

# **LENGUAJES DE MARCAS Y SISTEMAS DE INFORMACIÓN**

**UT5. CSS**

**M<sup>a</sup> JESÚS BRAVO 2024-2025**

A decorative wavy line in a light purple color runs vertically along the left side of the slide, separating the light purple background from the dark blue background.

# **CSS. COLORES, MEDIDAS Y MÁS**

# MEDIDAS



# UNIDADES DE MEDIDA

- Altura, anchura, márgenes, tamaño de la letra...
- Se definen con un número entero o decimal seguido de la unidad SIN ESPACIO:

```
div {  
    height: 200px;  
    width: 300px;  
}  
  
/* Márgenes en em y porcentaje */  
p {  
    margin: 1em 2em;  
}  
  
/* Tamaño de Letra en puntos */  
h1 {  
    font-size: 24pt;  
}
```

# UNIDADES DE MEDIDA

- **Las medidas relativas**

- Definen su valor **en relación con otra medida**, por lo que, para obtener su valor real, se debe realizar alguna operación con el valor indicado.
- Se adaptan mejor a diferentes tamaños de pantalla y dispositivos.
- Los cambios en el tamaño de fuente o contenedor padre afectan automáticamente a los elementos hijos.

- **Las medidas absolutas**

- Establecen de forma completa el valor de una medida, por lo que su valor real es directamente el valor indicado.
- Mantienen un tamaño fijo independientemente del contexto.
- Útil en impresión de documentos.

- A tener en cuenta:

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

# UNIDADES DE MEDIDA

## Unidades Relativas:

em	Relativo al tamaño de letra del elemento (2em = 2x el tamaño de la letra actual) Em (tamaño de la M)
ex	Relativo al alto de la letra X de la fuente actual. Raramente usado.
ch	Relativo al alto de la letra “O” de la fuente actual. Raramente usado.
rem	Relativo al tamaño de la letra del elemento raíz (html). Si el tamaño de la fuente del elemento raíz es 16 píxeles, 1rem será igual a 16 píxeles.
vw	Relativo al porcentaje del ancho total del viewport (ventana gráfica visible en el navegador). Por ejemplo, si defines un elemento con width: 50vw, ese elemento ocupará el 50% del ancho total del viewport. Si el viewport es de 50cm, 1vw es 0,5 cm.
vh	Similar a vw, pero en lugar de ser relativo al ancho, es relativo a la altura del viewport. Si defines un elemento con height: 50vh, ocupará el 50% de la altura total del viewport.
vmin	<b>vmin</b> representa el valor mínimo entre el ancho y la altura del viewport,
vmax	<b>vmax</b> representa el valor máximo entre el ancho y la altura del viewport. Por ejemplo, si defines width: 50vmin, el ancho del elemento será el 50% del valor mínimo entre el ancho y la altura del viewport.
%	Relativo al elemento padre

# UNIDADES DE MEDIDA

## Unidades Relativas:

**Em:** Para la unidad em, la referencia es el tamaño de letra del elemento padre

```
<p>Texto a 16px</p>
<div class="article">
  <div>Texto a 2rem</div>
  <p class="pequeno">Texto a 0.5rem.</p>
</div>
```

```
html {
  font-size: 100%; /* 100% = 16px */
}
.article {
  font-size: 2em; /* Ahora 2em = 32px */
}
.pequeno {
  font-size: 0.5em; /* 0,5 x 32px = 16px */
}
```

Texto a 16px

Texto a 2em

Texto a 0.5em.

# UNIDADES DE MEDIDA

## Unidades Relativas:

**Rem:** tamaño relativo a la fuente del elemento raíz (<html>).

```
<p>Texto a 16px</p>
<div class="article">
  <div>Texto a 2rem</div>
  <p class="pequeno">Texto a 0.5rem.</p>
</div>
```

```
html {
  font-size: 100%; /* 100% = 16px */
}
.article {
  font-size: 2rem; /* Ahora 1em = 32px */
}
.pequeno {
  font-size: 0.5rem; /* 0,5 × 16px = 8px */
}
```

Texto a 16px

Texto a 2rem

Texto a 0.5rem.

<https://cybmeta.com/em-y-rem>



# UNIDADES DE MEDIDA

- Porcentaje:
  - Para el tamaño de letra de los elementos equivale a em. No se usa.
  - Para el tamaño de elementos con width y height

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

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

# UNIDADES DE MEDIDA

<b>vw</b>	Relativo al porcentaje del ancho total del viewport (ventana gráfica visible en el navegador). Por ejemplo, si defines un elemento con <code>width: 50vw</code> , ese elemento ocupará el 50% del ancho total del viewport.
<b>vh</b>	Similar a <code>vw</code> , pero en lugar de ser relativo al ancho, es relativo a la altura del viewport. Si defines un elemento con <code>height: 50vh</code> , ocupará el 50% de la altura total del viewport.
<b>vmin</b>	<b>vmin</b> representa el valor mínimo entre el ancho y la altura del viewport.
<b>vmax</b>	<b>vmax</b> representa el valor máximo entre el ancho y la altura del viewport. Por ejemplo, si defines <code>width: 50vmin</code> , el ancho del elemento será el 50% del valor mínimo entre el ancho y la altura del viewport.

```
.vw h1{  
    font-size: 10vw;  
}  
.vw p{  
    font-size: 16px;  
}
```

```
<div class="vw">  
  <h1>Cabecera con 10 vw</h1>  
  <p>Parrafo con tamaño fijo 16px</p>  
</div>
```

# UNIDADES DE MEDIDA

- Absolutas

px *	Píxel, unidad de medida que representa un punto en una pantalla. (1px = 1/96th of 1in)
pt	Punto, unidad de medida comúnmente utilizada para tamaños de texto en impresión. Un punto es aproximadamente igual a 1/72 de pulgada.
in	Pulgada, unidad de medida que representa una pulgada física en la pantalla. (1in = 96px = 2.54cm)
cm	Centímetro, unidad de medida que representa un centímetro físico en la pantalla.
mm	Milímetro, unidad de medida que representa un milímetro físico en la pantalla.

•**Absoluta:** En teoría, un píxel es una unidad de medida absoluta, que representa un punto en una pantalla.

**Relativa:** Sin embargo, la percepción de tamaño puede variar en dispositivos con diferentes densidades de píxeles.

# UNIDADES DE MEDIDA

- Fuentes predeterminadas en el navegador:
- Chrome: `chrome://settings/fonts`
- Firefox: `about:preferences#general`



# FONT

...tanto, ei...  
...hor, em que nunca...  
...senhor, que fez subir os filh...  
...ael da terra do Egito;  
...Mas: Vive o Senhor, que fez...  
...que trouxe a geração da casa...  
...el da terra do norte, e de to...  
...ras para onde os tinha arroj...  
...bitarão na sua terra.

## ***Contra os falsos profetas***

Quanto aos profetas. O me...  
...está quebrantado dentro de...  
...os os meus ossos estremece...  
...no um homem embriagado, e...  
...homem vencido do vinho, p...  
...hor, e por causa das p...

# TERMINOLOGÍA DE TIPOGRAFÍA

## AaBbCc

- Fuentes Serif

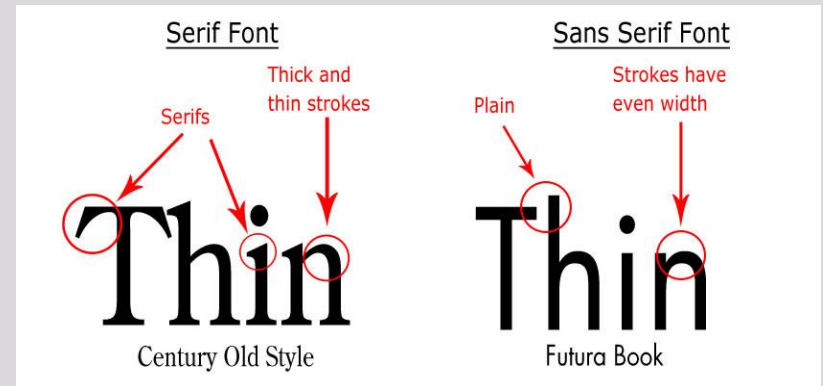
- Tiene detalles adicionales en el trazo. Inspirada en la caligrafía y utilizada tradicionalmente para extensos pasajes de texto en impresión.
- Ejemplos: Georgia, Times New Roman

- Fuentes Sans-Serif

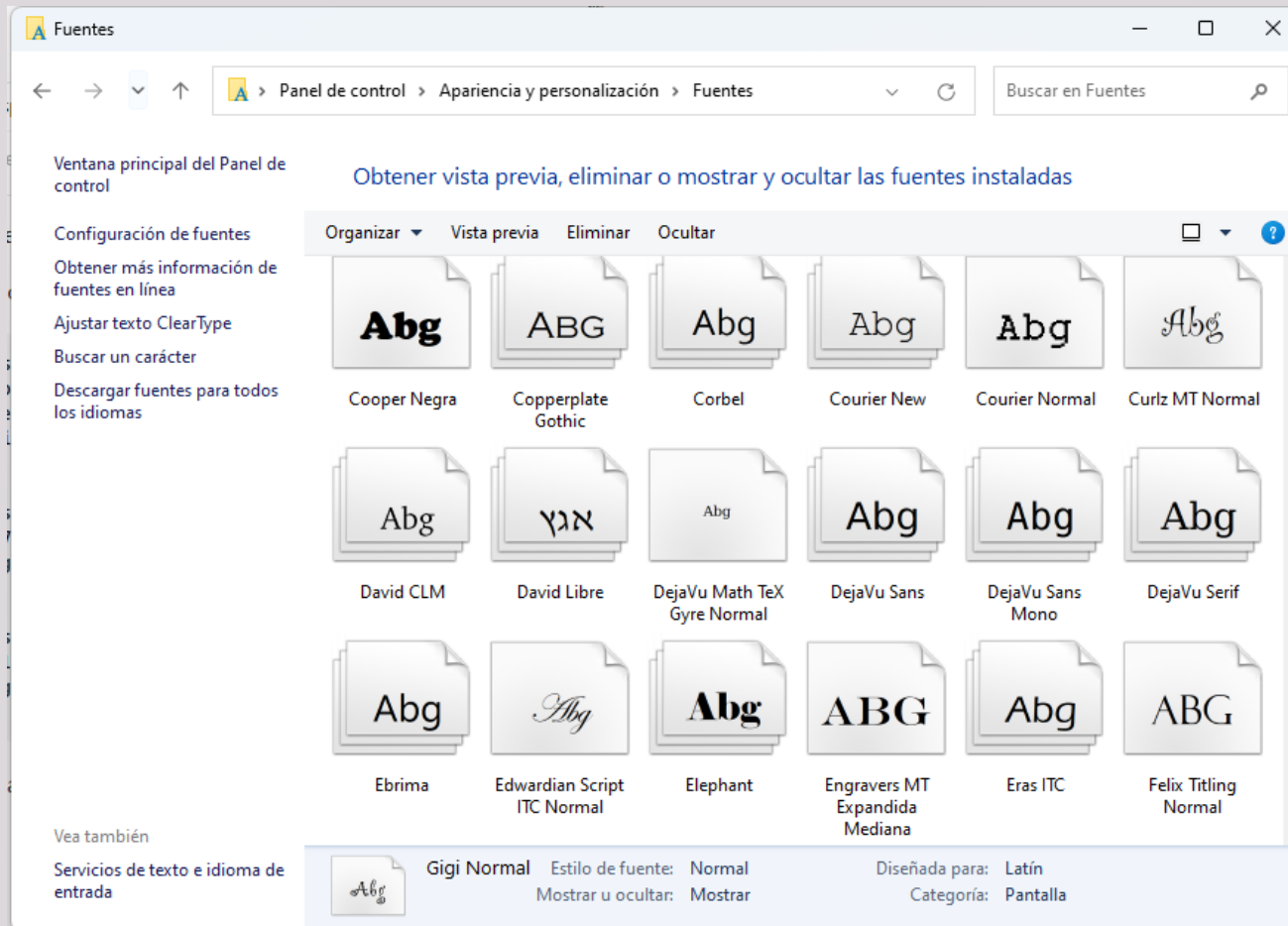
- Tiene extremos rectos en las letras, un diseño mucho más limpio y moderno. Mejora la legibilidad. Más adecuado para disléxicos.
- Ejemplos: Arial, Lucida Sans, Verdana

- Fuentes Monospace

- Tiene anchos iguales para todas las letras. Excelente para presentar ejemplos de código o información técnica.
- Ejemplos: Consolas, "Courier New", Inconsolata



# FONT



# FONT-FAMILY

- Permite especificar el tipo de letra.
- El valor de la propiedad es el nombre del tipo de letra que desees usar.
- Los usuarios que visitan tu sitio deben tener instalados los tipos de letra que estás especificando para poder verlos.
- Debido a esto, debes especificar tipos de letra "alternativos" con comas, finalizando con un nombre genérico de fuente para el tipo de letra que estás utilizando.

```
h1, h2 {  
    font-family: "Courier New", Courier, monospace;  
}
```

\*Si no está disponible "*Courier New*" utiliza *Courier* y si no, el navegador utilizará el tipo de letra monoespacio por defecto.



