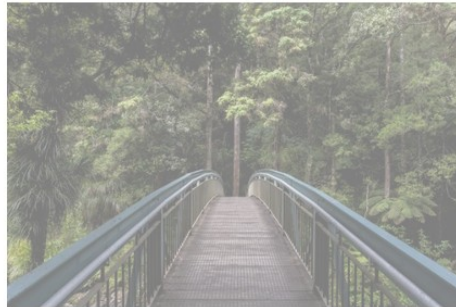
- Propiedades de fondos
 - **background-size:** permite establecer el tamaño de la imagen de fondo.
 - Valores: **auto**, **contain**, **cover**, **valores numéricos**
 - **background-clip:** permite definir la zona en la que aparece el fondo (color o imagen).
 - Valores: **border-box**, **padding-box**, **content-box**
 - **background-origin:** permite definir la zona en la que se sitúa el origen de la posición del fondo.
 - Valores: **border-box**, **padding-box**, **content-box**
- Ejemplos:
<http://www.mclibre.org/consultar/amaya/css/css-fondos.html>

Colores y fondos

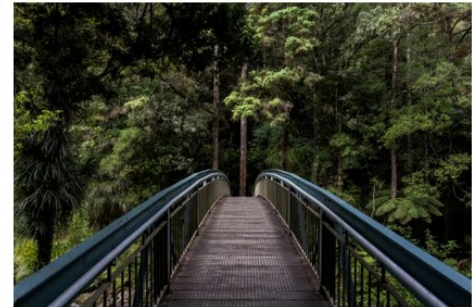
- Opacidad
 - **Opacity:** permite definir el grado de transparencia de un elemento.
 - Valores: `<valor_alpha> | inherit`
 - Valor_alpha: entre 0 y 1
 - Valor por defecto: 1
 - Ejemplo:



opacity 0.2



opacity 0.5



opacity 1
(default)

Colores y fondos

- **Gradientes (CSS3)**

- Efecto en el que un color inicial que se transforma poco a poco en uno o varios colores sucesivamente.
- Se usa en los fondos.
- Tipos:
 - Lineal
 - Radial

- **Gradientes lineales:** permite crear fondos con los colores gradientes indicados y una cierta dirección definida.

```
background: linear-gradient(direction, color-stop1, color-stop2, ...);
```

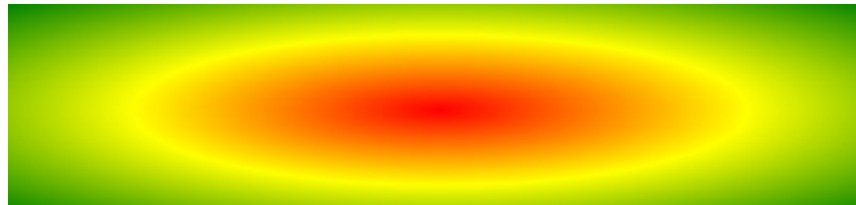
Colores y fondos

- Gradiente Radial: con forma circular.

```
background: radial-gradient(shape size at position, start-color, ..., last-color);
```

- Ejemplos

```
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-linear-gradient(red, yellow); /* For Safari 5.1 to 6.0 */  
  background: -o-linear-gradient(red, yellow); /* For Opera 11.1 to 12.0 */  
  background: -moz-linear-gradient(red, yellow); /* For Firefox 3.6 to 15 */  
  background: linear-gradient(red, yellow); /* Standard syntax */  
}  
  
#grad {  
  background: red; /* For browsers that do not support gradients */  
  background: -webkit-radial-gradient(red, yellow, green); /* Safari 5.1 to 6.0 */  
  background: -o-radial-gradient(red, yellow, green); /* For Opera 11.6 to 12.0 */  
  background: -moz-radial-gradient(red, yellow, green); /* For Firefox 3.6 to 15 */  
  background: radial-gradient(red, yellow, green); /* Standard syntax */  
}
```



2.10 Fuentes y Textos

- Las propiedades relacionadas con el tipo de letra (fuente) definidas en CSS son:
 - La propiedad compuesta: **font**
 - Tipo de letra: **font-family**
 - Tamaño del tipo de letra: **font-size**
 - Subfamilias tipográficas
 - Inclínación (itálica): **font-style**
 - Versalitas: **font-variant**
 - Grosor del trazo (negrita): **font-weight**
 - Interlineado: **line-height**
 - Ajuste del tamaño del tipo de letra: **font-size-adjust**
 - Anchura: **font-stretch**

Fuentes y Textos

- **Font**

- Permite definir simultáneamente las propiedades relacionadas con el tipo de letra: *font-style*, *font-variant*, *font-weight*, *font-size*, *line-height* y *font-family* (en este orden).
- Obligatorio: *font-size* y *font-family*. Si se escribe la propiedad *line-height* debe aparecer separada de *font-size* por una barra (/).
- Otros valores que admite: **caption**, **icon**, **menu**, **message-box**, **small-caption** y **status-bar** (*hacen referencia a los tipos de letra que utiliza el sistema operativo para esos elementos.*)

Fuentes y Textos

- **Font-family**

- Permite establecer el tipo de letra (fuente) del elemento.
- Se puede indicar la fuente concreta o bien uno de los nombres genéricos: **serif, sans-serif, cursive, fantasy o monospace** (los navegadores tienen asociada una fuente a cada nombre genérico).
- Ejemplo:

```
p {  
  font-family: "Tahoma", "Geneva", sans-serif;  
}
```

- **Font-size**

- Permite establecer el tamaño del tipo de letra (fuente) del elemento.
- Valores: *tamaño absoluto, tamaño relativo, distancia o porcentaje, aunque generalmente se aconseja utilizar unidades relativas (% o em).*

Fuentes y Textos

- Font-size:

- Tamaño absoluto: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large` y `xx-large`.
- Tamaño relativo: **larger** o **smaller**.
- Valor numérico con unidades absolutas:
- Valor numérico con unidades relativas

```
p {  
    font-size: 16pt;  
}  
p {  
    font-size: 150%;  
}
```

- Font-style

- Variante letra: inclinada
- Valores: **normal**, **oblique** o **italic**.

```
p {  
    font-family: sans-serif;  
    font-size: 200%;  
    font-style: italic;  
}
```

- Font-variant

- Letra versalitas (mayúsculas)
- Valores: **normal** y **small-caps**

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

Fuentes y Textos

- **Font-weight**

- Permite elegir el grosor del trazo.
- Valores:
 - numéricos (**100, 200, 300, 400, 500, 600, 700, 800, 900**), del más fino al más grueso.
 - **Normal** (valor 400)
 - **Bold** (valor 700)
 - **Lighter** (valor inferior en la lista de valores numéricos)
 - **Bolder** (valor superior en la lista de valores numéricos)

```
p {  
  font-weight: normal;  
}  
  
ul {  
  font-weight: bolder;  
}
```

Este es un párrafo normal

- Lista con trazo más grueso
 - Lista con trazo todavía más grueso
 - Lista con trazo todavía más grueso
 - Lista con trazo todavía más grueso

Fuentes y Textos

- **Font-size-adjust**

- Permite establecer la proporción entre el tamaño del tipo y el de la letra x, indicándolo como número decimal.

```
p {  
  font-size: 16px;  
  font-size-adjust: 1.5;  
}
```

Esta letra x tiene $16 \times 1.5 = 24$ px de altura.

- **Font-stretch**

- Permite elegir que un tipo de letra más o menos condensado o expandido.
- Valores: **ultra-condensed, extra-condensed, condensed, semi-condensed, normal, semi-expanded, expanded, extra-expanded, ultra-expanded.**

Fuentes y Textos

- **Fuentes de iconos**

- Las fuentes de iconos son un tipo de fuentes (tipos de letra) en los que cada carácter es un icono que se han hecho populares en los últimos años.
- Si a una etiqueta se le asocia la fuente que contiene los iconos, se muestran los iconos correspondientes. Con códigos Unicode.
- Las fuentes de iconos se distribuyen con **hojas de estilo** que contienen todas las clases correspondientes a todos los caracteres para que sólo haya que enlazar la hoja de estilo para poder utilizar los caracteres.

Fuentes y Textos

- Fuentes de iconos. Ejemplo

```
...  
  <link rel="stylesheet" type="text/css" href="css/font-awesome.css" />  
  <link rel="stylesheet" type="text/css" href="css/estilo.css" />  
</head>  
<body>  
<p><i class="fa fa-check-circle"></i> Me gusta</p>  
  
<p><i class="fa fa-times-circle"></i> No me gusta</p>  
</body>
```

✓ Me gusta

✗ No me gusta

Fuentes y Textos

- Fuentes de iconos

- Recopilación de sitios web con iconos:
<https://css-tricks.com/flat-icons-icon-fonts/>
- Font Awesome:
<http://fontawesome.io/> - [Font Awesome releases](#)
- Weather icons:
<http://erikflowers.github.io/weather-icons/> -
[Descargar weather icons](#)
- Explicación sobre el uso de fuentes de iconos:
<http://gomakethings.com/icon-fonts/>
- Otras webs de iconos:
 - IconMonster: <http://iconmonstr.com/>
 - freepik: <http://www.freepik.com/>
 - IconFinder: <https://www.iconfinder.com/>
 - IconArchive: <http://www.iconarchive.com/commercialfree.html>

Fuentes y Textos

- La regla **@font-face**
 - Permite utilizar fuentes que no están instaladas en el equipo local.
 - CSS3
 - Formatos: TTF, OTF, WOFF, WOFF 2.0, SVG Fonts, EOT
 - Ejemplo:

```
@font-face {  
    font-family: myFirstFont;  
    src: url(sansation_light.woff);  
}  
  
div {  
    font-family: myFirstFont;  
}
```

Fuentes y Textos

- **Google Fonts**

- Google ofrece un servicio de alojamiento de fuentes libres, [Google Fonts](https://fonts.google.com/).
- Google Fonts: las fuentes pueden utilizarse en nuestras páginas web sin necesidad de alojarlas en nuestro propio servidor. También se pueden descargar.
- 2 formas de incluirlas en nuestras páginas:

```
@import url(https://fonts.googleapis.com/css?family=Mystery+Quest);
```

```
p {  
  font-family: "Mystery Quest", sans-serif;  
  font-size: 150%;  
}
```

```
<head>
```

```
...
```

```
<link rel="stylesheet" type="text/css"
```

```
href="https://fonts.googleapis.com/css?family=Mystery+Quest" />
```

```
...
```

```
</head>
```

En la hoja de
estilo CSS

En página
HTML

Fuentes y Textos

Propiedades para **textos**

Propiedad CSS	Valores posibles	Significado
text-align	Left, right, center y justify	Alineación horizontal
vertical-align	baseline, sub, super, top, text-top, middle, bottom y text-bottom	Alineación vertical
text-decoration	none (ninguno), underline, overline (sobrerayado) y line-through (tachado)	Decoración del texto
text-transform	none, capitalize, uppercase y lowercase	Mayúsculas / minúsculas
text-shadow	Color dist_hor dist_ver diámetro	Sombreado del texto
text-overflow	clip ellipsis <i>texto</i>	Comportamiento cuando el texto no cabe

Fuentes y Textos

Propiedades para **textos**

Propiedad CSS	Valores posibles	Significado
letter-spacing	normal tamaño	Espacio entre letras (tracking)
line-height	normal tamaño	Espacio entre líneas (interlineado)
text-indent	tamaño	Indentación de texto (sangría)
word-spacing	normal tamaño	Espacio entre palabras
white-space	normal nowrap pre pre-line pre-wrap	Comportamiento de los espacios
tab-size	número de espacios tamaño	Ancho de las tabulaciones
direction	ltr rtl	Dirección del texto