# FONT

La última opción de font-family no es una tipografía, sino una «web-safe fonts» (fuente segura).

No es una tipografía, es una **palabra clave** con la que se designa una categoría de tipografías.

Esto indica al navegador que debe buscar una tipografía instalada en el sistema que entre dentro de esa categoría.

- Mejores fuentes para web:
- Arial (sans-serif)
- Verdana (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

Fuente	Significado	Fuentes de ejemplo
serif	Tipografía con serifa	Times New Roman, Georgia...
sans-serif	Tipografía sin serifa	Arial, Verdana, Tahoma...
cursive	Tipografía en cursiva	Sanvito, Corsiva...
fantasy	Tipografía decorativa	Critter, Cottonwood...
monospace	Tipografía monoespaciada	Courier, Courier New...

# FONT

## Fuente estándar

Times New Roman ▼

16: Benjamín pidió una bebida de kiwi y fresa. Noé, sin vergüenza, la más exquisita champaña del menú.

## Fuente Serif

Times New Roman ▼

16: Benjamín pidió una bebida de kiwi y fresa. Noé, sin vergüenza, la más exquisita champaña del menú.

## Fuente Sans-serif

Arial ▼

16: Benjamín pidió una bebida de kiwi y fresa. Noé, sin vergüenza, la más exquisita champaña del menú.

- Mejores fuentes para web:
- Arial (sans-serif)
- Verdana (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

# FONT

- **@font-face** Se utiliza para especificar la URL de un archivo de fuente que el navegador descargará.
  - Requiere un atributo "src" para especificar la ruta del archivo de la fuente.
  - Requiere un atributo "format" para proporcionar el nombre del formato de la fuente. (ttf es truetype)
  - Además, necesitas un atributo "font-family" para dar el nombre de la fuente. Es necesario proporcionar múltiples formatos para distintos navegadores. Debes aplicar la fuente en una declaración "font-family" en una regla por separado.

```
@font-face {  
  font-family: 'RUBIK';  
  src: url(RubikBubbles-Regular.ttf) format('truetype');  
}  
  
p {  
  font-family: 'RUBIK', sans-serif;  
}
```

# FONT

Propiedad	Descripción	Valores
font-family	tipo de letra (fuente)	[ nombre-fuente   familia-genérica ] [, nombre-fuente   familia-genérica ]*
font-size	tamaño	tamaño-absoluto   tamaño-relativo   distancia   porcentaje
font-weight *	grosor del trazo (negrita)	normal   bold   bolder   lighter   100   200   300   400   500   600   700   800   900
font-style	inclinación (cursiva)	normal   italic   oblique
font-variant	variante	normal   small-caps   initial   inherit;

# FONT-WEIGHT

\*Algunas tipografías carecen de determinados pesos y sólo tienen ciertos pesos específicos.

Esto es muy sencillo de ver en

[Google Fonts](#)

## Font-weight test case **Firefox 3.1b2**

See [acceder al blog post](#) for more details.

100 · why pangolins dream of quiche  
200 · why pangolins dream of quiche  
300 · why pangolins dream of quiche  
400 · why pangolins dream of quiche  
500 · why pangolins dream of quiche  
**600 · why pangolins dream of quiche**  
**700 · why pangolins dream of quiche**  
**800 · why pangolins dream of quiche**  
**900 · why pangolins dream of quiche**

This is a Light sentence with a Bold word in the middle.



[Tipografías en CSS - CSS en español - Lenguaje CSS](#)

Font weight test: <https://codepen.io/carlosvs/pen/raaJZo>

# FONT-WEIGHT

- Permite poner negrita. Los valores son bold y normal;

```
.importante{  
    font-weight:bold;  
}
```

# FONT-STYLE

- Itálica: italic, oblique, and normal

```
.credits {  
    font-style: italic;  
}
```

[Tipografías en CSS - CSS  
en español - Lenguaje  
CSS](#)

# FONT

- ATAJO:
- Resumir las propiedades en una:
- Los valores de font-size y font-family son obligatorios. Si falta uno de los otros valores, se utiliza su valor predeterminado.

- font-style
- font-variant
- font-weight
- font-size
- font-family

```
selector {  
    font: [estilo] [variante] [peso] [tamaño]/[altura-de-línea]  
         [tipo de letra], [otro tipo de letra], ...;  
}
```

```
.container {  
    font: italic small-caps bold 16px Arial, Verdana, sans-serif;  
}
```



**TEXT**

...tanto, ei...  
...hor, em que nunca...  
...senhor, que fez subir os filh...  
...ael da terra do Egito;  
Mas: Vive o Senhor, que fez...  
...que trouxe a geração da casa...  
...el da terra do norte, e de to...  
...ras para onde os tinha arroj...  
...bitarão na sua terra.

### ***Contra os falsos profetas***

Quanto aos profetas. O me...  
...o está quebrantado dentro de...  
...los os meus ossos estremece...  
...no um homem embriagado, e...  
...homem vencido do vinho, p...  
...hor, e por causa das...



# WHITE-SPACE

- `&nbsp;`; `<!--` ensucia el código `-->`

Propiedad	Valor	Significado
white-space	normal   nowrap   pre   pre-line   pre-wrap	Comportamiento de los espacios
tab-size	numpx   num	Ancho de las tabulaciones (espacio o tamaño)

La diferencia entre pre-wrap y pre-line es que este último respeta literalmente los espacios que están antes del texto, mientras que si sobran después del texto, los suprime.

nowrap	Los espacios consecutivos se transforman en uno solo.	Ignora saltos de línea.
pre	Respetar y muestra literalmente los espacios.	Ignora saltos de línea.
pre-wrap	Respetar y muestra literalmente los espacios.	Se ajusta al contenedor.
pre-line	Respetar literalmente los espacios y suprime los espacios del final.	Se ajusta al contenedor.

# TEXT-DECORATION

Propiedad	Descripción	Valores	Ejemplo
text-align	alineación del texto	center   justify   left   right	text-align: center;
text-decoration	decoración del texto	none   blink   line-through   overline   underline	<pre>a{text-decoration: none;} h1 {   text-decoration: underline overline dotted red; } <a href="https://www.w3schools.com/cssref/pr_text_text-decoration.php">https://www.w3schools.com/cssref/ pr_text_text-decoration.php</a></pre>
text-transform	mayúsculas / minúsculas	none   capitalize   lowercase   uppercase	text-transform: uppercase;

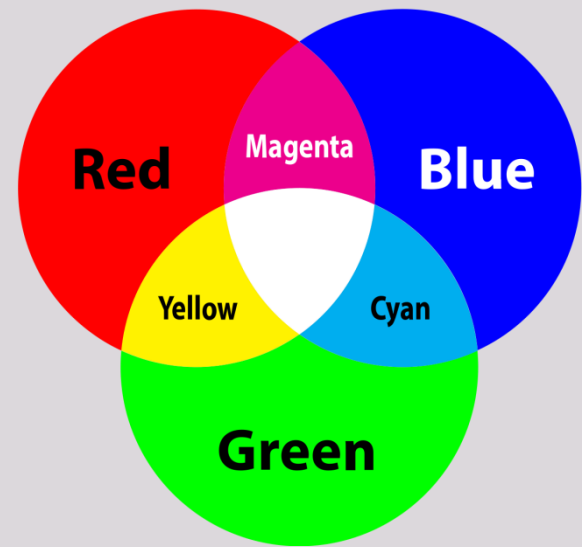
**COLOR**

# COLOR

**Se pueden indicar de 5 formas diferentes:**

- Nombres de colores predefinidos para varios navegadores
- Colores RGB
- Colores RGBA
- Colores HSL
- Colores HSLA
- Colores hexadecimales
- Colores hexadecimales con transparencia
- Con la palabra clave `currentcolor`

**RGB Colour Model**




# COLOR

## Nombres de colores predefinidos

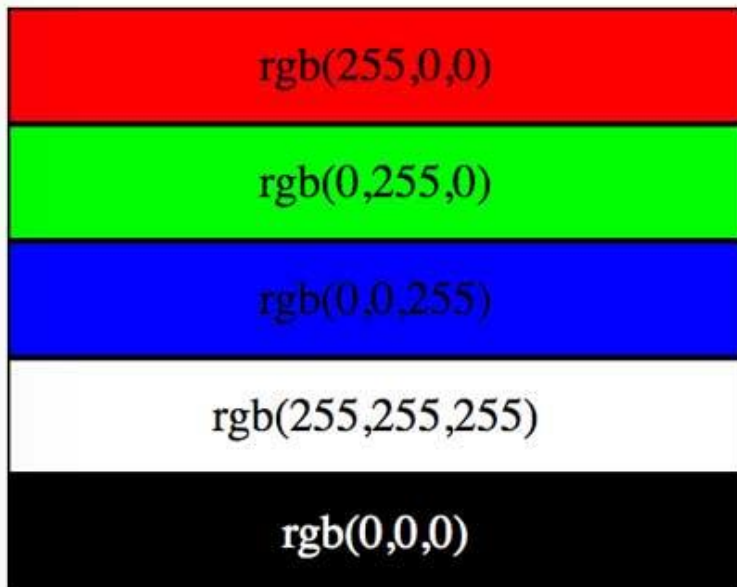
[https://www.w3schools.com/cssref/css\\_colors.php](https://www.w3schools.com/cssref/css_colors.php)

### CSS Colors

 black	 bisque	 forestgreen	 slategrey
 dimgray	 darkorange	 limegreen	 lightsteelblue
 dimgrey	 burlywood	 darkgreen	 cornflowerblue
 gray	 antiquewhite	 green	 royalblue
 grey	 tan	 lime	 ghostwhite
 darkgray	 navajowhite	 seagreen	 lavender
 darkgrey	 blanchedalmond	 mediumseagreen	 midnightblue
 silver	 papayawhip	 springgreen	 navy
 lightgray	 moccasin	 mintcream	 darkblue
 lightgrey	 orange	 mediumspringgreen	 mediumblue
 gainsboro	 wheat	 mediumaquamarine	 blue
 whitesmoke	 oldlace	 aquamarine	 slateblue
 white	 floralwhite	 turquoise	 darkslateblue
 snow	 darkgoldenrod	 lightseagreen	 mediumslateblue
 rosybrown	 goldenrod	 mediumturquoise	 mediumpurple
 lightcoral	 cornsilk	 azure	 rebeccapurple
 indianred	 gold	 lightcyan	 blueviolet
 brown	 lemonchiffon	 paleturquoise	 indigo
 firebrick	 khaki	 darkslategray	 darkorchid
 maroon	 palegoldenrod	 darkslategrey	 darkviolet
 darkred	 darkkhaki	 teal	 mediumorchid
 red	 ivory	 darkcyan	 thistle
 mistyrose	 beige	 aqua	 plum
 salmon	 lightyellow	 cyan	 violet
 tomato	 lightgoldenrodyellow	 darkturquoise	 purple
 darksalmon	 olive	 cadetblue	 darkmagenta
 coral	 yellow	 powderblue	 fuchsia
 orangered	 olivedrab	 lightblue	 magenta
 lightsalmon	 yellowgreen	 deepskyblue	 orchid
 sienna	 darkolivegreen	 skyblue	 mediumvioletred
 seashell	 greenyellow	 lightskyblue	 deeppink
 chocolate	 chartreuse	 steelblue	 hotpink
 saddlebrown	 lawngreen	 aliceblue	 lavenderblush
 sandybrown	 honeydew	 dodgerblue	 palevioletred
 peachpuff	 darkseagreen	 lightslategray	 crimson
 peru	 palegreen	 lightslategrey	 pink
 linen	 lightgreen	 slategray	 lightpink

# RGB

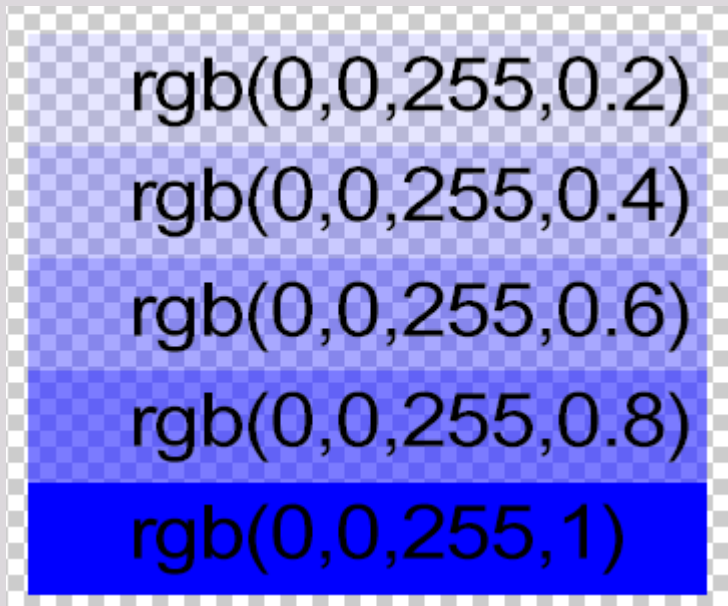
- El modelo RGB consiste en definir un color indicando que cantidad de color rojo, verde y azul se debe mezclar para obtener el color.
- `rgb (red, green, blue)`



```
.caja-1{background-color:rgb(255,0,0);}
.caja-2{background-color:rgb(0,255,0);}
.caja-3{background-color:rgb(0,0,255);}
.caja-4{background-color:rgb(255,255,255);}
.caja-5{background-color:rgb(0,0,0);}
```