Fuentes y Textos

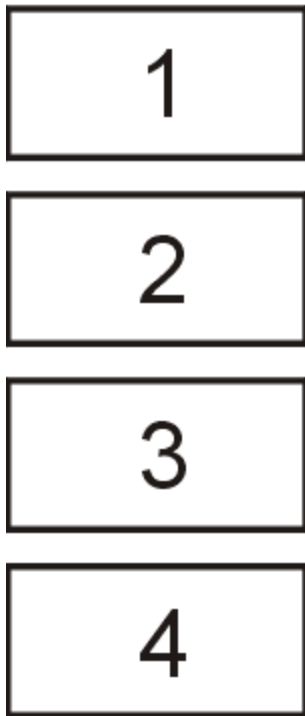
- Ejemplos de las propiedades de textos:
 - http://www.mclibre.org/consultar/htmlcss/css/css_texto.html
 - CSS3:
http://www.mclibre.org/consultar/htmlcss/css/css3_text_decoration.html

2.11 Posicionamiento y visualización

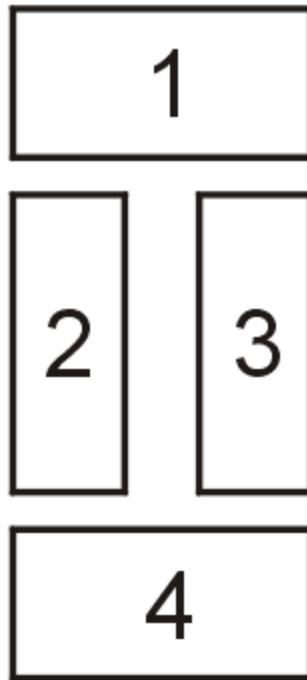
- Los elementos de una página web están contenidos en una caja rectangular, de acuerdo con el [modelo de caja](#).
- La manera en que los diferentes elementos de una página web se distribuyen en la pantalla dependen del esquema de posicionamiento elegido.
- Existen tres esquemas de posicionamiento:
 - **normal** (por defecto)
 - **flotante** (elemento no ocupa todo el ancho, se coloca a la izda o la dcha de otros elementos)
 - **absoluto** (en cualquier posición)

Posicionamiento y visualización

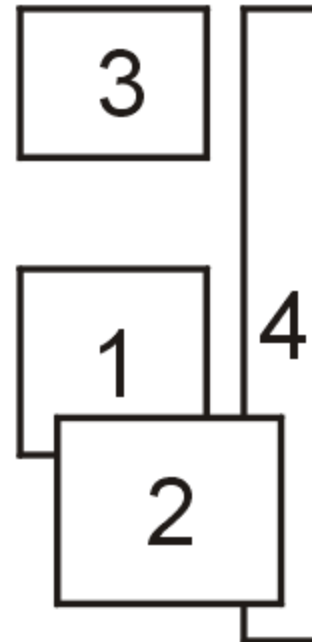
Esquema normal



Esquema flotante



Esquema absoluto



Posicionamiento y visualización

- **Posicionamiento Flotante. Propiedades:**

- **float**

- Establece el esquema de posicionamiento flotante para un elemento. Se puede aplicar a cualquier elemento.
 - Valores: **left** y **right**
 - Esta propiedad no permite centrar -> `text-align: center`

- **clear**

- Permite que un elemento no tenga elementos flotantes a su lado.
 - Valores:
 - **left**, hace que no haya elementos flotantes a la izquierda,
 - **right**, hace que no haya elementos flotantes a la derecha,
 - **both**, hace que no haya elementos flotantes ni a derecha ni a izquierda,
 - **none**, permite que haya elementos flotantes a derecha y a izquierda (valor por omisión).

Posicionamiento y visualización

- **Posicionamiento Flotante**

- Tamaño de los elementos flotantes
 - Hay que establecerlo con las propiedades **width** y **height**
 - Ejemplo:

```
p {  
  border: black 2px solid;  
  float: left;  
  height: 100px;  
  margin: 3px;  
  padding: 3px;  
  width: 100px;  
}
```

Esto es el primer párrafo	Esto es el segundo párrafo	Esto es el tercer párrafo	Esto es el cuarto párrafo	Esto es el quinto párrafo	Esto es el sexto párrafo
------------------------------	----------------------------------	------------------------------	------------------------------	------------------------------	-----------------------------

Posicionamiento y visualización

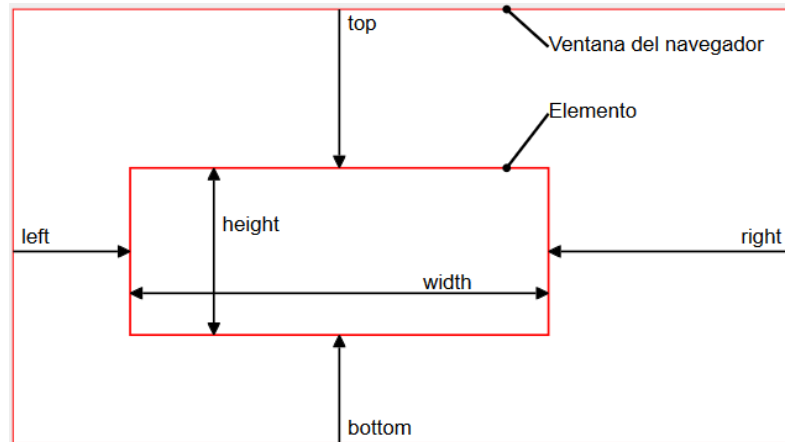
- **Posicionamiento absoluto.** Propiedades:

- **position**

- Establece el esquema de posicionamiento absoluto de un elemento.
 - 4 esquemas de posicionamiento absoluto:
 - **static**: posicionamiento absoluto estático (equivalente al esquema normal).
 - **relative**: posicionamiento absoluto relativo (se reserva espacio para el elemento, pero se puede colocar)
 - Desplazamiento: **top**, **bottom**, **left** y **right**.
 - **fixed**: posicionamiento absoluto fijo (el elemento se coloca donde se quiera, permanece fijo en scroll)
 - **absolute**: posicionamiento absoluto absoluto (el elemento se coloca en cualquier posición de la pantalla)
 - Posición: **top**, **bottom**, **left** y **right**. (para fixed y absolute)
 - **Sticky**: elemento fijo alterna entre relative y fixed, según la posición de desplazamiento. Se coloca en relación hasta que se alcanza una posición de desplazamiento determinada en la ventana gráfica

Posicionamiento y visualización

- **Posicionamiento absoluto.**



- **top**: posición del lado superior del elemento contando desde el borde superior de la ventana (de arriba a abajo, es decir, cuanto más grande es el valor más hacia abajo se coloca el elemento).
 - **bottom**: posición del lado inferior del elemento contando desde el borde inferior de la ventana (de abajo a arriba, es decir, cuanto más grande es el valor más hacia arriba se coloca el elemento).
 - **height**: altura del elemento.
 - **left**: posición del lado izquierdo del elemento contando desde el borde izquierdo de la ventana (de izquierda a derecha, es decir, cuanto más grande es el valor más hacia la derecha se coloca el elemento).
 - **right**: posición del lado derecho del elemento contando desde el borde derecho de la ventana (de izquierda a derecha, es decir, cuanto más grande es el valor más hacia la izquierda se coloca el elemento).
 - **width**: anchura del elemento.
- Ejemplo: <https://www.mclibre.org/consultar/htmlcss/css/css-posicionamiento-absoluto.html>

Posicionamiento y visualización

- **Visualización**

- CSS define cuatro propiedades para controlar el modo de visualización de los elementos:
 - **display**
 - **visibility**
 - **overflow**
 - **z-index**
- Utilizando algunas de estas propiedades es posible ocultar y/o hacer invisibles las cajas de los elementos, por lo que son importantes para realizar efectos avanzados y animaciones.

Posicionamiento y visualización

- **display**

- Permite controlar la forma de visualizar un elemento u ocultar completamente el mismo, haciendo que desaparezca de la página.
- Como el elemento oculto no se muestra, el resto de elementos de la página se mueven para ocupar su lugar.
- Se aplica a cualquier elemento
- Valores: inline | block | none | list-item | run-in | inline-block | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | inherit
- Valor por defecto: inline
- Ejemplo de utilización: *la propiedad **display: inline** se puede utilizar en las listas (,) que se quieren mostrar horizontalmente y la propiedad **display: block** se emplea frecuentemente para los enlaces que forman el menú de navegación*

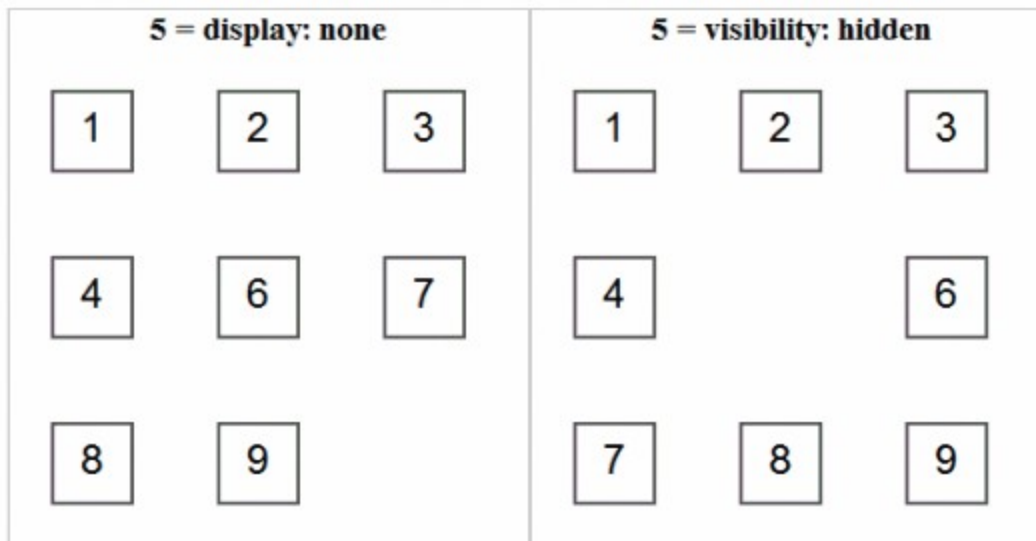
Posicionamiento y visualización

- **visibility**

- Permite hacer visible o invisible un elemento de la página.
- Si se hace invisible, el navegador crea la caja del elemento pero no la muestra.
- En este caso, el resto de elementos de la página no modifican su posición, ya que aunque la caja no se ve, sigue ocupando sitio.
- Se aplica a todos los elementos
 - Valores: **visible** | **hidden** | **collapse** | **inherit**
- Valor por defecto: visible
- El valor **collapse** de la propiedad visibility sólo se puede utilizar en las filas, grupos de filas, columnas y grupos de columnas de una tabla.