# RGBA

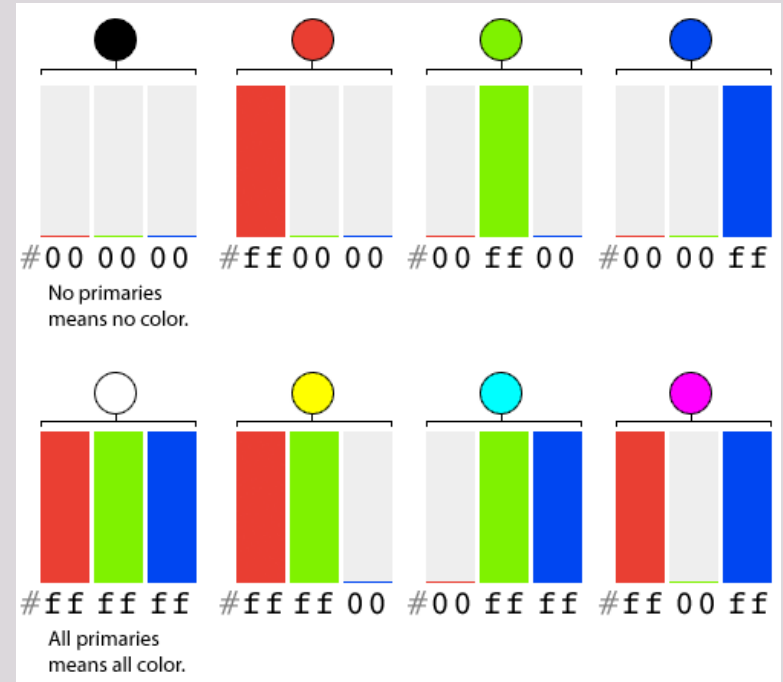
- El modelo RGBA consiste en definir un color indicando que cantidad de color rojo, verde y azul se debe mezclar para obtener el color. Además se añade el canal Alpha, que indica el nivel de transparencia. 0 es transparencia total.
- `rgba(red, green, blue, alpha)`



```
.caja-1{background-color:rgba(0,0,255,.2);}
.caja-2{background-color:rgba(0,0,255,.4);}
.caja-3{background-color:rgba(0,0,255,.6);}
.caja-4{background-color:rgba(0,0,255,.8);}
.caja-5{background-color:rgba(0,0,255,1);}
```

# HEXADECIMALES

- En este método, se hace uso del sistema numérico hexadecimal. En el sistema hexadecimal se utilizan 16 símbolos (de ahí su nombre): del 0 al 9 y de la A a la F.
- `#rrggbb`
- El valor numérico de cada componente se indica en el sistema numérico hexadecimal. (00 es nada de color y FF es el total del color)
- Para obtener el color completo, se concatenan los valores de las componentes RGB y se añade el prefijo #.
- Se puede comprimir: `#rgb`
  - `#AAA = #AAAAAA`
  - `#FFF = #FFFFFF`
  - `#A0F = #AA00FF`
  - `#369 = #336699`



```
body { background-color: #FFF; }  
h1, h2, h3 { color: #F00; }
```

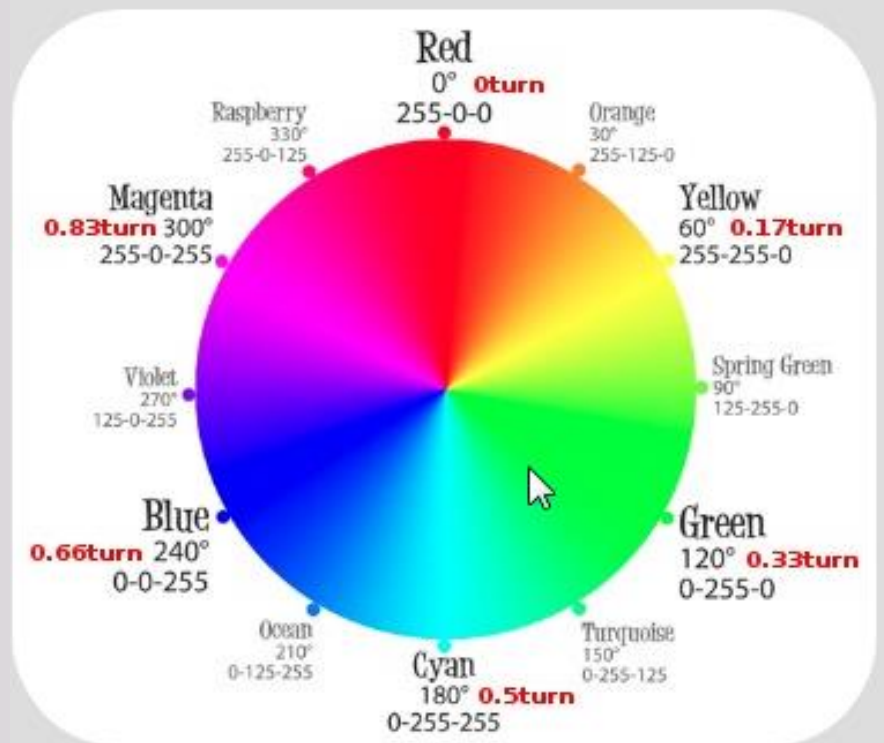


# HSL

- Hue, Saturation, Lighness es decir, tono, saturación y brillo.
- Los colores HSL se construyen especificando 3 componentes:
- una tonalidad o matiz de color (Hue, en inglés)
- un grado de saturación o contraste (S) del 0% al 100% (gris, color)
- un grado de luminosidad o brillo (L) del 0% al 100% (negro/blanco)

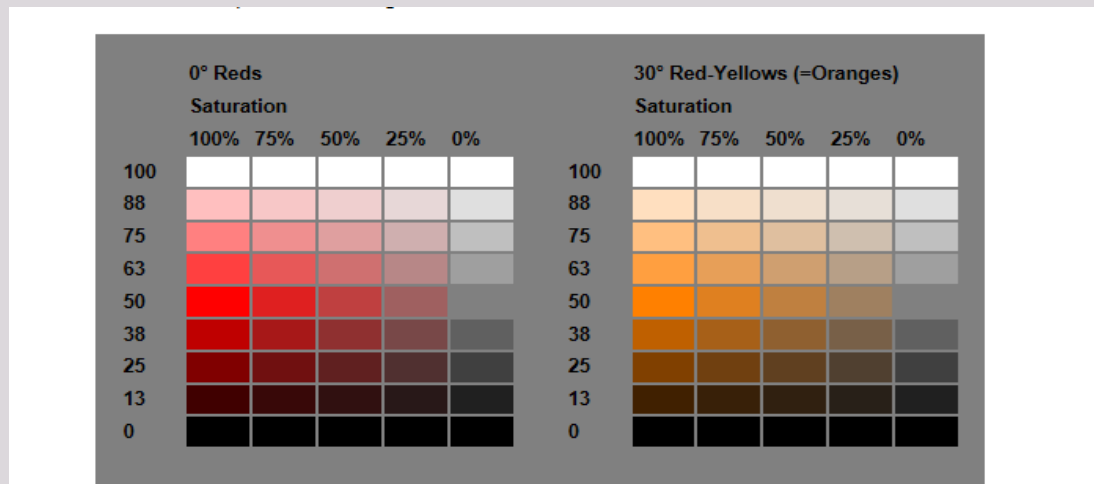
# HSL

- Hue, Saturation, Lighness es decir, tono, saturación y brillo.
- Los colores HSL se construyen especificando 3 componentes:
- una tonalidad o matiz de color (Hue, en inglés)
  - 0, sería el **rojo**.
  - 120, sería el **verde**.
  - 240, sería el **azul**.
  - 360, volvería a ser **rojo**.



# HSL

- Hue, Saturation, Ligthness es decir, tono, saturación y brillo.
- Los colores HSL se construyen especificando 3 componentes:
- una tonalidad o matiz de color (Hue, en inglés)
- un grado de saturación o contraste (S) del 0% al 100% (gris, color)



- un grado de luminosidad o brillo (L) del 0% al 100% (0=negro, 50% normal, 100%=blanco)
- [https://www.w3schools.com/Css/css\\_colors\\_hsl.asp](https://www.w3schools.com/Css/css_colors_hsl.asp)

# HSLA

- HSL con canal Alpha (transparencia)
- `hsla(0, 100%, 50%, 0.5)`

# SOMBRAS



# TEXT-SHADOW

**Texto con sombra**

```
h1 { text-shadow: green 2px 2px 4px };
```

Lleva cuatro parámetros:

- Color: expresado de diferentes maneras, es además opcional.
- X e Y: los dos valores siguientes son obligatorios e indican la distancia de la sombra.
- Suavizado: el último valor es opcional y especifica lo difuminada que se mostrará la sombra

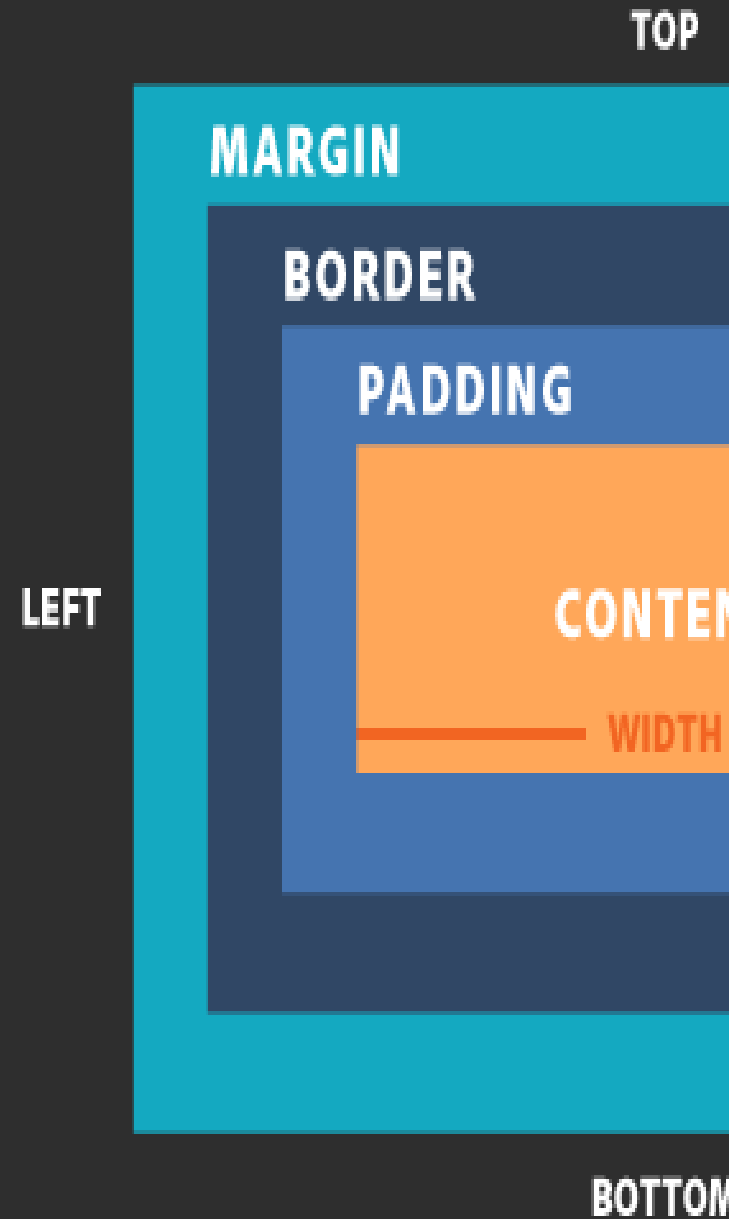
# BOX-SHADOW

```
<div class="sombra">  
  <p>Contenedor con sombra</p>  
</div>
```

```
.sombra{  
    box-shadow: 8px 8px 10px red;  
    width:30%;  
}
```