Posicionamiento y visualización

- Ejemplo: diferencia entre ocultar una caja (5) mediante las propiedades **display** o **visibility**



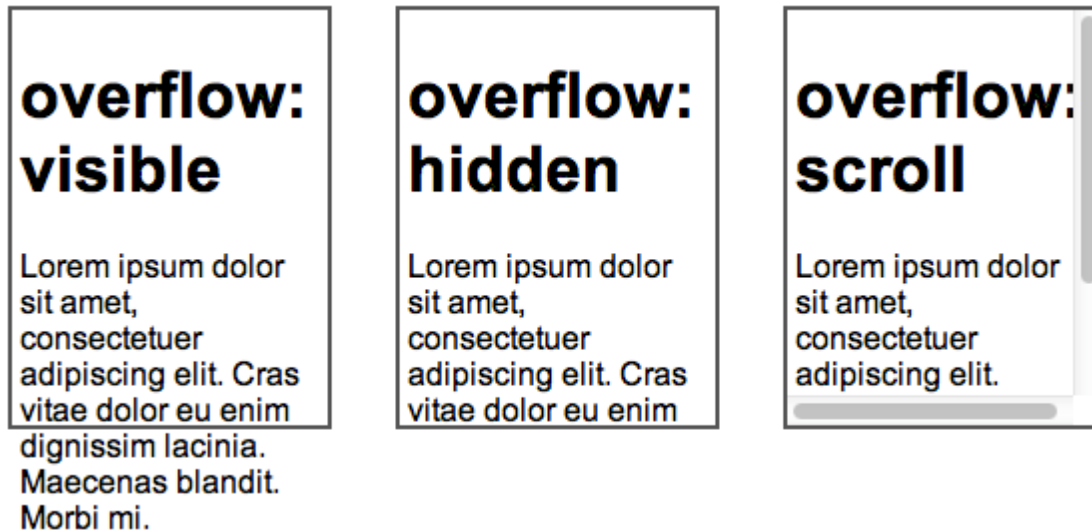
Posicionamiento y visualización

- **overflow**

- Normalmente, los contenidos de un elemento se pueden mostrar en el espacio reservado para ese elemento. Sin embargo, en algunas ocasiones el contenido de un elemento no cabe en el espacio reservado para ese elemento y se desborda.
- CSS define la propiedad **overflow** para controlar la forma en la que se visualizan los contenidos que sobresalen de sus elementos.
- Se aplica a elementos de bloque y celdas de tablas.
 - Valores **visible** | **hidden** | **scroll** | **auto** | **inherit**
- Valor por defecto: visible

Posicionamiento y visualización

- Overflow. Ejemplo



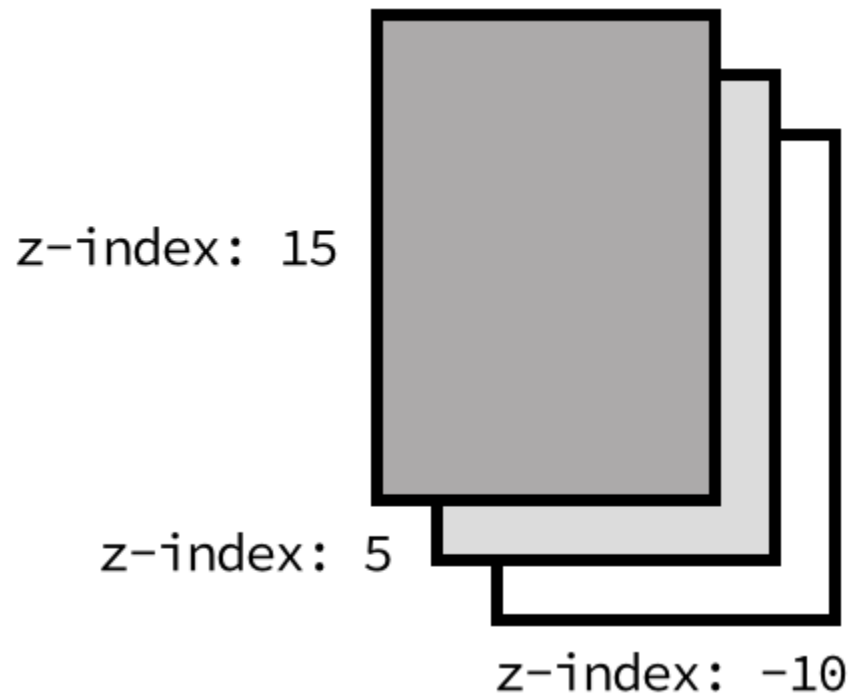
Posicionamiento y visualización

- **z-index**

- CSS permite controlar la posición tridimensional de las cajas posicionadas. De esta forma, es posible indicar las cajas que se muestran delante o detrás de otras cajas cuando se producen solapamientos.
- La posición tridimensional de un elemento se establece sobre un tercer eje llamado Z y se controla mediante la propiedad **z-index**.
- Se aplica a elementos con *posicionamiento absoluto* (mediante la propiedad **position**).
- Valores: **auto** | **número** | **inherit**
- Valor inicial: auto
- A mayor número, el elemento se posiciona encima.

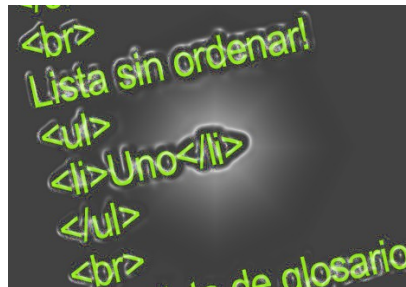
Posicionamiento y visualización

- **z-index.** Ejemplo



2.12 Propiedades de Listas

- Las propiedades CSS que permiten modificar el aspecto de los elementos de lista son:
 - **list-style**
 - **list-style-type**
 - **list-style-image**
 - **list-style-position**



Listas

- **list-style**

- Permite establecer en una sola propiedad los valores de las propiedades **list-style-type**, **list-style-image** y **list-style-position**.
- El orden de las propiedades en este caso no influye. El aconsejado:

```
list-style: <type> <position> <image>
```

- Ejemplo:

```
div {  
    list-style: circle inside none;  
}
```

Listas

- **list-style-type**

- Permite establecer el tipo de viñeta mostrada para una lista.

Propiedad	Valor	Significado
list-style-type	disc circle square none	Viñetas sin orden
list-style-type	decimal decimal-leading-zero lower-roman upper-roman	Viñetas numéricas
list-style-type	lower-alpha upper-alpha lower-greek	Viñetas alfabéticas

- Valor por defecto: disc
- Cuando no se quiere mostrar viñeta o número: **none**

Listas

- list-style-type. Ejemplo

- list-style-type: disc
- Elemento
- Elemento

- list-style-type: circle
- Elemento
- Elemento

- list-style-type: square
- Elemento
- Elemento

- i. list-style-type: lower-roman
- ii. Elemento
- iii. Elemento

- α. list-style-type: lower-greek
- β. Elemento
- γ. Elemento

- A. list-style-type: upper-latin
- B. Elemento
- C. Elemento

- a. list-style-type: lower-latin
- b. Elemento
- c. Elemento

- Ա. list-style-type: armenian
- Բ. Elemento
- Գ. Elemento

- A. list-style-type: upper-alpha
- B. Elemento
- C. Elemento

- 01. list-style-type: decimal-leading-zero
- 02. Elemento
- 03. Elemento

```
<ul style="list-style-type: square">  
  <li>list-style-type: square</li>  
  <li>Elemento</li>  
  <li>Elemento</li>  
</ul>
```


Listas

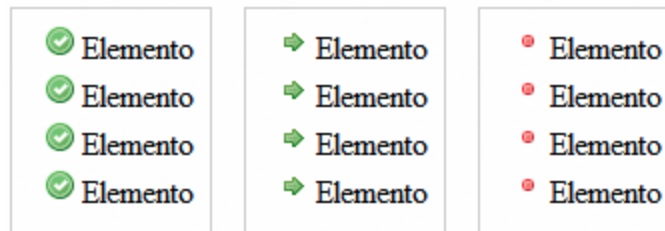
- **list-style-image**

- Permite reemplazar las viñetas automáticas por una imagen personalizada.

- **Valores** **url | none | inherit**

- Valor por defecto: none

- Ejemplo:



```
ul.ok { list-style-image: url("imagenes/ok.png"); }  
ul.flecha { list-style-image: url("imagenes/flecha.png"); }  
ul.circulo { list-style-image: url("imagenes/circulo_rojo.png"); }
```

Listas

- **list-style-position**

- Permite establecer la posición de la viñeta de cada elemento de una lista.

- Valores **inside | outside | inherit**

- Valor por defecto: outside

- Ejemplo:

■ **list-style-position: outside**
■ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc in diam. Praesent a justo. Nam odio. Quisque a libero vel massa malesuada scelerisque. Curabitur metus.
■ Nunc lobortis tortor. Etiam nec nibh. In tincidunt urna ut erat. Integer velit ante, tempus ut, egestas convallis, euismod in, erat.

■ **list-style-position: inside**
■ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc in diam. Praesent a justo. Nam odio. Quisque a libero vel massa malesuada scelerisque. Curabitur metus.
■ Nunc lobortis tortor. Etiam nec nibh. In tincidunt urna ut erat. Integer velit ante, tempus ut, egestas convallis, euismod in, erat.

Listas. Utilidades

- **Menú vertical con listas**

- Una buena técnica para el diseño de sitios web es emplear las listas de elementos para crear los menús de navegación.
- *Ejemplos:*

```
<ul>  
  <li><a href="#">Elemento 1</a></li>  
  <li><a href="#">Elemento 2</a></li>  
  <li><a href="#">Elemento 3</a></li>  
  <li><a href="#">Elemento 4</a></li>  
  <li><a href="#">Elemento 5</a></li>  
  <li><a href="#">Elemento 6</a></li>  
</ul>
```

Lista de enlaces inicial



Elemento 1
Elemento 2
Elemento 3
Elemento 4
Elemento 5
Elemento 6

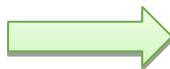
Menú vertical final

Listas. Utilidades

- **Menú vertical con listas**

- **Paso 1:** definir el ancho del menú

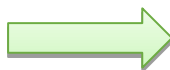
```
ul.menu { width: 180px; }
```



- [Elemento 1](#)
- [Elemento 2](#)
- [Elemento 3](#)
- [Elemento 4](#)
- [Elemento 5](#)
- [Elemento 6](#)

- **Paso 2:** eliminar viñetas, márgenes y espacios

```
ul.menu {  
    list-style: none;  
    margin: 0;  
    padding: 0;  
    width: 180px;  
}
```



[Elemento 1](#)
[Elemento 2](#)
[Elemento 3](#)
[Elemento 4](#)
[Elemento 5](#)
[Elemento 6](#)

Listas. Utilidades

- **Menú vertical con listas**
 - **Paso 3:** Añadir un borde al menú de navegación y establecer el color de fondo y los bordes de cada elemento del menú:

```
ul.menu {  
    border: 1px solid #7C7C7C;  
    border-bottom: none;  
    list-style: none;  
    margin: 0;  
    padding: 0;  
    width: 180px;  
}  
ul.menu li {  
    background: #F4F4F4;  
    border-bottom: 1px solid #7C7C7C;  
    border-top: 1px solid #FFF;  
}
```



Elemento 1
Elemento 2
Elemento 3
Elemento 4
Elemento 5
Elemento 6

Listas. Utilidades

- **Menú vertical con listas**

- **Paso 4:** Aplicar estilos a los enlaces: mostrarlos como un elemento de bloque para que ocupen todo el espacio de cada `` del menú, añadir un espacio de relleno y modificar los colores y la decoración por defecto

```
ul.menu li a {  
    color: #333;  
    display: block;  
    padding: .2em 0 .2em .5em;  
    text-decoration: none;  
}
```



Elemento 1
Elemento 2
Elemento 3
Elemento 4
Elemento 5
Elemento 6

Aspecto final del menú vertical

Listas. Utilidades

- **Menú horizontal básico**

- Partiendo de una lista de elementos, la clave está en modificar el posicionamiento original de los elementos de la lista.