Contenedor con sombra

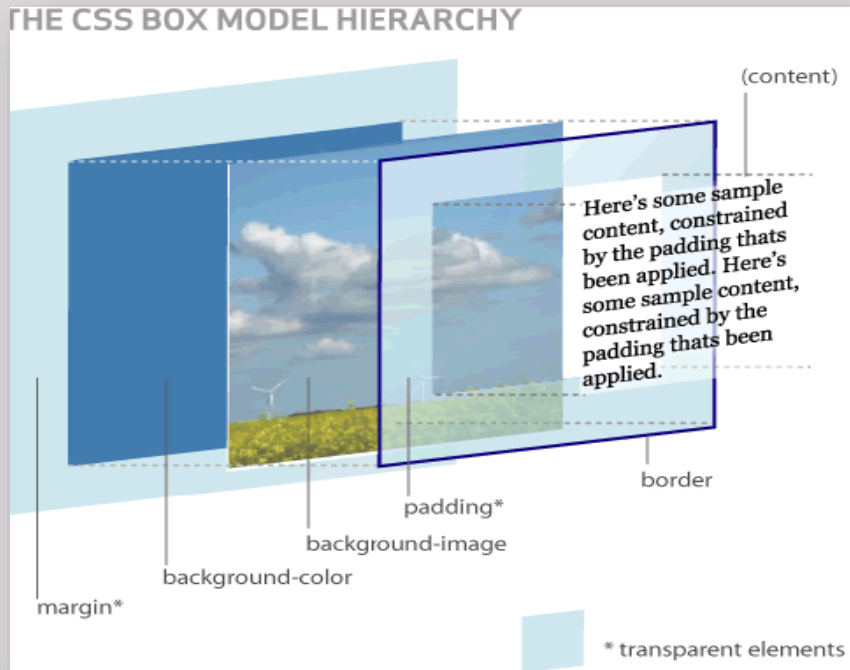
# BOX-MODEL





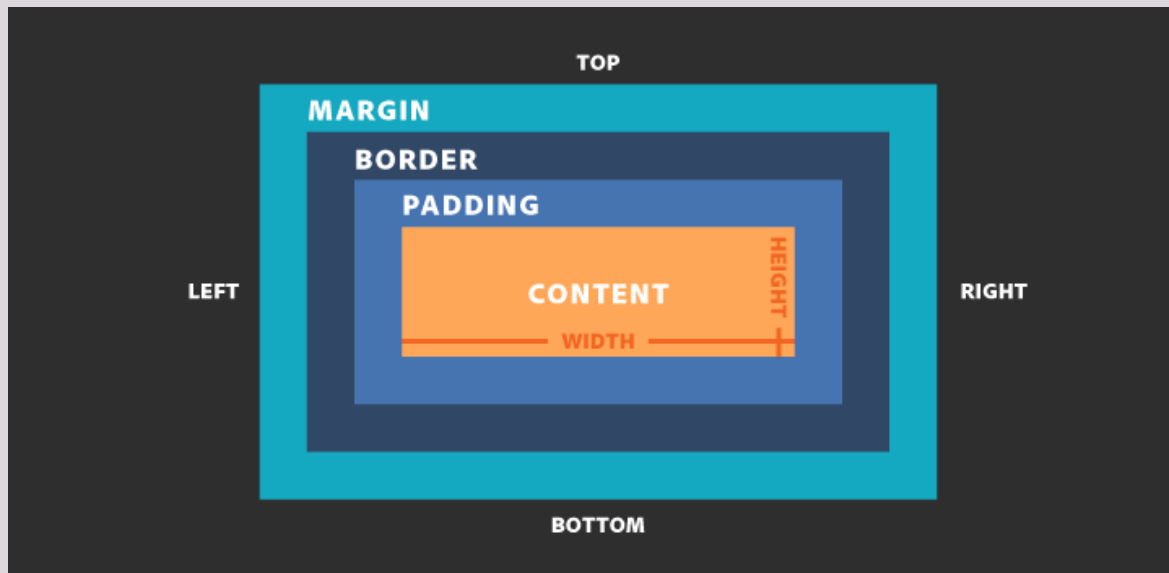
# BOX-MODEL

- El "box model" es el comportamiento de CSS que hace que todos los elementos incluidos en una página HTML se representen mediante cajas rectangulares. CSS permite controlar el aspecto de todas las cajas.



# BOX-MODEL

- El "box model" es el comportamiento de CSS que hace que todos los elementos incluidos en una página HTML se representen mediante cajas rectangulares. CSS permite controlar el aspecto de todas las cajas.



# WIDTH Y HEIGHT

- **Width:** Valores <medida> | <porcentaje> | auto | inherit
- Se aplica a: Todos los elementos, salvo los elementos en línea que no sean imágenes, las filas de tabla y los grupos de filas de tabla
- Valor inicial auto. Inherit es la heredada del padre
- **Height:** igual que width

Este es un contenedor padre

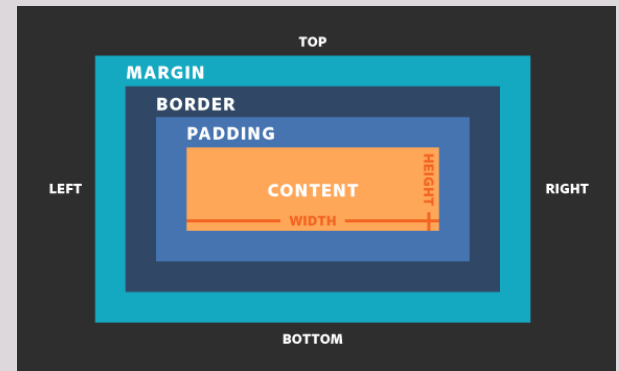
Este es un contenedor hijo 1

Este es un contenedor hijo 2

Este es un párrafo con dentro

```
.padre{  
  border: 2px solid blue;  
  width:80%;  
  height: 200px;  
}  
.hijo{  
  border: 2px solid red;  
  width:30%;  
  height:20%;  
}
```

# MARGIN



- CSS define cuatro propiedades para controlar cada uno de los márgenes horizontales y verticales de un elemento.
  - `margin-top`
  - `margin-right`
  - `margin-bottom`
  - `margin-left`
- ```
div img {  
  margin-top: .5em;  
  margin-bottom: .5em;  
  margin-left: 1em;  
  margin-right: .5em;  
}
```
- Valores `<medida>` | `<porcentaje>` | `auto` | `inherit`
  - Se aplica a: Todos los elementos, salvo `margin-top` y `margin-bottom` que sólo se aplican a los elementos de bloque y a las imágenes.
  - El margen entre los elementos adyacentes será el mayor de los dos márgenes.

# MARGIN

## Shorthand

- Si se indican los cuatro valores: (ORDEN AGUJAS RELOJ): superior, derecho, inferior E izquierdo.
- Si solo se indica un valor: todos iguales.
- Si se indican dos valores: primer valor: superior e inferior,. Segundo valor: izquierdo y derecho.
- Si se indican tres valores,: primer valor: superior, tercero: inferior segundo: izquierdo y derecho.

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

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

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

# MARGIN

Shorthand

```
/* Aplica a todos los cuatro lados */  
margin: 1em;
```

```
/* Vertical | Horizontal */  
margin: 5% auto;
```

```
/* Arriba | Horizontal | Abajo */  
margin: 1em auto 2em;
```

```
/* Arriba | Derecha | Abajo | Izquierda  
(ORDEN de Las AGUJAS deL RELOJ */  
margin: 2px 1em 0 auto;
```

# MARGIN

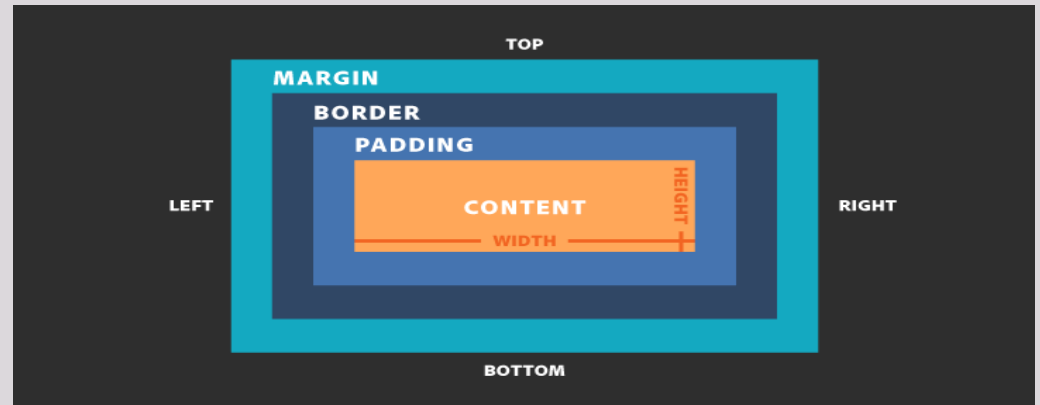
```
div img {  
  margin: .5em;  
  margin-left: 1em;  
}
```

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

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

\*Para centrar algo horizontalmente en navegadores modernos, usa `display: flex; justify-content: center; .`  
Sin embargo, en navegadores antiguos como IE8-9, flexbox no está disponible. Para poder centrar un elemento horizontalmente con respecto a su contenedor, usa `margin: 0 auto;`

# PADDING



- Espacio entre el contenido y el borde.
- Valores <medida> | <porcentaje> | inherit
- Se aplica a: Todos los elementos excepto algunos elementos de tablas como grupos de cabeceras y grupos de pies de tabla
- Funciona igual que el margin.

```
.hijo{  
  border: 2px solid red;  
  width:30%;  
  height:20%;  
  margin:1em;  
}  
.hijo2{  
  border: 2px solid pink;  
  width:30%;  
  height:20%;  
  margin:1em 2em;  
  padding:2em;  
}
```

Este es un contenedor hijo 1

Este es un contenedor hijo 1

Este es un contenedor hijo 2

**Shorthand**



# BORDER

| Bordes                      |                               |                                                                                               |                                                                                                                                                                                                          |
|-----------------------------|-------------------------------|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Propiedad                   | Descripción                   | Valores                                                                                       | Ejemplo                                                                                                                                                                                                  |
| <b><u>border</u></b>        | cuatro bordes simultáneamente | border-color    border-width    border-style                                                  | border: 2px solid red;                                                                                                                                                                                   |
| <b><u>border-radius</u></b> | esquinas redondeadas          | [ distancia   porcentaje ] {1, 4}                                                             | border-radius: 10px;<br>Border radius: 15px 20px 25 px 30px;<br>border-top-right-radius: 15px;<br>border-bottom-right-radius: 20px;<br>border-bottom-left-radius: 25px;<br>border-top-left-radius: 30px; |
| <b><u>border-color</u></b>  | color de los bordes           | [ color   transparent ] {1, 4}                                                                | border-color: red green blue yellow;<br>border-color:red;                                                                                                                                                |
| <b><u>border-width</u></b>  | grosor de los bordes          | [ medium   thick   thin   distancia ] {1, 4}                                                  | border-width: 2px 1px 3px 4px;<br>border-width: 2px;                                                                                                                                                     |
| <b><u>border-style</u></b>  | estilos de los bordes         | [ none   hidden   dashed   dotted   double   groove   inset   outset   ridge   solid ] {1, 4} | border-style: solid dotted double dashed;<br>border-style: solid;                                                                                                                                        |

# BORDER-WIDTH

- **Valores** ( <medida> | thin | medium | thick ) | inherit
- **border-top-width** Anchura del borde superior
- **border-right-width** Anchura del borde derecho
- **border-bottom-width** Anchura del borde inferior
- **border-left-width** Anchura del borde izquierdo

## Shorthand

- **border-width**

# BORDER-COLOR

- **Valores** (<color> | transparent | inherit
- **border-top-color** Color del borde superior
- **border-right-color** Color del borde derecho
- **border-bottom-color** Color del borde inferior
- **border-left-color** Color del borde izquierdo

## Shorthand

- **border-color**

# BORDER-STYLE

- **border-top-style** Estilo del borde superior
- **border-right-style** Estilo del borde derecho
- **border-bottom-style** Estilo del borde inferior
- **border-left-style** Estilo del borde izquierdo
- 
- **Valores** none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit
- **Se aplica a** Todos los elementos
- **Valor inicial** none
- **Descripción** Establece el estilo de cada uno de los cuatro bordes de los elementos

**Shorthand**

- **border-style**

None

Hidden

Dotted

Dashed

Solid

Double

Groove

Ridge

Inset

Outset

# BORDER

## Shorthand

- **border-top** Estilo completo del borde superior
- **border-right** Estilo completo del borde derecho
- **border-bottom** Estilo completo del borde inferior
- **border-left** Estilo completo del borde izquierdo
- 
- **Valores**
- ( <medida\_borde> || <color\_borde> || <estilo\_borde> ) | inherit
- **Se aplica a** Todos los elementos
- **Valor inicial** -
- **Descripción** Establece el estilo completo de cada uno de los cuatro bordes de los elementos

```
h1 {border-bottom: solid red;}
```

# BORDER

## Shorthand

```
div {  
  border-top: 1px solid red;  
  border-right: 1px solid red;  
  border-bottom: 1px solid red;  
  border-left: 1px solid red;  
}  
/* es lo mismo que esto */  
div {  
  border: 1px solid red;  
}
```

```
h1 {border-bottom: solid red;}
```