Código HTML del menú horizontal:

```
<ul>
  <li><a href="#">Elemento 1</a></li>
  <li><a href="#">Elemento 2</a></li>
  <li><a href="#">Elemento 3</a></li>
  <li><a href="#">Elemento 4</a></li>
  <li><a href="#">Elemento 5</a></li>
</ul>
```



Aspecto final del menú horizontal:

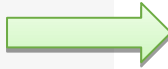
Elemento 1	Elemento 2	Elemento 3	Elemento 4	Elemento 5
------------	------------	------------	------------	------------

Listas. Utilidades

- **Menú horizontal básico**

- **Paso 1:** Aplicar los estilos CSS básicos para establecer el estilo del menú (similares a los del menú vertical anterior):

```
ul.menu {  
    background: #F4F4F4;  
    border: 1px solid #7C7C7C;  
    list-style: none;  
    margin: 0;  
    padding: 0;  
}  
  
ul.menu li a {  
    color: #333;  
    display: block;  
    padding: .3em;  
    text-decoration: none;  
}
```



Elemento 1
Elemento 2
Elemento 3
Elemento 4
Elemento 5

Listas. Utilidades

- **Menú horizontal básico**

- **Paso 2:** Establecer la anchura de los elementos del menú. En este caso se asigna un ancho de 20% a cada elemento. Se posiciona de modo flotante.

```
ul.menu li {  
    float: left;  
    width: 20%;  
}
```



Elemento 1	Elemento 2	Elemento 3	Elemento 4	Elemento 5
------------	------------	------------	------------	------------

```
ul.menu {  
    overflow: hidden;  
}
```

Listas. Utilidades

- **Menú horizontal básico**

- **Paso 3:** Establecer los bordes de los elementos que forman el menú.

```
ul.menu li a {  
    border-left: 1px solid #FFF;  
    border-right: 1px solid #7C7C7C;  
}
```

Elemento 1	Elemento 2	Elemento 3	Elemento 4	Elemento 5
------------	------------	------------	------------	------------

- **Paso 4:** se elimina el borde derecho del último elemento de la lista, para evitar el borde duplicado.

```
<ul>  
    <li><a href="#">Elemento 1</a></li>  
    <li><a href="#">Elemento 2</a></li>  
    <li><a href="#">Elemento 3</a></li>  
    <li><a href="#">Elemento 4</a></li>  
    <li><a href="#" style="border-right: none">Elemento 5</a></li>  
</ul>
```



Aspecto final del menú horizontal

Elemento 1	Elemento 2	Elemento 3	Elemento 4	Elemento 5
------------	------------	------------	------------	------------

2.13 Propiedades de tablas

- Propiedades de estilo aplicables a tablas en CSS:
 - **border-collapse**
 - **border-spacing**
 - **caption-side**
 - **table-layout**
 - **empty-cells**



SO más utilizados en Vida MRR

Sistema Operativo	Vistas	Porcentaje
Windows	20.000	79,58%
Mac OS X	3.000	12,50%
Linux	700	2,96%
iOS	630	2,82%
Android	345	1,54%
Blackberry	46	0,21%
Windows Phone	46	0,12%
PlayStation 3	4	0,01%

Tablas

- **border-collapse**

- Permite elegir el modo de presentación de los bordes de las celdas y de la tabla.
 - Valores **initial** | **collapse** | **separate** | **inherit**
- Valor por defecto: separate
- Ejemplo:

A	B	C	D	E
a	1	2	3	4
b	1	2	3	4
c	1	2	3	4
d	1	2	3	4

Borde por defecto

A	B	C	D	E
a	1	2	3	4
b	1	2	3	4
c	1	2	3	4
d	1	2	3	4

border-collapse: collapse

Tablas

- **border-spacing**

- Permite establecer una separación entre las celdas cuando se utiliza el modo de bordes separado.

- Valores **unidad de medida | inherit**

- Ejemplo:

These cells	have 5px of spacing	all around them
-------------	---------------------	-----------------

These cells	have 5px of horizontal spacing	and 10px vertical
-------------	--------------------------------	-------------------

```
.horizontal {  
    border-collapse: separate;  
    border-spacing: 5px;  
    border: 1px solid #000;  
}  
  
.both {  
    border-collapse: separate;  
    border-spacing: 5px 10px;  
    border: 1px solid #000;  
}
```

Tablas

- **caption-side**

- Permite elegir la posición de la leyenda (<caption>) con respecto a la tabla.
- **Valores:** **top** (arriba), **right** (derecha), **bottom** (abajo) y **left** (izquierda)
- Ejemplos:

<pre>caption { }</pre>	<div>Esto es la leyenda</div> <table><tr><td>Celda 1</td><td>Celda 2</td></tr><tr><td>Celda 3</td><td>Celda 4</td></tr></table>	Celda 1	Celda 2	Celda 3	Celda 4
Celda 1	Celda 2				
Celda 3	Celda 4				
<pre>caption { caption-side: top; }</pre>	<div>Esto es la leyenda</div> <table><tr><td>Celda 1</td><td>Celda 2</td></tr><tr><td>Celda 3</td><td>Celda 4</td></tr></table>	Celda 1	Celda 2	Celda 3	Celda 4
Celda 1	Celda 2				
Celda 3	Celda 4				
<pre>caption { caption-side: bottom; }</pre>	<table><tr><td>Celda 1</td><td>Celda 2</td></tr><tr><td>Celda 3</td><td>Celda 4</td></tr></table> <div>Esto es la leyenda</div>	Celda 1	Celda 2	Celda 3	Celda 4
Celda 1	Celda 2				
Celda 3	Celda 4				

<pre>caption { caption-side: left; }</pre>	<div>Esto es la leyenda</div> <table><tr><td>Celda 1</td><td>Celda 2</td></tr><tr><td>Celda 3</td><td>Celda 4</td></tr></table>	Celda 1	Celda 2	Celda 3	Celda 4
Celda 1	Celda 2				
Celda 3	Celda 4				
<pre>caption { caption-side: right; }</pre>	<table><tr><td>Celda 1</td><td>Celda 2</td></tr><tr><td>Celda 3</td><td>Celda 4</td></tr></table> <div>Esto es la leyenda</div>	Celda 1	Celda 2	Celda 3	Celda 4
Celda 1	Celda 2				
Celda 3	Celda 4				

Tablas

- **table-layout**

- Permite elegir el modo en el que el navegador calcula el tamaño total de la tabla y el de cada fila o columna.
- **Valores:** **auto** | **fixed** | **initial** | **inherit**
- En el modo fijo (fixed), el tamaño de las tablas y de las celdas depende de las propiedades width y height.
- En el modo fijo (auto), el tamaño de las tablas y de las celdas depende de su contenido (y si es posible, de las propiedades width y height).

Tablas

- **empty-cells**

- Permite establecer si se muestran o no los bordes de las celdas vacías, en el modo de bordes separado.
- **Valores:** **show** | **hide** | **initial** | **inherit**
- Si no se establece la propiedad, los navegadores aplican el valor show.
- Ejemplo:

<pre>table { }</pre>	<div>Esto es la leyenda</div> <table><tr><td>Celda 1</td><td></td></tr><tr><td></td><td>Celda 4</td></tr></table>	Celda 1			Celda 4
Celda 1					
	Celda 4				
<pre>table { empty-cells: show; }</pre>	<div>Esto es la leyenda</div> <table><tr><td>Celda 1</td><td></td></tr><tr><td></td><td>Celda 4</td></tr></table>	Celda 1			Celda 4
Celda 1					
	Celda 4				
<pre>table { empty-cells: hide; }</pre>	<div>Esto es la leyenda</div> <table><tr><td>Celda 1</td><td></td></tr><tr><td></td><td>Celda 4</td></tr></table>	Celda 1			Celda 4
Celda 1					
	Celda 4				

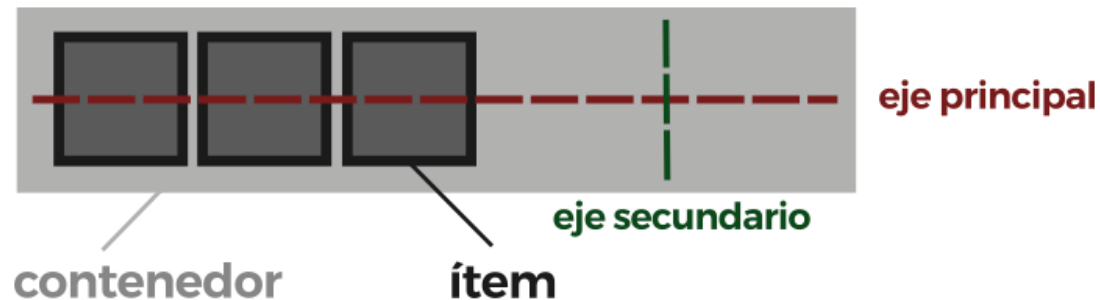
2.14 Otras propiedades CSS3

- FlexBox (cajas flexibles)
- Box-sizing
- Filtros CSS (efectos de imagen)
- Cursor



FlexBox CSS3 (cajas flexibles)

- Es un sistema de **elementos flexibles** en la que los elementos HTML se adaptan y colocan automáticamente y es más fácil personalizar los diseños.
- Elementos de este esquema:
 - Contenedor
 - Ítem
 - Eje principal
 - Eje secundario



FlexBox

- Para utilizar este modo de colocar los elementos en la página se emplea la propiedad **display**, con los valores:

inline-flex	Establece un contenedor de ítems flexible en línea, de forma equivalente a inline-block.
flex	Establece un contenedor de ítems flexible en bloque, de forma equivalente a block.

De esta manera los elementos no se desbordarán ni tendrán problemas con los porcentajes.

- En HTML:

```
<div id="contenedor"> <!-- contenedor flex -->
  <div class="item item-1">1</div> <!-- cada uno de los ítems flexibles -->
  <div class="item item-2">2</div>
  <div class="item item-3">3</div>
</div>
```

FlexBox

- **Dirección de los ejes**

Propiedad	Valor	Significado
flex-direction	row row-reverse column column-reverse	Cambia la orientación del eje principal.
flex-wrap	nowrap wrap wrap-reverse	Evita o permite el desbordamiento (multilínea).

- flex-flow: <flex-direction> <flex-wrap>

- **Ejemplo:**

```
#contenedor {  
  background: #CCC;  
  display: flex;  
  flex-direction: column;  
}  
  
.item {  
  background: #777;  
}
```

```
#contenedor {  
  background: #CCC;  
  display: flex;  
  width: 200px;  
  flex-wrap: wrap;  
}  
  
.item {  
  background: #777;  
  width: 50%;  
}
```

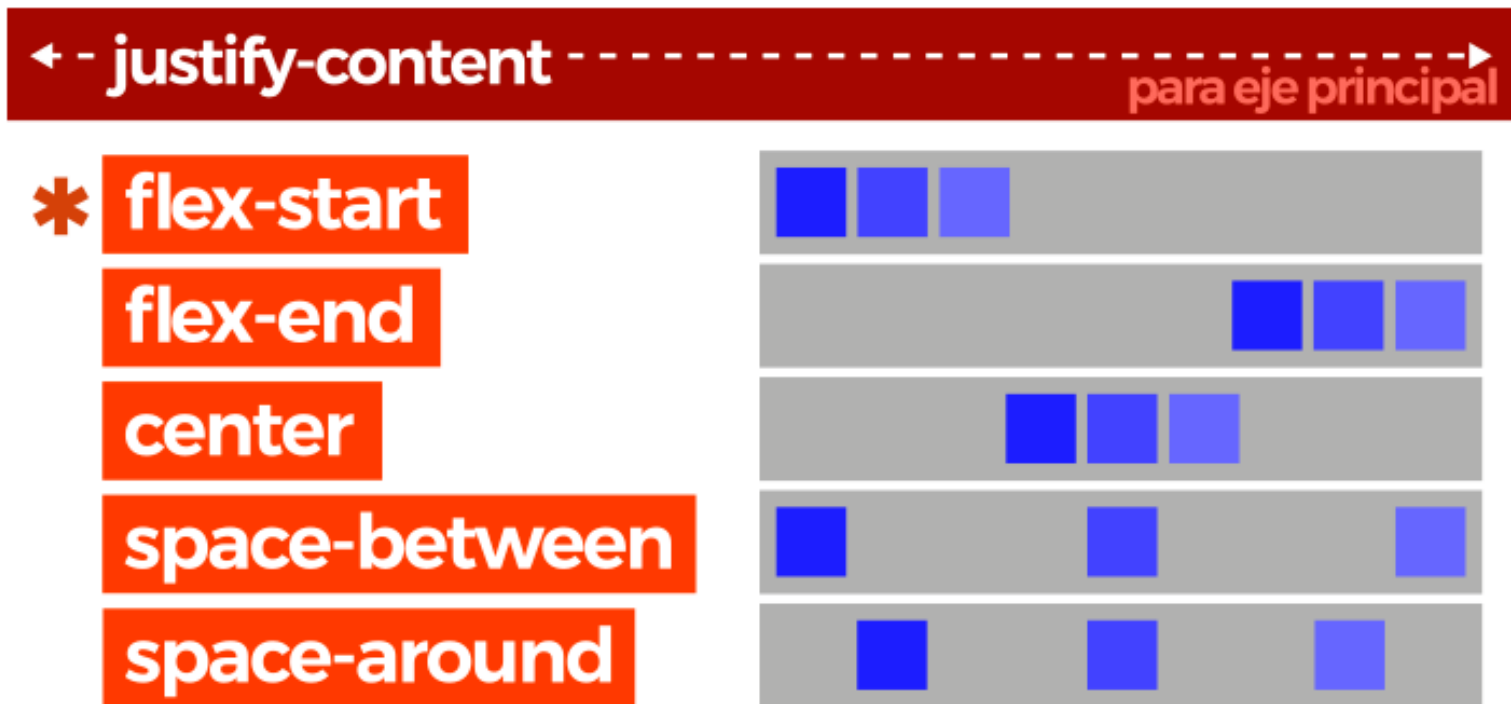
FlexBox

- **Propiedades de alineación de ítems**

Propiedad	Valor	Actúa sobre
<u>justify-content</u>	flex-start flex-end center space-between space-around	Eje principal
align-content	flex-start flex-end center space-between space-around stretch	Eje principal
<u>align-items</u>	flex-start flex-end center stretch baseline	Eje secundario
align-self	auto flex-start flex-end center stretch baseline	Eje secundario

FlexBox

- Alineación de ítems en el eje principal

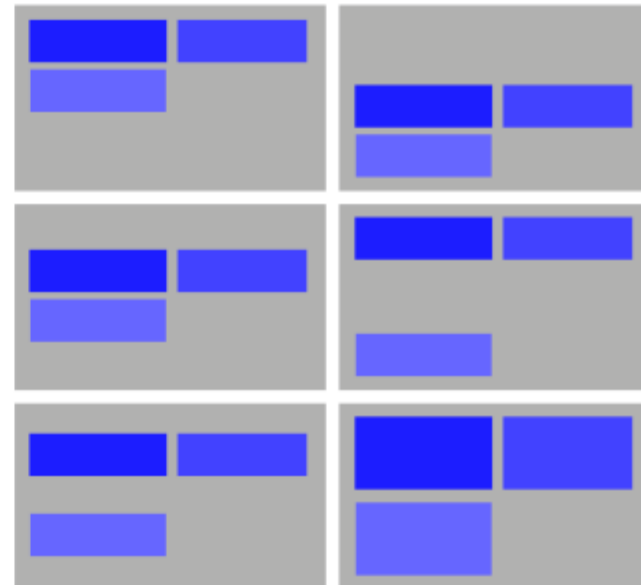


FlexBox

- Alineación de ítems en el eje principal multilínea

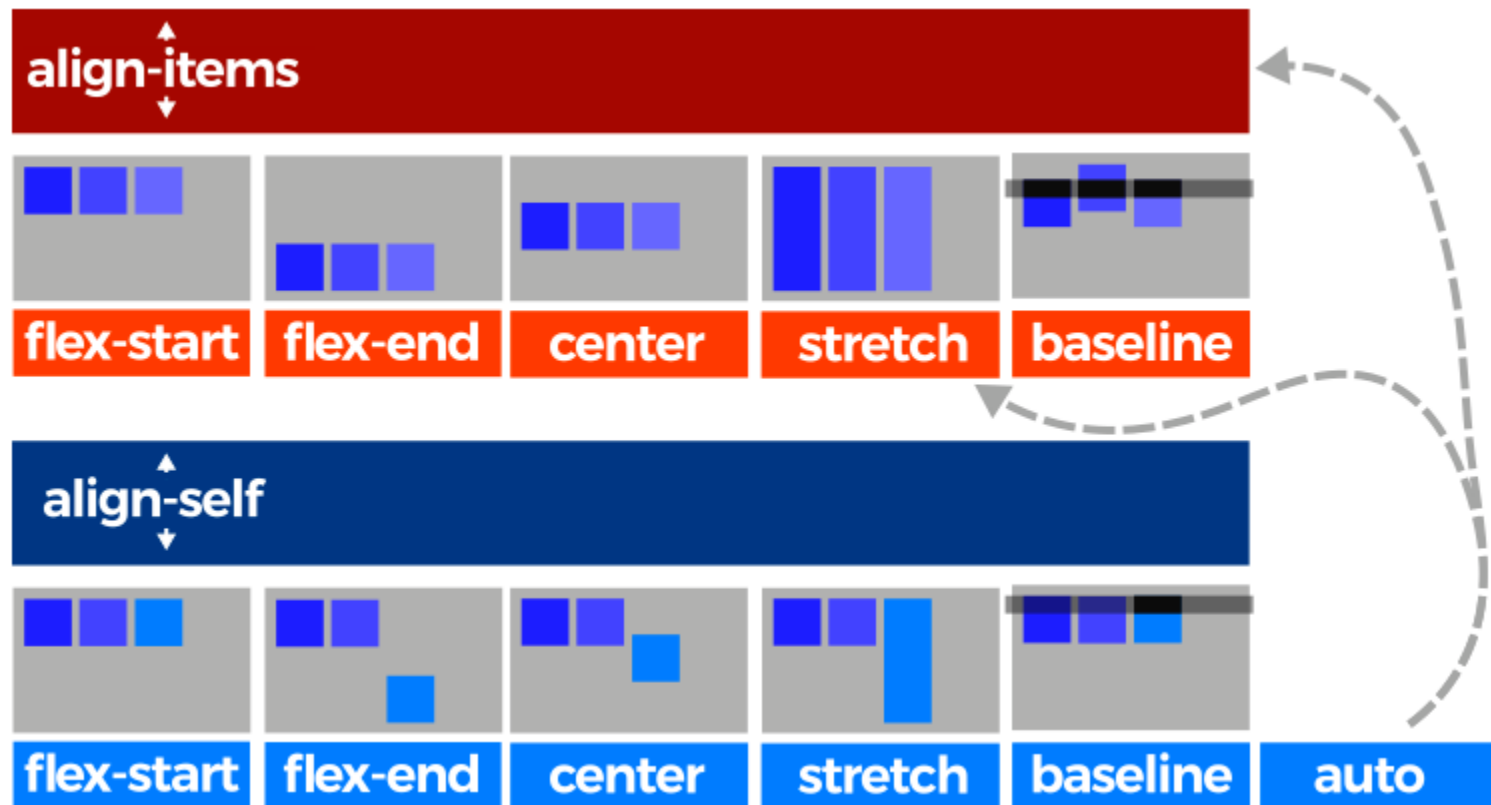


flex-start
flex-end
center
space-between
space-around
*** stretch**



FlexBox

- Alineación en el eje secundario



FlexBox

- Propiedades que se aplican a los elementos hijos (del contenedor)

Propiedad	Valor	Descripción
flex-grow	0 [factor de crecimiento]	Número que indica el crecimiento del ítem respecto al resto.
flex-shrink	1 [factor de decrecimiento]	Número que indica el decrecimiento del ítem respecto al resto.
flex-basis	[tamaño base] content	Tamaño base de los ítems antes de aplicar variación.
order	[número]	Número que indica el orden de aparición de los ítems.

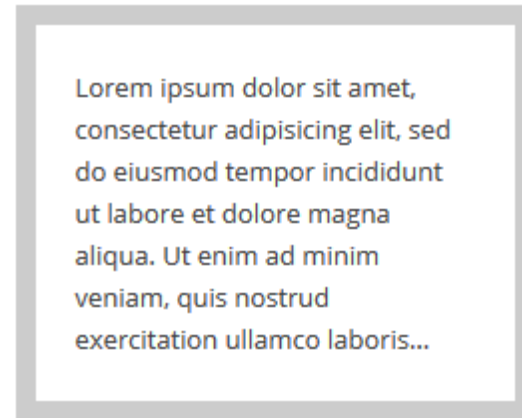
Box-sizing

- La propiedad CSS **box-sizing** se usa para alterar el modelo de caja por defecto usado para calcular alturas y anchuras de elementos.
- **Valores:** **content-box** | **padding-box** | **border-box**
- Por defecto: content-box
 - *content-box*: width y height se miden incluyendo sólo el contenido, pero no el borde, margen o relleno.
 - *padding-box*: width y height incluyen el tamaño del relleno pero no incluyen el borde ni margen.
 - *border-box*: width y height incluyen el relleno y el borde, pero no el margen.

Box-sizing

- Comparado con el modelo de cajas

```
<div style="width:200px;  
padding: 20px;  
border: 10px solid #ccc;  
margin: 0 auto;">  
Lorem ipsum...  
</div>
```



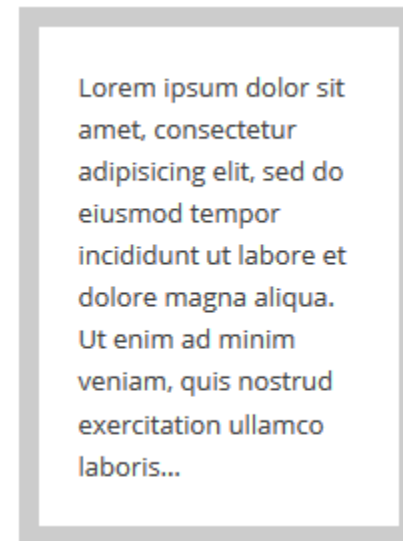
El elemento **<div>** no mide 200px de ancho, sino **260px**. Es decir: 200px de ancho inicial, más 20px de padding izquierdo, más 20px de padding derecho, más 10px de borde izquierdo, más 10px de borde derecho.

Equivalente a `box-sizing: content-box`

Box-sizing

- Comparado con el modelo de cajas
 - `box-sizing: border-box`

```
<div style="width: 200px;  
padding: 20px;  
border: 10px solid #ccc;  
-webkit-box-sizing: border-box;  
-moz-box-sizing: border-box;  
box-sizing: border-box;  
margin: 0 auto;">  
Lorem ipsum...  
</div>
```



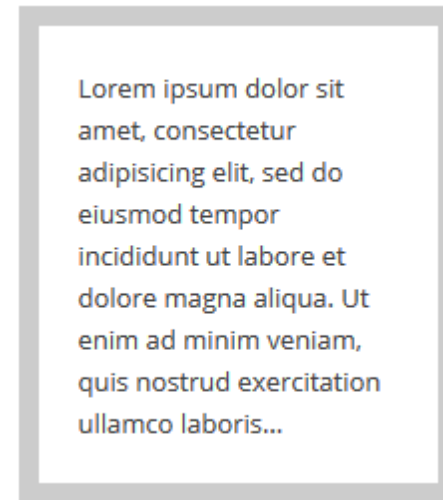
En este caso el elemento `<div>` ocupa **exactamente 200px**, ni uno más, ni uno menos.

Esto es **muy útil para elementos fluidos**, cuando necesitas que el elemento ocupe (por ejemplo) un 33% del ancho, y si ocupa un píxel más toda la estructura se descoloca.

Box-sizing

- Comparado con el modelo de cajas
 - `box-sizing: padding-box`

```
<div style="width:200px;  
padding: 20px;  
border: 10px solid #ccc;  
-webkit-box-sizing: padding-box;  
-moz-box-sizing: padding-box;  
box-sizing: padding-box;  
margin: 0 auto;">  
Lorem ipsum...  
</div>
```



En este caso el elemento es **220px de ancho** (200 entre contenido y padding y 10 de cada lado de borde).
No es tan útil como el caso anterior.