# BORDER-RADIUS

```
.bordesRedondeados{  
  border: 2px solid blue;  
  border-radius: 25px;  
}
```

```
<div class="bordesRedondeados">  
  <p>Bordes redondeados</p>  
</div>
```

Bordes redondeados

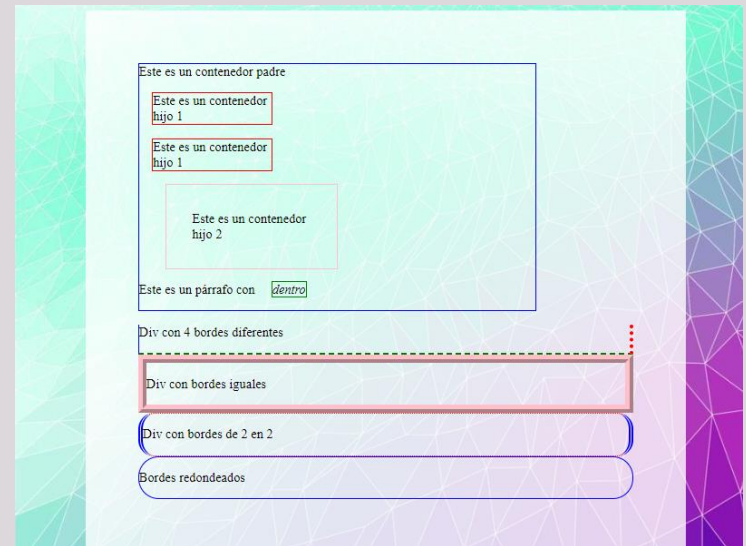
# FONDOS

Fondos			
Propiedad	Descripción	Valores	Ejemplo
<b>background</b>	propiedad compuesta	background-attachment    background-color    background-image    background-position    background-repeat	background: #f0f0f0 url('imagen-fondo.jpg') center top no-repeat fixed;
<b>background-attachment</b>	ligadura de la imagen	fixed   scroll	background-attachment: fixed;
<b>background-color</b>	color de fondo	transparent   color	background-color: #f0f0f0;
<b>background-image</b>	imagen de fondo	none   uri	background-image: url('imagen-fondo.jpg');
<b>background-position</b>	posición de la imagen de fondo	[ [ left   center   right   distancia   porcentaje ] [ top   center   bottom   distancia   porcentaje ] ? ]   [ [ left   center   right ]    [ top   center   bottom ] ]	background-position: center top;
<b>background-repeat</b>	repetición de la imagen de fondo	no-repeat   repeat   repeat-x   repeat-y   space   round	background-repeat: no-repeat;



# BACKGROUND-COLOR BACKGROUND-IMAGE

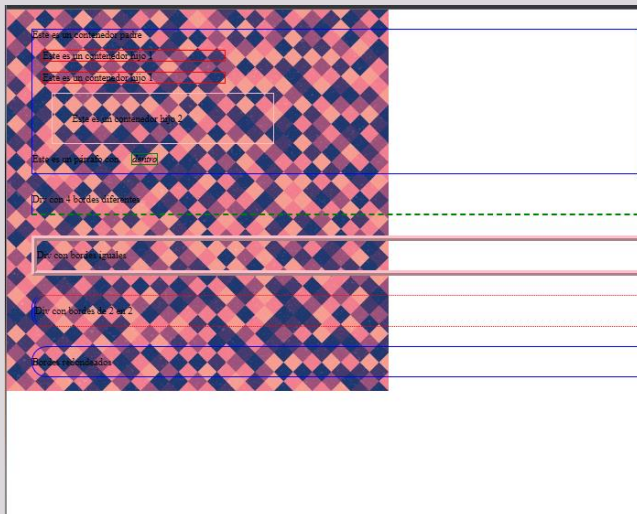
- La imagen se muestra delante del color, por lo que solamente si la imagen contiene zonas transparentes es posible ver el color de fondo.
- **Problema:** El texto u otros elementos pueden ser difíciles de leer si la imagen de fondo es muy llamativa.



```
body{  
  background-color: #F5F5F5;  
  background-image: url("img/fondoColores.jpg");  
}  
.contenedor{  
  background-color: rgba(255, 255, 255, 0.8);  
  margin: 0em 5em;  
  padding: 4em;  
}
```

# BACKGROUND-REPEAT

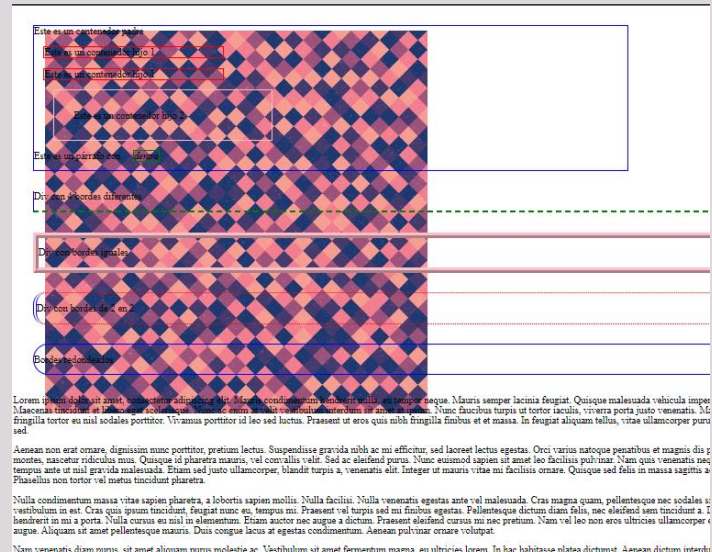
- Repetición de la imagen de fondo
- **Valores** repeat | repeat-x | repeat-y | no-repeat | inherit
- Repeat (por defecto)
- no-repeat muestra una sola vez la imagen y no se repite en ninguna dirección.
- repeat-x repite la imagen sólo horizontalmente
- repeat-y repite la imagen solamente de forma vertical.



# BACKGROUND-POSITION

- left top
- left center
- left bottom
- right top
- right center
- right bottom
- center top
- center center
- center bottom
- (center por defecto)
- x% y% (posición horizontal y posición vertical)
- xpos ypos
- initial (por defecto)
- inherit (Heredado)

```
body{
    background-image: url("img/patron.jpg");
    background-repeat: no-repeat;
    background-position: 10% 10%;
}
```



# BACKGROUND-ATTACHMENT

- Controla si la imagen de fondo de un elemento se desplaza con respecto al contenido del elemento o si permanece fija en su posición mientras el contenido se desplaza.
- Valores: fixed | scroll | local | initial | inherit
- Valor inicial: scroll

```
body{  
  background-image: url("img/fondoColores.jpg");  
  background-attachment: fixed;  
}
```

# BACKGROUND

**Shorthand**

```
body {  
    background: #222d2d url("img/patron.jpg") repeat-x 0 0;  
}
```

*/\* La propiedad shorthand anterior es equivalente a las siguientes propiedades \*/*

```
body {  
    background-color: #222d2d;  
    background-image: url("img/patron.jpg");  
    background-repeat: repeat-x;  
    background-position: 0 0;  
}
```

# LISTAS

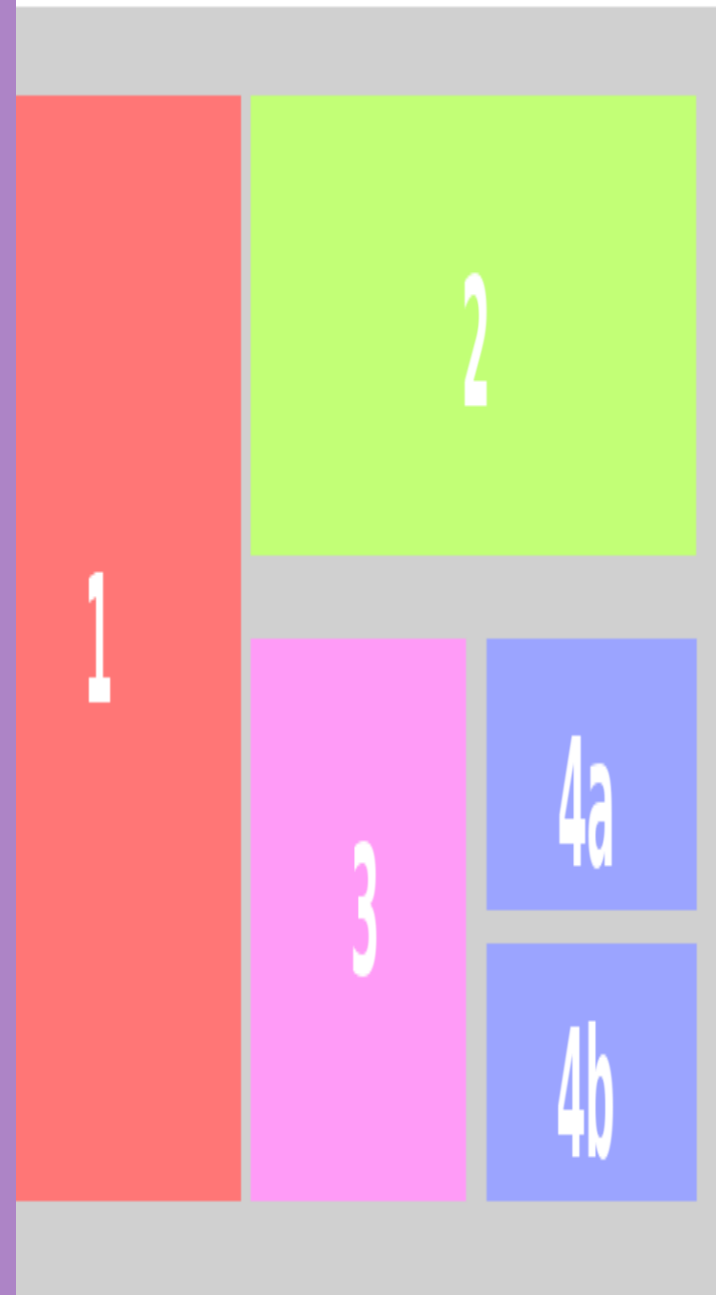
```
type="a">  
li>Elemento 1  
li>Elemento 2  
li>Elemento 3  
li>Elemento 4
```

```
type="I">  
li>Elemento 1  
li>Elemento 2  
li>Elemento 3  
li>Elemento 4
```

# LISTAS

Property	Description
<b>list-style</b>	Establece todas las propiedades para una lista en una sola declaración
<b>list-style-image</b>	Especifica una imagen como el marcador del elemento de lista
<b>list-style-position</b>	Especifica la posición de los marcadores de elementos de lista (viñetas)
<b>list-style-type</b>	Especifica el tipo de marcador de elemento de lista

# POSICION AMIENTO





# ELEMENTOS LÍNEA Y BLOQUE

- Los **elementos de bloque** ("block elements" en inglés) siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea.
  - address, blockquote, center, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, isindex, menu, noframes, noscript, ol, p, pre, table, ul, dd, dt, frameset, li, tbody, td, tfoot, th, thead, tr.
- Los **elementos en línea** ("inline elements" en inglés) no empiezan necesariamente en nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos.
  - a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.
- Los siguientes elementos pueden ser en línea y de bloque según las circunstancias:
- button, del, iframe, ins, map, object, script.

Los párrafos son elementos de bloque.

[Los enlaces son elementos en línea](#)

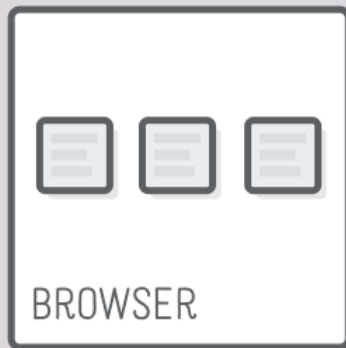
Dentro de un párrafo, [los enlaces](#) siguen siendo elementos en línea.

# POSITION

- Valores:
- **static**: corresponde al posicionamiento normal o estático. Si se utiliza este valor, se **ignoran los valores de las propiedades top, right, bottom y left** que se verán a continuación.
- **relative**: corresponde al posicionamiento relativo. El desplazamiento de la caja se controla con las propiedades top, right, bottom y left.
- **absolute**: corresponde al posicionamiento absoluto. El desplazamiento de la caja también se controla con las propiedades top, right, bottom y left, pero su interpretación es mucho más compleja, ya que el origen de coordenadas del desplazamiento depende del posicionamiento de su elemento contenedor.
- **fixed**: corresponde al posicionamiento fijo. El desplazamiento se establece de la misma forma que en el posicionamiento absoluto, pero en este caso el elemento permanece inamovible en la pantalla.
- **sticky**: corresponde al posicionamiento adhesivo. alterna entre relative y fixed, dependiendo de la posición de desplazamiento. Se coloca en relación hasta que se alcanza una posición de desplazamiento determinada en la ventana gráfica; luego se "pega" en su lugar (como posición: fija).

# POSITION

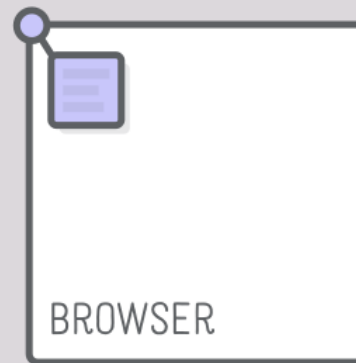
STATIC



RELATIVE



ABSOLUTE

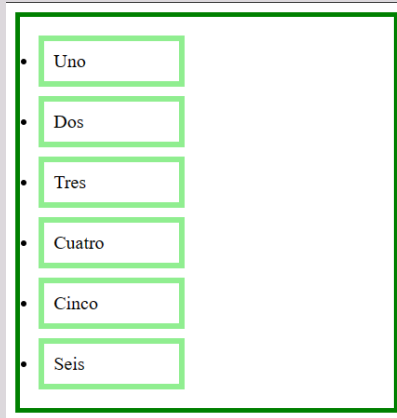


FIXED



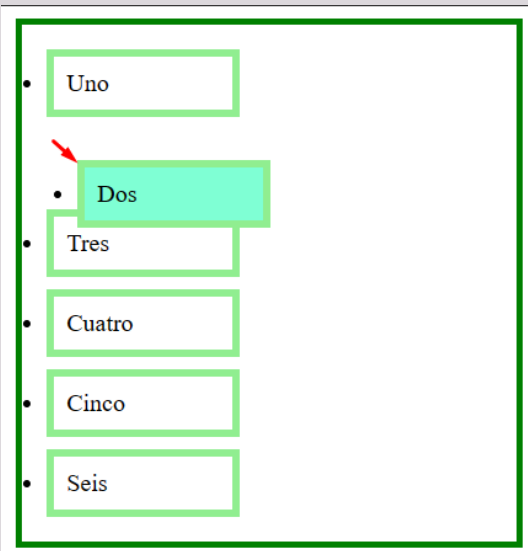
# POSITION

- Position:static



```
<ul>
  <li class="primero">Uno</li>
  <li class="segundo">Dos</li>
  <li class="tercero">Tres</li>
  <li class="cuarto">Cuatro</li>
  <li class="quinto">Cinco</li>
  <li class="sexto">Seis</li>
  <li class="septimo">Siete</li>
  <li class="octavo">Ocho</li>
</ul>
```

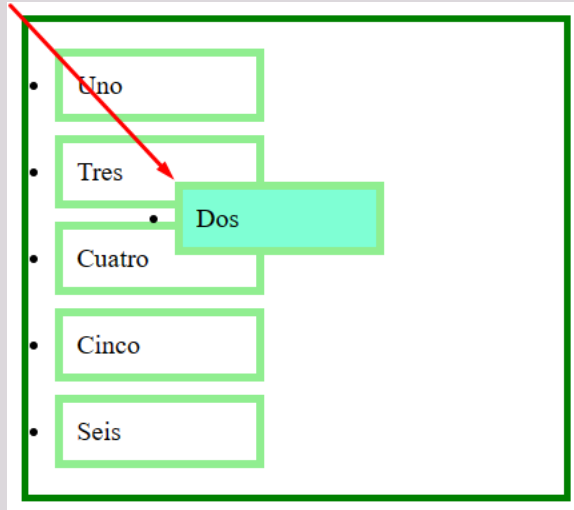
- Position:relative



```
ul{
  border: 4px solid green;
  margin: 0.5em;
  padding:0.51em;
  margin:0.5em;
}
li{ /* son en bloque */
  /* display inline | block | inherit | initial
  /none*/
  border: 5px solid lightgreen;
  margin:0.5em;
  padding:0.5em;
  width: 100px;
}
.segundo{
  top:20px;
  left:20px;
  position: relative;
  background-color: aquamarine;
}
```

# POSITION

- Position:absolute

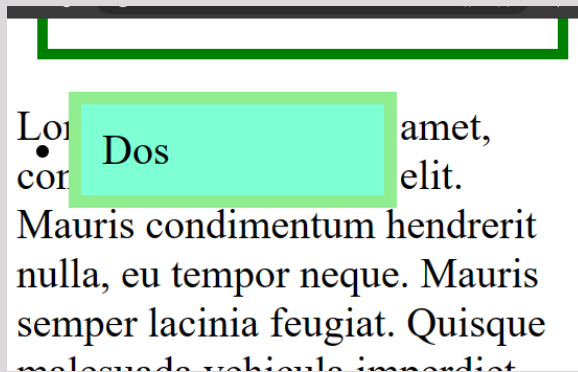


```
<ul>
  <li class="primero">Uno</li>
  <li class="segundo">Dos</li>
  <li class="tercero">Tres</li>
  <li class="cuarto">Cuatro</li>
  <li class="quinto">Cinco</li>
  <li class="sexto">Seis</li>
  <li class="septimo">Siete</li>
  <li class="octavo">Ocho</li>
</ul>
```

```
ul{
  border: 4px solid green;
  margin: 0.5em;
  padding:0.51em;
  margin:0.5em;
}
li{ /* son en bloque */
  /* display inline | block | inherit | initial
  |none*/
  border: 5px solid lightgreen;
  margin:0.5em;
  padding:0.5em;
  width: 100px;
}
.segundo{
  top:100px;
  left:100px;
  position: absolute;
  background-color: aquamarine;
}
```

# POSITION

- Position: fixed



```
<ul>
  <li class="primero">Uno</li>
  <li class="segundo">Dos</li>
  <li class="tercero">Tres</li>
  <li class="cuarto">Cuatro</li>
  <li class="quinto">Cinco</li>
  <li class="sexto">Seis</li>
  <li class="septimo">Siete</li>
  <li class="octavo">Ocho</li>
</ul>
```

```
ul{
  border: 4px solid green;
  margin: 0.5em;
  padding:0.51em;
  margin:0.5em;
}
li{ /* son en bloque */
  /* display inline | block | inherit | initial
  |none*/
  border: 5px solid lightgreen;
  margin:0.5em;
  padding:0.5em;
  width: 100px;
}
.segundo{
  top:100px;
  left:100px;
  position: fixed;
  background-color: aquamarine;
}
```

# POSITION:

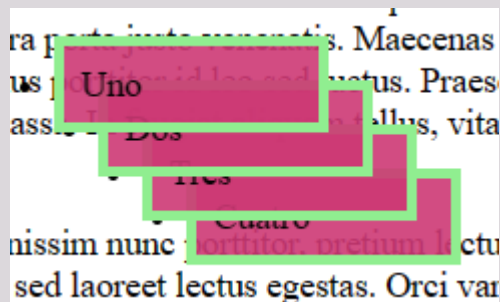
- sticky

```
.segundo{  
  top:20px;  
  left:20px;  
  position: sticky;  
  position: -webkit-sticky; /* Safari */  
  background-color: aquamarine;  
}
```

# 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.

```
<ul class="l4">
  <li class="primero">Uno</li>
  <li class="segundo">Dos</li>
  <li class="tercero">Tres</li>
  <li
    class="cuarto">Cuatro</li>
</ul>
```



```
ul.l4{
  border: 4px solid rgb(205, 57, 116);
  margin:1em;
  padding:1em;
  margin:1em;
}
.l4 li{
  position: absolute;
  background-color: rgba(205, 57, 116, .9);
}
.l4 .primero{
  clear: left;
  top:500px;
  left: 200px;
  z-index: 4;
}
.l4 .segundo{
  top:520px;
  left: 220px;
  z-index: 3;
}
.l4 .tercero{
  top:540px;
  left: 240px;
  z-index: 2;
}
.l4 .cuarto{
  top:560px;
  left: 260px;
  z-index: 1;
}
```



# FLOAT

- El posicionamiento flotante es el más difícil de comprender. Está siendo desplazado por flex y grid.
- Cuando una caja se posiciona con el modelo de posicionamiento flotante, automáticamente se convierte en una caja flotante, lo que significa que se desplaza hasta la zona más a la izquierda o más a la derecha de la posición en la que originalmente se encontraba.
- Valores: left | right | none | inherit

```
img{  
  width: 20%;  
  float:left;  
  margin:2em;  
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris condimentum hendrerit nulla, eu tempor neque. Mauris semper lacinia feugiat. Quisque malesuada vehicula imperdiet. Maecenas tincidunt et libero eget scelerisque. Nunc ac enim at velit vestibulum interdum sit amet at ipsum. Nunc faucibus turpis ut tortor iaculis, viverra porta justo venenatis. Maecenas fringilla tortor eu nisl sodales porttitor.

Vivamus porttitor id leo sed luctus. Praesent ut eros quis nibh fringilla finibus et et massa. In feugiat aliquam tellus, vitae ullamcorper purus mattis sed.

# CLEAR

```
<ul class="l1">
  <li class="primero">Uno</li>
  <li class="segundo">Dos</li>
  <li class="tercero">Tres</li>
  <li class="cuarto">Cuatro</li>
  <li class="quinto">Cinco</li>
  <li class="sexto">Seis</li>
  <li class="septimo">Siete</li>
  <li class="octavo">Ocho</li>
</ul>
```

• Uno

• Dos

• Tres • Cuatro • Cinco

• Seis

• Siete

• Ocho

```
.l1 li{
  border: 2px solid red;
  margin:1em;
  padding:1em;
}
ul.l1{
  border: 4px solid pink;
  margin:1em;
  padding:1em;
  margin:1em;
}
.l1 .tercero,.l1 .cuarto,.l1
.quinto{
  float:left;
}
.l1 .sexto{
  clear: left;
}
```

# DISPLAY

- Valores
- inline | block | contents | flex | grid | inline-block | inline-flex | inline-grid | inline-table | list-item | run-in | table | table-caption | table-column-group | table-header-group | table-footer-group | table-row-group | table-cell | table-column | table-row | none | initial | inherit