Filtros CSS (efectos de imagen)

- Permite aplicar ciertos efectos de imagen propios de aplicaciones de retoque fotográfico como **sepia** o variación de **contraste** al vuelo, sin hacer cambios permanentes sobre una imagen.
- Propiedad: **filter**
- Ejemplo de uso:

```
img {  
  filter: grayscale(75%);  
}
```

Filtros. Valores de filter

Función	Significado	Valor	Descripción
grayscale	Escala de blanco y negro	porcentaje	0% = original, 100% = B&N
blur	Grado de difuminado	píxeles	Radio de desenfoque (en píxeles)
sepia	Grado de color sepia	porcentaje	0% = original, 100% = sepia
saturate	Grado de saturación	porcentaje	0% = B&N, 100% = original, permite >100%
opacity	Grado de transparencia	porcentaje	0% = invisible, 100% = visible
brightness	Brillo	porcentaje	0 = negro, 100% = original, permite >100%
contrast	Contraste	porcentaje	0 = gris, 100% = original, permite >100%
hue-rotate	Rotación de color (matiz)	grados	0 - 360deg = original
invert	Invertir	porcentaje	0% original, 100% = invertido
drop-shadow	Sombra idéntica	x y blur color	Ejemplo: filter: drop-shadow(9px 9px 9px rgba(0,0,0,0.3))

En funcionamiento:

- <https://devcode.la/tutoriales/filtros-de-instagram-con-css/>
- <https://developer.mozilla.org/es/docs/Web/CSS/filter>

Filtros. Modos de fusión

- Propiedad: **mix-blend-mode**
 - Permite utilizar los denominados **modos de fusión**, muy comunes en programas profesionales de retoque fotográfico, sobre elementos de contenido como imágenes o similares.
- Propiedad: **background-blend-mode**
 - Para aplicar el modo de fusión a imágenes de fondo mediante las propiedades background-image o background.
- Ejemplo:

```
body {  
  background:green;  
}  
  
img {  
  mix-blend-mode:multiply;  
}
```


Filtros. Modos de fusión

Modo de fusión	Significado
normal	Sin modo de fusión.
multiply	Multiplicar.
screen	Trama.
overlay	Superponer.
darken	Oscurecer.
lighten	Aclarar.
color-dodge	Sobreexponer color.
color-burn	Subexponer color.
hard-light	Luz fuerte.
soft-light	Luz suave.
difference	Diferencia.
exclusion	Exclusión.
hue	Tono.
saturation	Saturación.
color	Color.
luminosity	Luminosidad.

Valores a utilizar
en las propiedades
anteriores

En funcionamiento: <https://css-tricks.com/almanac/properties/m/mix-blend-mode/>

Cursor

- Permite modificar el puntero del ratón.

Propiedad	cursor
Valores	((url ,)* (auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help progress)) inherit
Se aplica a	Todos los elementos
Valor inicial	auto
Descripción	Permite personalizar el puntero del ratón

Cursor

- Se puede seleccionar un puntero de los disponibles en el sistema operativo o bien indicar una URL de una imagen que se quiera mostrar.
- Ejemplo:

```
cursor: url(images/cursor.cur);  
cursor: url(images/cursor.png), default;
```

Cursor

CSS2:

```
cursor: auto;  
cursor: inherit;  
cursor: crosshair;  
cursor: default;  
cursor: help;  
cursor: move;  
cursor: pointer;  
cursor: progress;  
cursor: text;  
cursor: wait;  
cursor: e-resize;  
cursor: ne-resize;  
cursor: nw-resize;  
cursor: n-resize;  
cursor: se-resize;  
cursor: sw-resize;  
cursor: s-resize;  
cursor: w-resize;
```

CSS3 (funcionan en los browsers modernos)

```
cursor: none;  
cursor: context-menu;  
cursor: cell;  
cursor: vertical-text;  
cursor: alias;  
cursor: copy;  
cursor: no-drop;  
cursor: not-allowed;  
cursor: ew-resize;  
cursor: ns-resize;  
cursor: nesw-resize;  
cursor: nwse-resize;  
cursor: col-resize;  
cursor: row-resize;  
cursor: all-scroll;
```

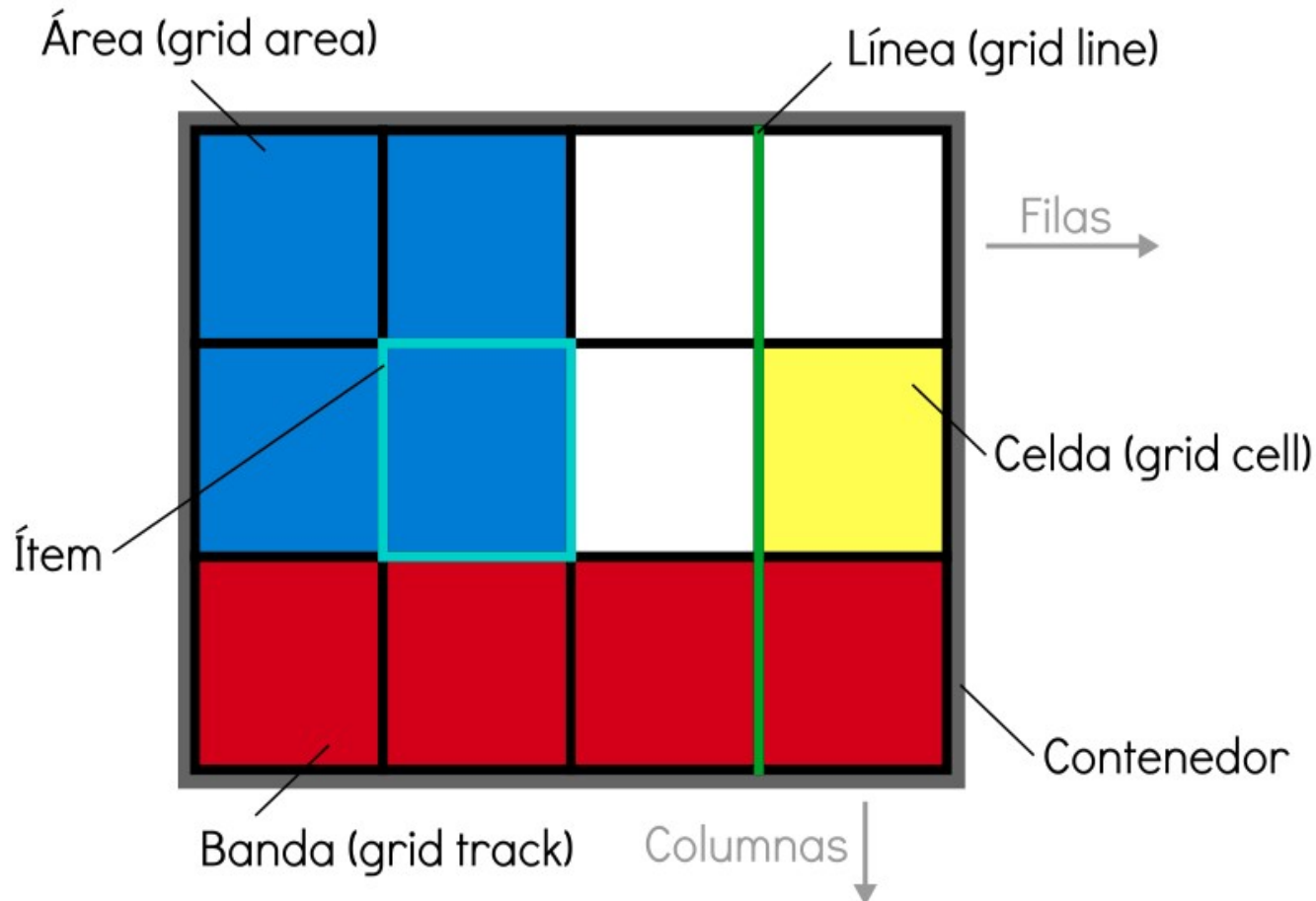
En funcionamiento: <http://www.csslab.cl/ejemplos/cursor/>

2.15 GRID (Cuadrícula)

El sistema flexbox es una gran mejora, sin embargo, está orientado a estructuras de una sola dimensión, por lo que aún necesitamos algo más potente para estructuras web. Con el paso del tiempo, muchos frameworks y librerías utilizan un sistema grid donde definen una cuadrícula determinada, y modificando los nombres de las clases de los elementos HTML, podemos darle tamaño, posición o colocación.

Grid CSS nace de esa necesidad, y recoge las ventajas de ese sistema, añadiéndole numerosas mejoras y características que permiten crear rápidamente cuadrículas sencillas y potentes de forma prácticamente instantánea.

2.15 GRID (Cuadrícula)



2.15 GRID (Cuadrícula)

Contenedor: Existe un elemento padre que es el contenedor que definirá la cuadrícula o rejilla.

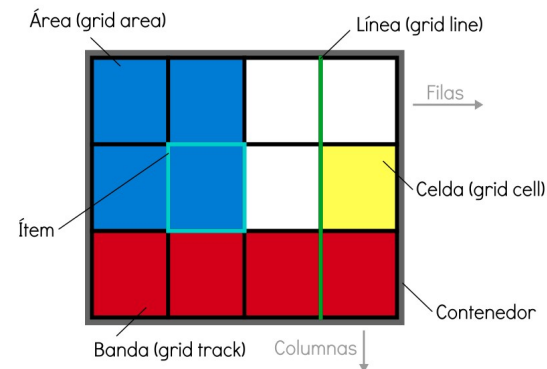
Ítem: Cada uno de los hijos que contiene la cuadrícula (elemento contenedor).

Celda (grid cell): Cada uno de los cuadritos (unidad mínima) de la cuadrícula.

Área (grid area): Región o conjunto de celdas de la cuadrícula.

Banda (grid track): Banda horizontal o vertical de celdas de la cuadrícula.

Línea (grid line): Separador horizontal o vertical de las celdas de la cuadrícula



2.15 GRID (Cuadrícula)

Trabajaremos bajo el siguiente escenario:

```
<div class="grid"> <!-- contenedor -->
  <div class="a">Item 1</div> <!-- cada uno de los ítems del grid -->
  <div class="b">Item 2</div>
  <div class="c">Item 3</div>
  <div class="d">Item 4</div>
</div>
```

Para activar la cuadrícula grid hay que utilizar sobre el elemento contenedor la propiedad display y especificar el valor grid o inline-grid.

Tipo de elemento	Descripción
inline-grid	Establece una cuadrícula con ítems en línea, de forma equivalente a inline-block.
grid	Establece una cuadrícula con ítems en bloque, de forma equivalente a block.

2.15 GRID- Grid con filas y columnas

Fracción restante en grid

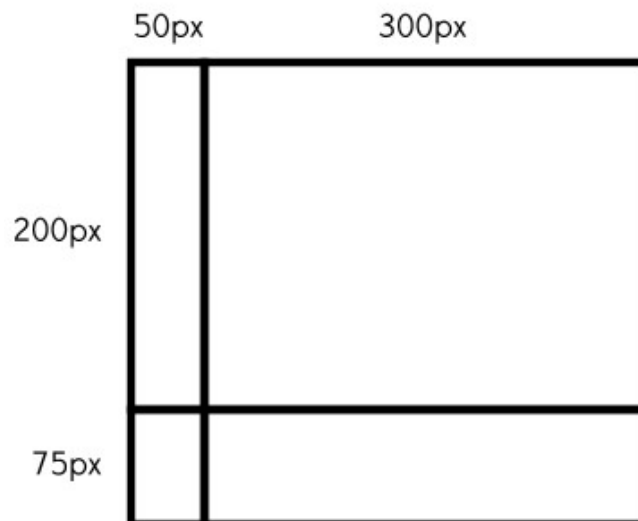
```
<div class="grid"> <!-- contenedor -->
  <div class="a">Item 1</div> <!-- cada uno de los items del grid -->
  <div class="b">Item 2</div>
  <div class="c">Item 3</div>
  <div class="d">Item 4</div>
</div>
```

Las propiedades CSS `grid-template-columns` y `grid-template-rows`, que sirven para indicar las dimensiones de cada celda de la cuadrícula, diferenciando entre columnas y filas.

Propiedad	Valor	Descripción
<code>grid-template-columns</code>	<code>[<u>c1</u>] [<u>c2</u>] ...</code>	Establece el TAMAÑO de cada columna (<u>col 1</u> , <u>col 2</u> ,...).
<code>grid-template-rows</code>	<code>[<u>f1</u>] [<u>f2</u>] ...</code>	Establece el TAMAÑO de cada fila (<u>fila 1</u> , <u>fila 2</u> ,...).

```
.grid {
  display: grid;
  grid-template-columns: 50px 300px;
  grid-template-rows: 200px 75px;
}
```

```
.grid {
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-template-rows: 2fr 1fr;
}
```



Nota: Se pueden combinar varias unidades diferentes, pudiendo utilizar píxeles (*px*) y fracciones restantes (*fr*), porcentajes (%) y fracciones restantes (*fr*) o combinaciones similares.

2.15 GRID- Grid con filas y columnas Fracción restante en grid

- Ejemplo 1:
 - https://www.w3schools.com/css/tryit.asp?filename=trycss_gridExe

```
<div class="grid"> <!-- contenedor -->
  <div class="a">Item 1</div> <!-- cada uno de los ítems del grid -->
  <div class="b">Item 2</div>
  <div class="c">Item 3</div>
  <div class="d">Item 4</div>
</div>
```

```
.grid {
  display: grid;
  grid-template-columns: 50px 300px;
  grid-template-rows: 200px 75px;
}
```

```
.grid {
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-template-rows: 2fr 1fr;
}
```

2.15 GRID- Filas y columnas repetitivas

Se puede utilizar la expresión **repeat()** para indicar repetición de valores, indicando el número de veces que se repiten y el tamaño en cuestión.

La expresión a utilizar sería la siguiente: **repeat([núm de veces], [valor o valores])**.

```
.grid {  
  display: grid;  
  grid-template-columns: 100px repeat(2, 50px) 200px;  
  grid-template-rows: repeat(2, 50px 100px);  
}
```

```
.grid {  
  display: grid;  
  grid-template-columns: 100px 50px 50px 200px;  
  grid-template-rows: 50px 100px 50px 100px;;  
}
```

2.15 GRID- Grid por áreas

```
<div class="grid"> <!-- contenedor -->
  <div class="a">Item 1</div> <!-- cada
  <div class="b">Item 2</div>
  <div class="c">Item 3</div>
  <div class="d">Item 4</div>
</div>
```

Mediante los grids CSS es posible indicar el nombre y posición concreta de cada área de una cuadrícula.

Propiedad	Descripción
grid-template-areas	Indica la disposición de las áreas en el grid. Cada texto entre comillas simboliza una fila.
grid-area	Indica el nombre del área. Se usa sobre ítems hijos del grid.

```
.grid {
  display: grid;
  grid-template-areas: "head head"
                      "menu main"
                      "foot foot";
}

.a { grid-area: head; background: blue }
.b { grid-area: menu; background: red }
.c { grid-area: main; background: green }
.d { grid-area: foot; background: orange }
```



En la propiedad **grid-template-areas**, palabra clave especial:

- **none**: se colocará ninguna celda en esta posición.
- **Uno o más puntos (.)**: se colocará una celda vacía en esta posición.

2.15 GRID- Grid por áreas

Ejemplos:

https://www.w3schools.com/cssref/tryit.asp?filename=trycss_grid-template-areas3

https://www.w3schools.com/cssref/tryit.asp?filename=trycss_grid-template-areas2

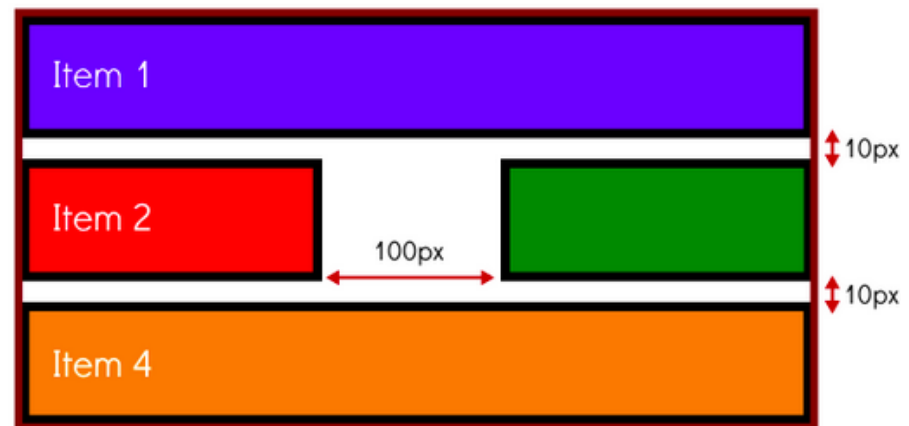
https://www.quackit.com/html/html_editors/scratchpad/?example=/html/templates/grid/_inc/inc_css_grid_layout_template_8.cfm

2.15 GRID- Grid con huecos / atajo

Para especificar los huecos (espacio entre celdas) podemos utilizar las propiedades **grid-column-gap** y/o **grid-row-gap**.

Propiedad	Descripción
grid-column-gap	Establece el TAMAÑO de los huecos entre columnas (<i>líneas verticales</i>).
grid-row-gap	Establece el TAMAÑO de los huecos entre filas (<i>líneas horizontales</i>).

```
.grid {  
  grid-column-gap: 100px;  
  grid-row-gap: 10px;  
}
```



```
.grid {  
  /* grid-gap: <row-gap> <column-gap> */  
  grid-gap: 20px 80px;  
  
  /* grid-gap: <row-column-gap> */  
  grid-gap: 40px;  
  /* Equivalente a grid-gap: 40px 40px; */  
}
```

2.15 GRID- Posición en el grid

Propiedad	Valores	Descripción
justify-items	start end center stretch	Distribuye los elementos en el eje horizontal.
align-items	start end center stretch	Distribuye los elementos en el eje vertical.

Estas propiedades se aplican sobre el elemento contenedor padre, pero afectan a los ítems hijos, por lo que actúan sobre la distribución de cada uno de los hijos. En el caso de que queramos que uno de los ítems hijos tengan una distribución diferente al resto, aplicamos la propiedad **justify-self** o **align-self** sobre el ítem hijo en cuestión, sobrescribiendo su distribución.

También podemos utilizar las propiedades **justify-content** o **align-content** para modificar la distribución de todo el contenido en su conjunto, y no sólo de los ítems por separado:

Propiedad	Valores
justify-content	start end center stretch space-around space-between space-evenly
align-content	start end center stretch space-around space-between space-evenly

2.15 GRID- Posición en el grid

```
15 justify-items:stretch;
16 height:200px;
17 align-items:stretch;
18
19 justify-content:stretch;
20 align-content:stretch;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



2.15 GRID- Ajuste automático de celdas

las propiedades **grid-auto-columns** y **grid-auto-rows** para darle un tamaño automático a las celdas de la cuadrícula.

podemos utilizar **grid-auto-flow** para indicar el flujo de elementos en la cuadrícula, y especificar por donde se irán añadiendo

Propiedad	Valores	Descripción
grid-auto-columns	TAMAÑO	Indica el tamaño automático de ancho que tendrán las columnas.
grid-auto-rows	TAMAÑO	Indica el tamaño automático de alto que tendrán las filas.
grid-auto-flow	row column dense	Utiliza un algoritmo de autocolocación (intenta rellenar huecos).



grid-auto-flow: row



grid-auto-flow: column



2.15 GRID- Atajo

Propiedad **grid** que sirve de atajo

```
.grid {  
  /* grid: <grid-template> <grid-auto-flow> <grid-auto-rows> / <grid-auto-columns> */  
  
  grid: 100px 20px;  
  grid: 200px repeat(2, 100px) 300px;  
  grid: row;  
  grid: column dense;  
  grid: row 200px;  
  grid: row 400px / 150px;  
}
```

Propiedad	Descripción
justify-self	Altera la justificación del ítem hijo en el eje horizontal.
align-self	Altera la alineación del ítem hijo en el eje vertical.
grid-area	Indica un nombre al área especificada, para su utilización con grid-template-areas .

2.15 GRID- Propiedades para ítems hijos

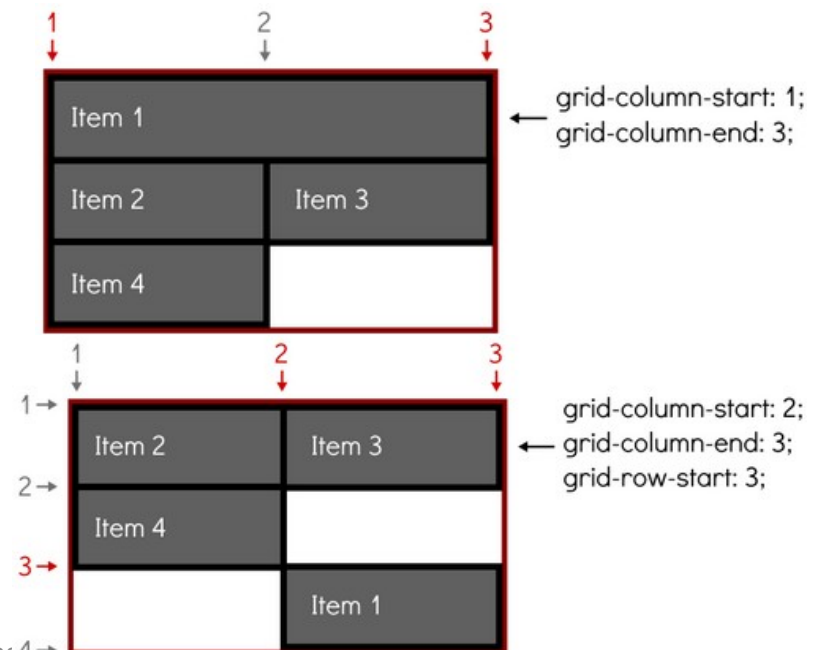
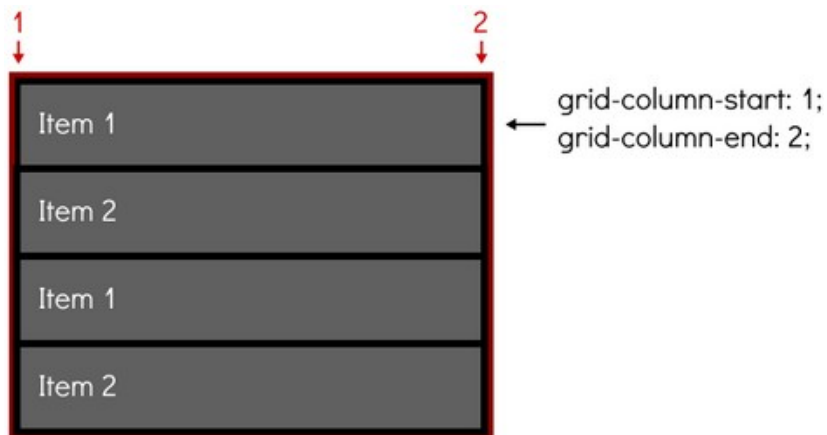
Algunas de las propiedades vistas hasta ahora son las siguientes:

Existen algunas propiedades más

Propiedad	Descripción
grid-column-start	Indica en que columna empezará el ítem de la cuadrícula.
grid-column-end	Indica en que columna terminará el ítem de la cuadrícula.
grid-row-start	Indica en que fila empezará el ítem de la cuadrícula.
grid-row-end	Indica en que fila terminará el ítem de la cuadrícula.