```
li{  
  border: 5px solid lightgreen;  
  margin:0.5em;  
  padding:0.5em;  
  width: 100px;  
  display:inline;  
}
```

Uno

Dos

Tres

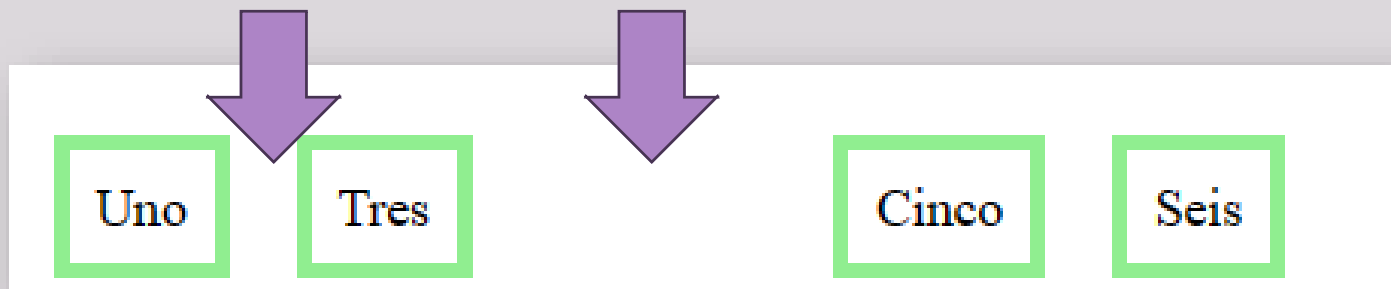
Cuatro

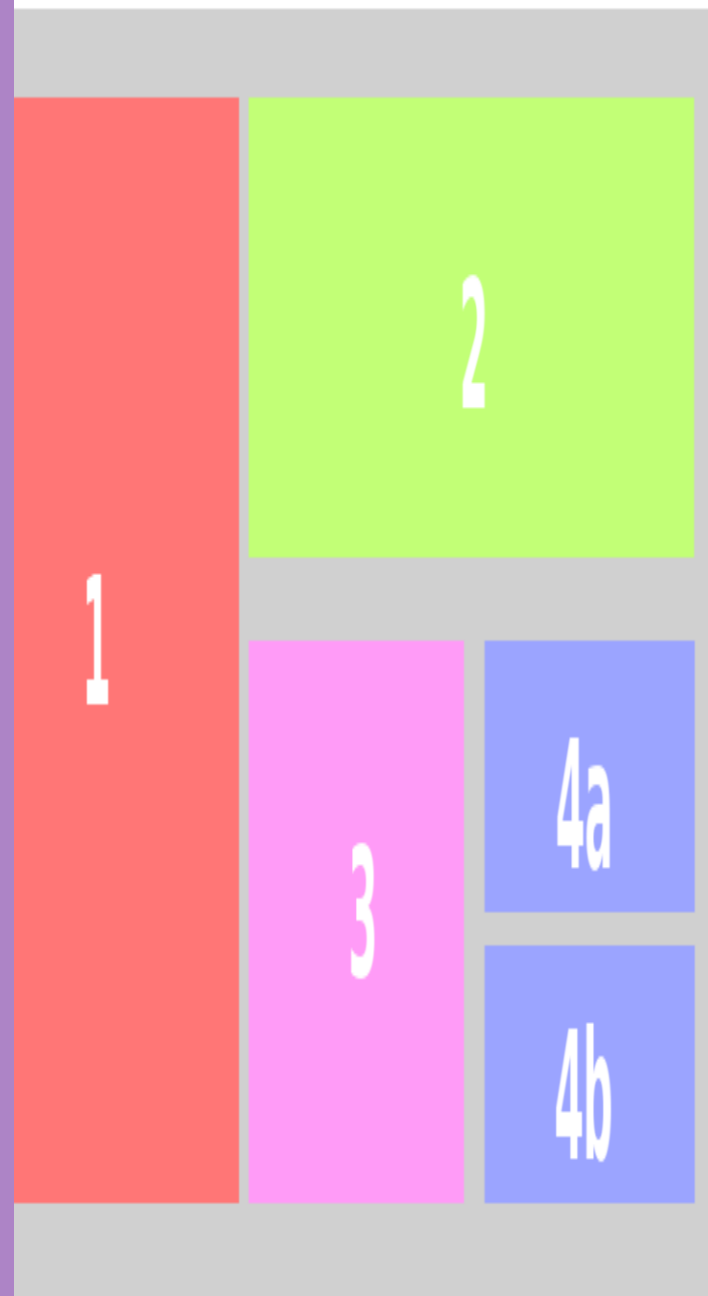
Cinco

Seis

# DISPLAY:NONE Y VISIBILITY:HIDDEN

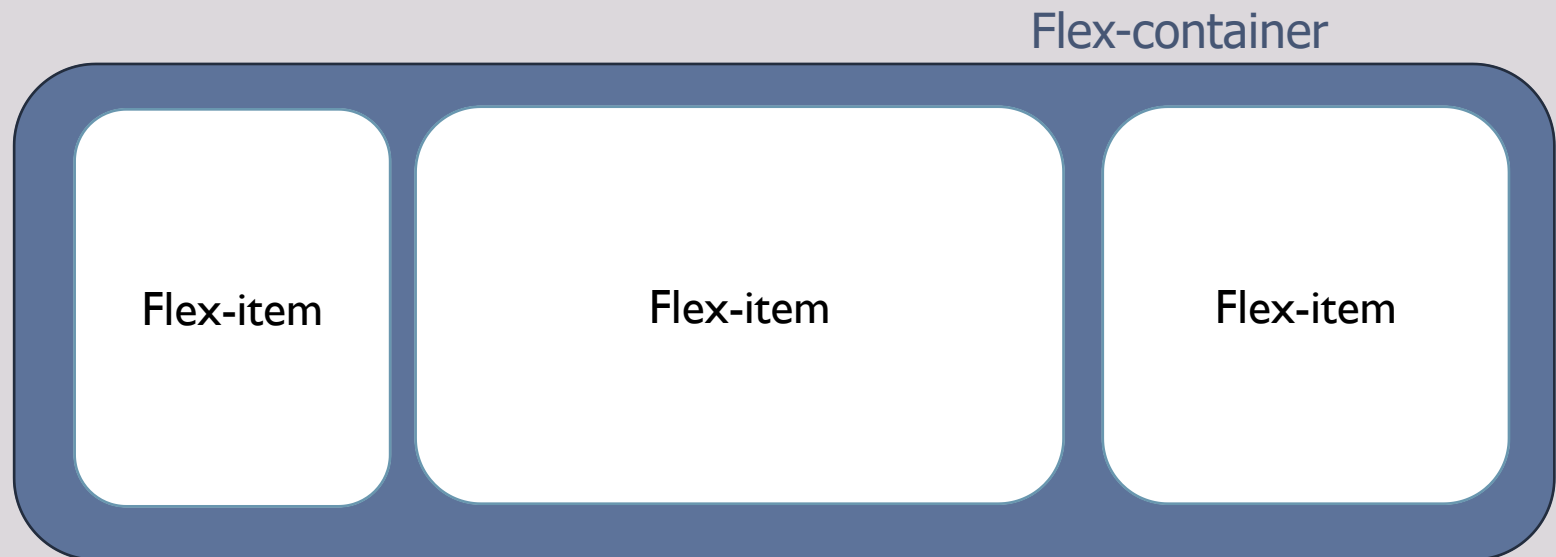
```
li{  
  border: 5px solid lightgreen;  
  margin:0.5em;  
  padding:0.5em;  
  width: 100px;;  
  display:inline  
}  
.segundo{  
  display:none;  
}  
.cuarto{  
  visibility: hidden;  
}
```





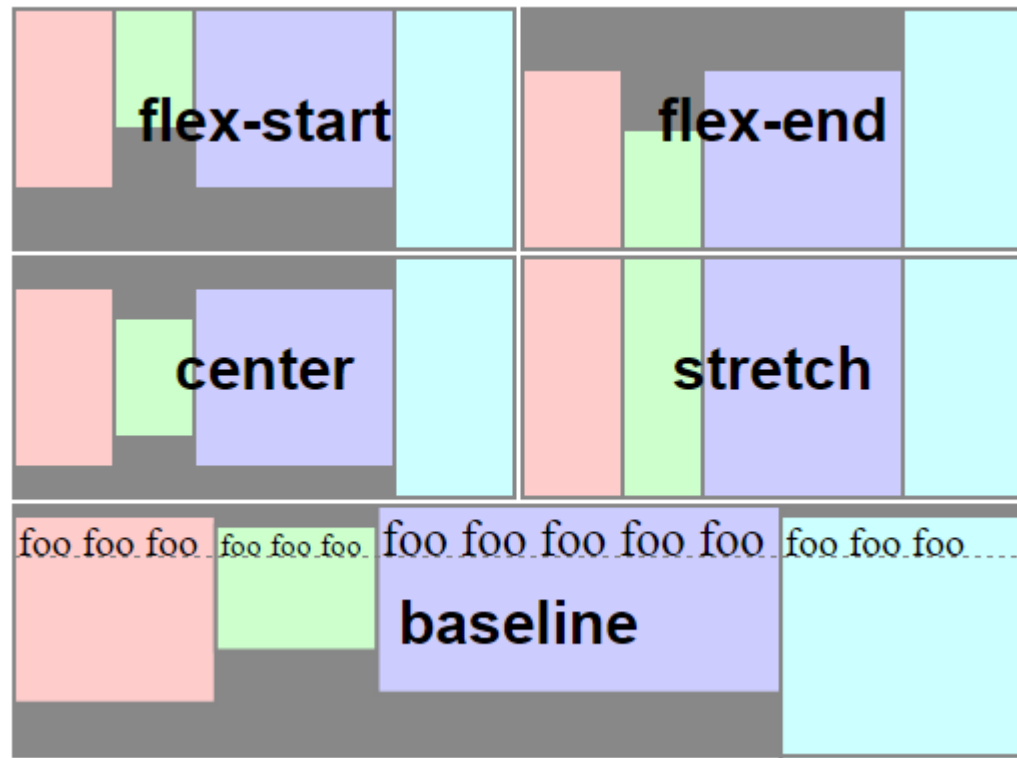
# FLEXBOX

- Flexbox es un sistema de diseño de página unidimensional
- Facilita la alineación y distribución de elementos en filas o columnas.
- Permite que los elementos de contenido se expandan para ocupar el espacio adicional disponible y se contraigan para adaptarse a espacios más reducidos.
- Alternativa a los métodos anteriores

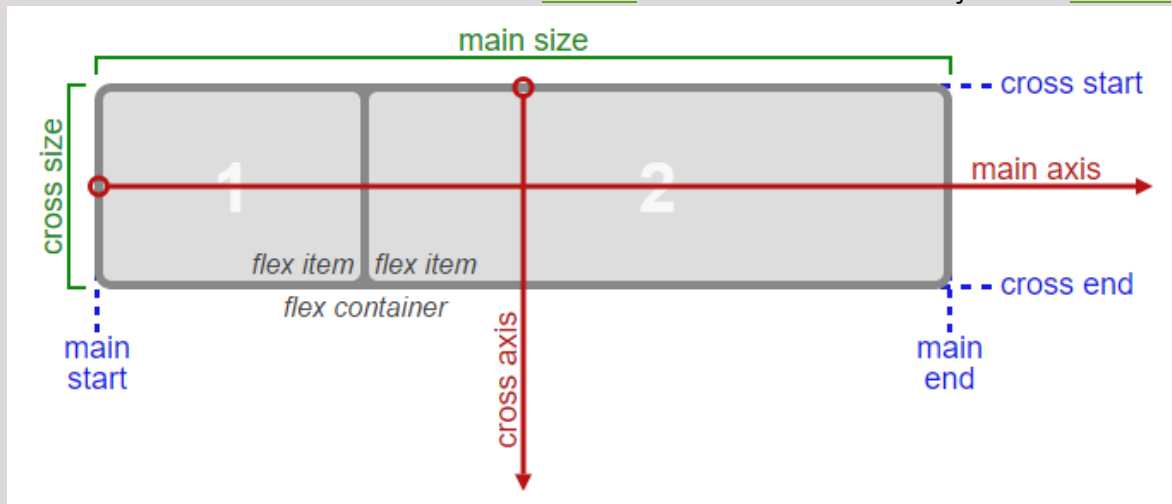


# FLEXBOX

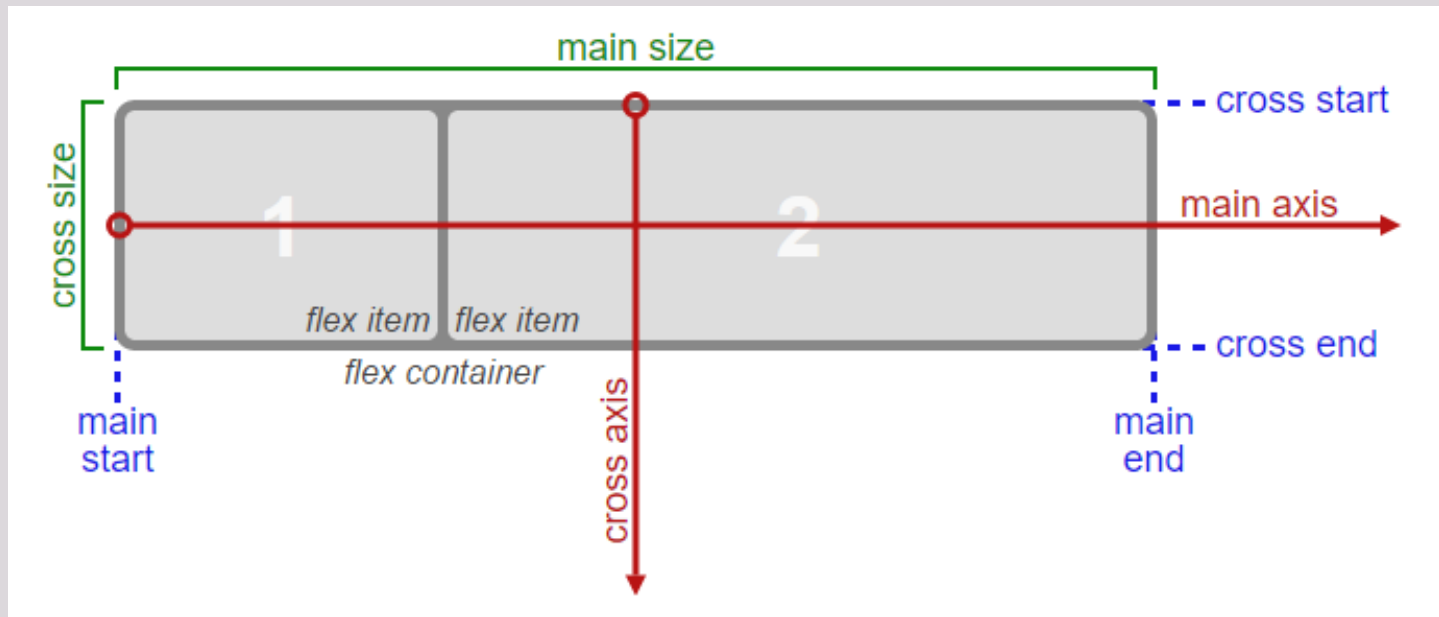
- Flexbox es un sistema de diseño
- Facilita la alineación y distribución
- Permite que los elementos ocupen el espacio adicional disponible en contenedores más reducidos.
- Alternativa a los métodos de



Esta foto de Autor desconocido está bajo licencia [CC BY-SA](#)



# FLEXBOX





# FLEXBOX. DISPLAY:FLEX

- **Display:flex**
- En el contenedor debemos poner la propiedad display con el valor flex

```
.flex-container{  
    display:flex;  
}
```

# FLEXBOX. FLEX-DIRECTION

- **flex-direction**
- Establece el eje principal, definiendo así la dirección en la que los flex ítems están alineados en el flex container. Flexbox es (con la excepción de un wrapping opcional) un concepto de diseño unidireccional. Piensa en los flex ítems inicialmente posicionados o en líneas horizontales o en columnas verticales.



flex-direction: row



flex-direction: row-reverse



flex-direction: column



flex-direction: column-reverse