```
.grid {
  display: grid;
}

.a {
  grid-column-start: 1;
  grid-row-end: 2;
}
```



2.15 GRID- Atajo: grid-column y grid-row

```
.grid {  
  display: grid;  
}  
  
.a {  
  /* grid-column: <grid-column-start> <grid-column-end> */  
  /* grid-row: <grid-row-start> <grid-row-end> */  
  grid-column: auto;  
  grid-column: 4 / 6;  
  grid-column: span 3;  
  grid-column: span 3 / 6;  
}
```

2.15 GRID

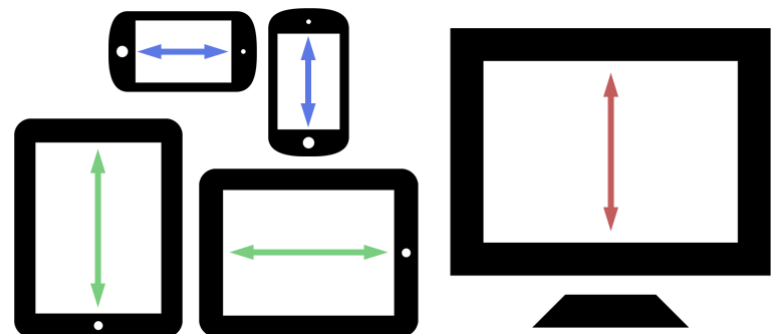
- Grid para formularios:
 - https://www.quackit.com/html/html_editors/scratchpad/?example=/css/grid/examples/forms_auto-placement_example_1
- Grid para tarjetas
 - https://www.quackit.com/html/html_editors/scratchpad/preview.cfm
- Grid para galería de fotos
 - https://www.quackit.com/html/html_editors/scratchpad/?example=/css/grid/examples/photo_gallery_flexible_width_span_3_columns_2_rows

2.15 GRID- Referencias

- <https://lenguajecss.com/p/css/propiedades/grid-css>
- https://www.quackit.com/css/grid/tutorial/css_grid_introduction.cfm
- <https://css-tricks.com/snippets/css/complete-guide-grid/>

2.16 Diseño web adaptativo (Responsive Web Design)

- Es una de las características que más se ha popularizado en los últimos años.
- Se le denomina así a los diseños web que tienen la capacidad de adaptarse al tamaño y formato de la pantalla en la que se visualiza el contenido, respecto a los diseños tradicionales en los que las páginas web estaban diseñadas para un tamaño o formato específico.



Diseño web adaptativo (Responsive Web Design)

- Antes de comenzar a crear un diseño web adaptativo, es importante tener claro una serie de conceptos:
 - Conoce las diferentes resoluciones más utilizadas:
 - <http://screensiz.es/phone>
 - <https://www.mydevice.io/>
 - Conoce el público de tu sitio web y tu objetivo:
 - ¿Acceden más usuarios desde móvil o desde escritorio?
 - ¿Predominan las tabletas o los móviles?
 - ¿Tu objetivo es tener más usuarios de móvil o de escritorio?
 - Elige una estrategia acorde a lo anterior

Diseño web adaptativo (Responsive Web Design)

- Ejemplos de páginas que utilizan Responsive Web Design: <http://mediaqueri.es/>
- Estadísticas usuarios: <https://www.google.com/analytics/>
- Estadísticas navegadores: http://www.w3schools.com/browsers/browsers_display.asp
- Estadísticas globales: <http://gs.statcounter.com/>

Diseño web adaptativo (Responsive Web Design)

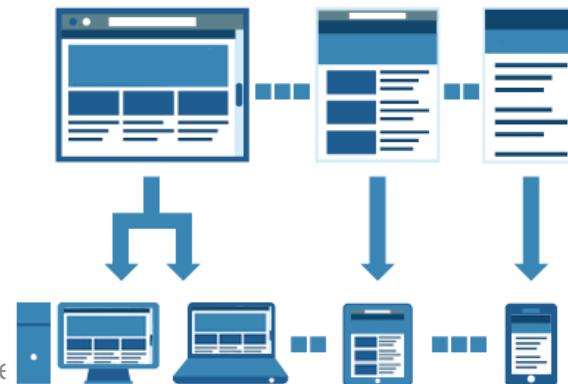
- **Diseño con porcentajes**

- Son medidas relativas al contenedor padre.
Si especificamos un % a un elemento, el navegador tomará dicho % del contenedor.
- En las propiedades: **width**
- No garantiza diseño adaptativo: hay que tener en cuenta los tamaños de relleno (padding) y bordes (border) para que no se descuadre el diseño.

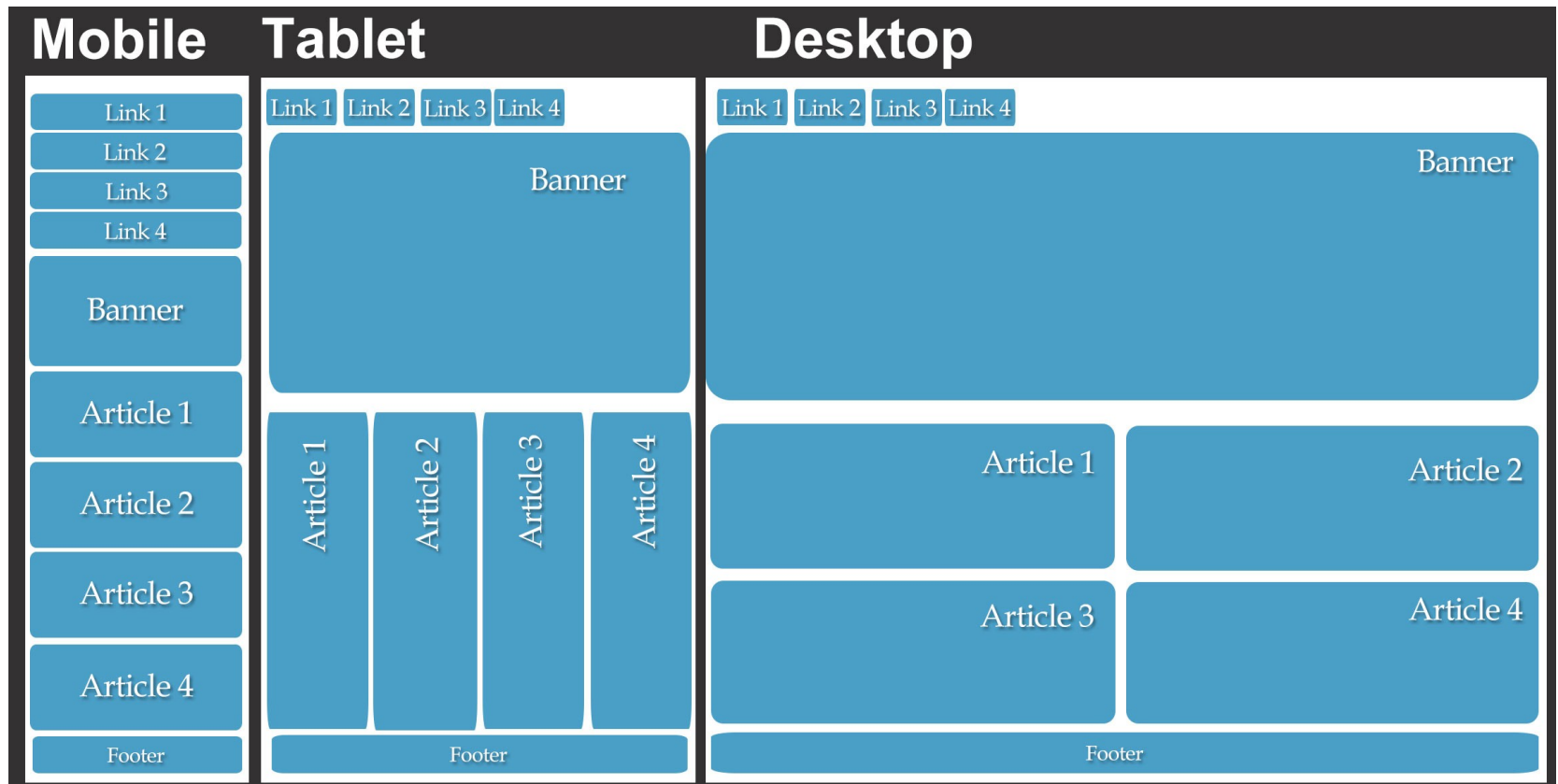
Diseño web adaptativo (Responsive Web Design)

- **Diseño con porcentajes**

- Son medidas relativas al contenedor padre.
Si especificamos un % a un elemento, el navegador tomará dicho % del contenedor.
- En las propiedades: **width**
- No garantiza diseño adaptativo: hay que tener en cuenta los tamaños de relleno (padding) y bordes (border) para que no se descuadre el diseño.



Diseño web adaptativo (Responsive Web Design)



Diseño web adaptativo (Responsive Web Design)

- **Tamaños máximos y mínimos**

- Mayor control con las propiedades:
max-width y **min-width** .

- Ejemplo:

```
div {  
    max-width: 800px;  
    min-width: 300px;  
}
```

- El elemento nunca se hará más pequeño de 300 o más grande de 800.
- En este caso el ancho (width) es automático. Si se establece esta propiedad, las de mínimo y máximo pierden efecto.
- Ejemplo con imágenes, vídeo y multimedia:

```
img, video, object, embed {  
    max-width: 100%;  
    height: auto;  
}
```

Diseño web adaptativo (Responsive Web Design)

- **Media queries**

```
@media screen and (*condición*) {  
    /* reglas CSS */  
    /* reglas CSS */  
}
```

- Con este método, especificamos que queremos aplicar los estilos CSS para tipos de medios concretos (*screen: sólo en pantallas, en este caso*) que cumplan las condiciones especificadas entre paréntesis.
- Aconsejable:
 - crear reglas CSS generales aplicadas a todo el documento: colores, tipo de fuente, etc. (*características comunes en móviles y escritorio*), y luego,
 - las particularidades que se aplicarían sólo en el dispositivo en cuestión.

Diseño web adaptativo (Responsive Web Design)

- **Media queries. Ejemplo**

```
@media screen and (max-width:640px) {  
    /* reglas CSS */  
}  
@media screen and (max-width:1024px) and (min-width:640px) {  
    /* reglas CSS */  
}  
@media screen and (min-width:1024px) {  
    /* reglas CSS */  
}
```

- Resoluciones menores de 640 píxeles de ancho.
 - Resoluciones menores de 1024 píxeles de ancho (*pero mayores que 640 píxeles*).
 - Resoluciones mayores de 1024 píxeles.
- Ver una página web en las diferentes resoluciones o formatos indicados:

<http://johnpolacek.github.io/Responsivator/>

Diseño web adaptativo (Responsive Web Design)

- **El «viewport»**

- Hace referencia a la **región visible del navegador**, o sea, la parte de la página que está visualizándose actualmente en el navegador.
- Podemos redimensionar la ventana del navegador para reducir el tamaño del viewport y simular que se trata de una pantalla y dispositivo más pequeño.
- Podemos editar el documento HTML para especificar el siguiente campo meta, antes de la parte del </head>:

```
<meta name="viewport" content="initial-scale=1, width=device-width">
```


Diseño web adaptativo (Responsive Web Design)

- device-width



Diseño web adaptativo (Responsive Web Design)

Valores equivalentes
yes=1
no=0.1
device-width=10
device-height=10

- El «**viewport**»

- Con esa etiqueta **meta**, estamos estableciendo unos parámetros de comportamiento para el **viewport**:

Propiedades	Valor	Significado
width	device-width	Indica un ancho para el viewport.
height	device-height	Indica un alto para el viewport.
initial-scale	1	Escala inicial con la que se visualiza la página web. (*)
minimum-scale	0.1	Escala mínima a la que se puede reducir al hacer zoom. (*)
maximum-scale	10	Escala máxima a la que se puede aumentar al hacer zoom. (*)
user-scalable	no/fixed yes/zoom	Posibilidad de hacer zoom en la página web (táctil)

Diseño web adaptativo (Responsive Web Design)

- Ejemplos:
 - <meta> “viewport”
 - https://www.w3schools.com/html/tryit.asp?filename=tryhtml_responsive_viewport
 - Web adaptable
 - https://www.w3schools.com/html/tryit.asp?filename=tryhtml_responsive_picture
 - Web responsive
 - https://www.w3schools.com/html/tryit.asp?filename=tryhtml_responsive_text
 - https://www.w3schools.com/html/tryit.asp?filename=tryhtml_responsive_media_query
 - https://www.w3schools.com/html/tryit.asp?filename=tryhtml_responsive_media_query3

Direcciones web de consulta

- <http://librosweb.es/>
- <https://developer.mozilla.org/es/docs/Web/CSS/CSS3>
- <http://www.arkaitzgarro.com/>
- <http://www.w3schools.com/css/default.asp>
- Listado de funciones CSS3 soportadas por cada navegador:
<http://caniuse.com/>
- <http://www.mclibre.org/>
- <https://lenguajecss.com/p/css/>
- <https://devcode.la/tutoriales/modelo-caja-css/>
- <http://css3clickchart.com>