```
.flex-container{  
  display: flex;  
  flex-direction: row-reverse;  
}
```

# FLEXBOX. FLEX-DIRECTION

- **flex-direction**
- Establece el eje principal, definiendo así la dirección en la que los flex ítems están alineados en el flex container. Flexbox es (con la excepción de un wrapping opcional) un concepto de diseño unidireccional. Piensa en los flex ítems inicialmente posicionados o en líneas horizontales o en columnas verticales.
- Valores: row | row-reverse | column | column-reverse
  - row (estándar): de la izquierda a la derecha en ltr (left to right), de la derecha a la izquierda en rtl (right to left)
  - row-reverse: de la derecha a la izquierda en ltr, de la izquierda a la derecha en rtl
  - column: mismo que row, pero de arriba a abajo
  - column-reverse: mismo que row-reverse pero de abajo hacia arriba



flex-direction: row



flex-direction: row-reverse



flex-direction: column

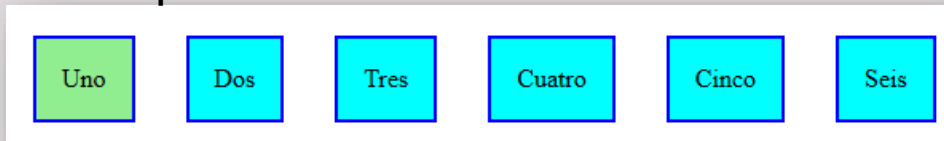


flex-direction: column-reverse

# FLEXBOX. FLEX-WRAP

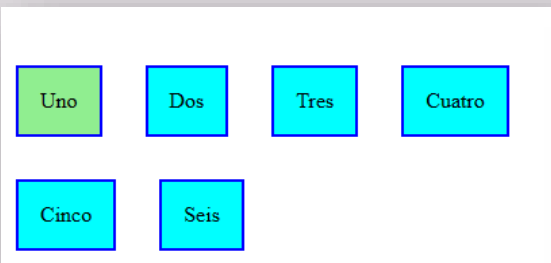
- Por estándar, los flex ítems todos intentarán encajarse en una sola línea. Con esta propiedad puedes modificar este comportamiento y permitir que los ítems pasen a la siguiente línea **según sea necesario (si no caben)**.
- **Valores:** nowrap | wrap | wrap-reverse

nowrap

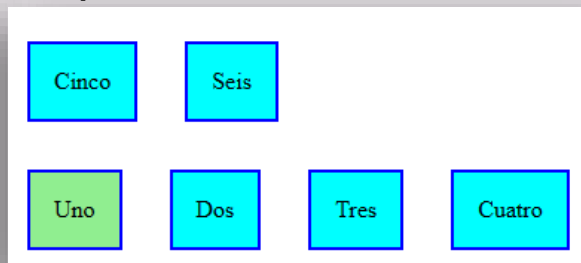


```
.flex-container{  
  display:flex;  
  flex-direction:row;  
  flex-wrap:wrap;  
}
```

wrap



wrap-reverse



# FLEX-BOX. FLEX-FLOW

**Shorthand**

- **flex-flow**
- shorthand que incluye flex-direction y flex-wrap. Determina cuáles serán los ejes principal y transversal del contenedor. El valor estándar es row nowrap.

```
.flex-container {  
  flex-flow: row nowrap | row wrap | column nowrap | column wrap;  
}
```

# FLEXBOX

- **flex-direction**
- Establece el eje principal, definiendo así la dirección en la que los flex ítems están alineados en el flex container. Flexbox es (con la excepción de un wrapping opcional) un concepto de diseño unidireccional. Piensa en los flex ítems inicialmente posicionados o en líneas horizontales o en columnas verticales.
- Valores: row | row-reverse | column | column-reverse
  - row (estándar): de la izquierda a la derecha en ltr (left to right), de la derecha a la izquierda en rtl (right to left)
  - row-reverse: de la derecha a la izquierda en ltr, de la izquierda a la derecha en rtl
  - column: mismo que row, pero de arriba a abajo
  - column-reverse: mismo que row-reverse pero de abajo hacia arriba

### justify-content

flex-start



center



flex-end



space-around



space-between

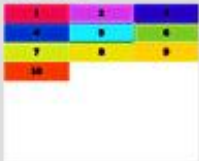


space-evenly

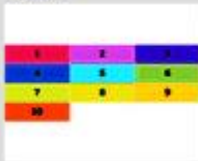


### align-content

flex-start



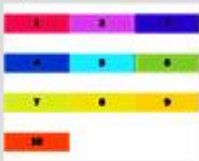
center



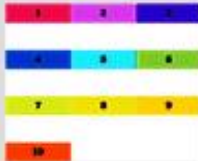
flex-end



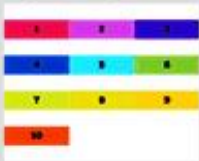
space-around



space-between



space-evenly



stretch



### align-items

stretch



stretch



baseline



flex-around



center



flex-end



### align-self

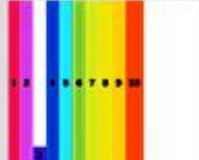
flex-start



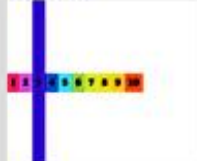
center



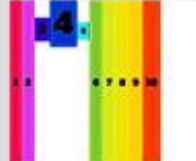
flex-end



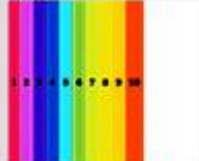
stretch



baseline

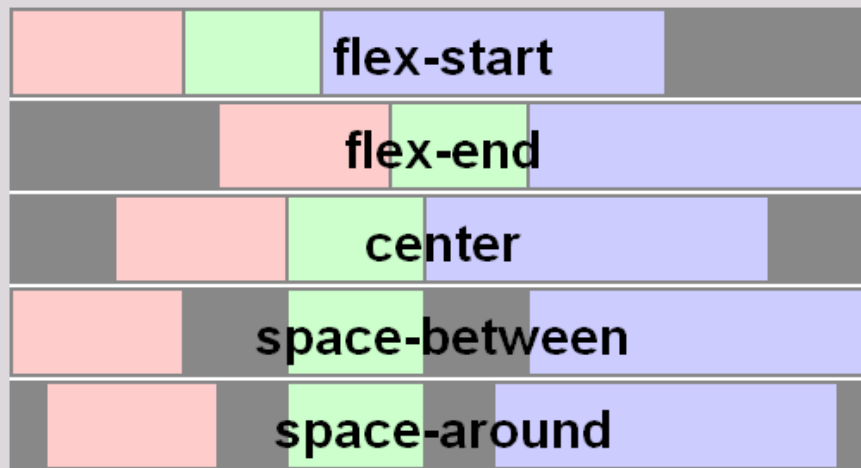


auto



# FLEXBOX. JUSTIFY-CONTENT

- (Falta Space-evenly)

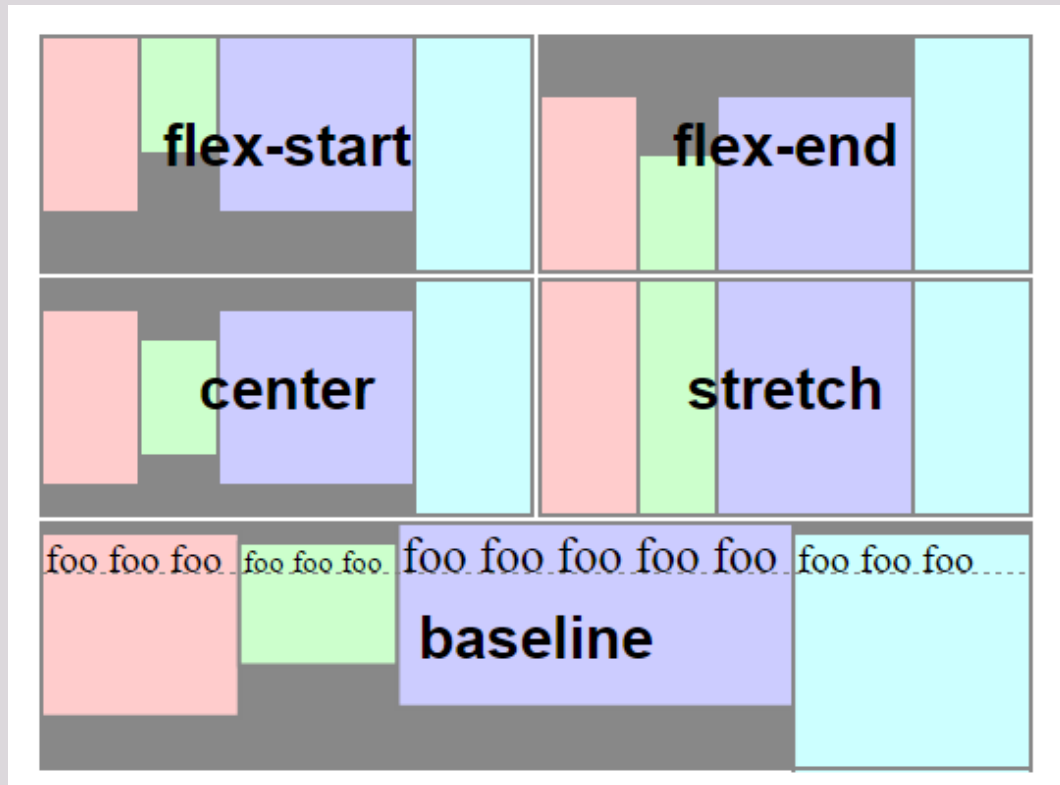


```
.flex-container{  
  display:flex;  
  flex-direction:row;  
  flex-wrap:no-wrap;  
  justify-content:space-between;  
}
```



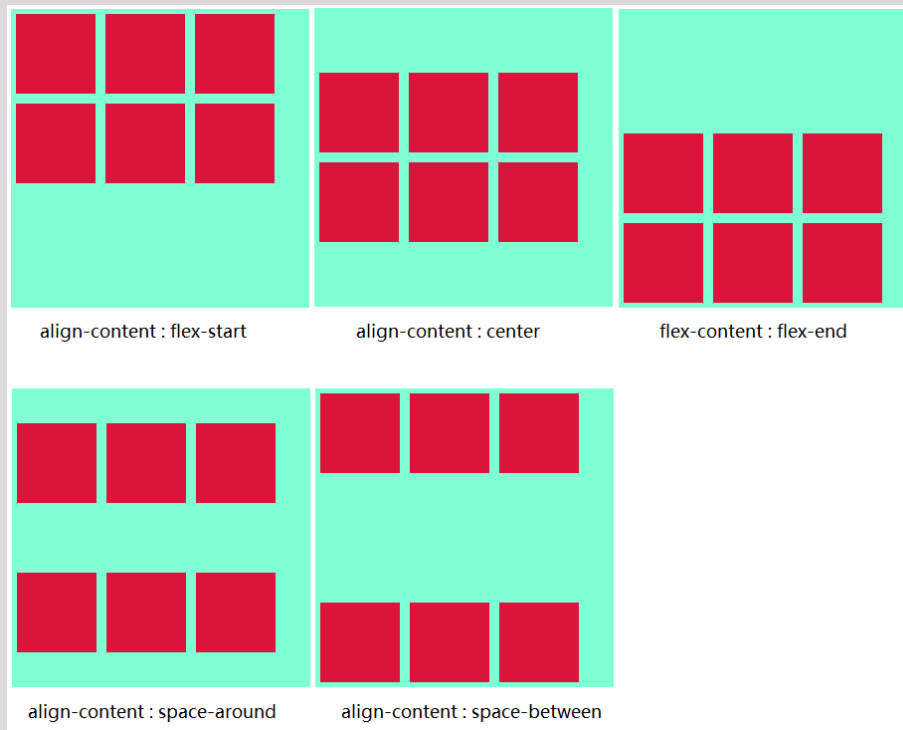
# FLEXBOX. JUSTIFY-CONTENT

- Establece el comportamiento estándar de cómo flex ítems están alineados según el eje transversal



# FLEXBOX. ALIGN-CONTENT

- Organiza las líneas dentro de un contenedor cuando hay espacio adicional en el eje transversal, similar a cómo justify-content alinea ítems individuales dentro del eje principal. Esta propiedad **no tiene efecto cuando solo hay una línea de flex-ítems en el contenedor**.
- Falta space-evenly en la siguiente imagen



# @MEDIA QUERIES

- Las Media Queries son una característica crucial en CSS que permite aplicar estilos específicos según las características del dispositivo o la pantalla. Estas consultas de medios permiten crear diseños web responsivos que se adaptan a diferentes tamaños de pantalla, resoluciones y dispositivos.
- Sintaxis Básica:

```
@media tipo_de_medio y (características) {  
    ...estilos...  
}
```

- Las Media Queries se incluyen en las hojas de estilo CSS mediante la utilización de la regla @media.
- El "tipo\_de\_medio" especifica el tipo de dispositivo, como screen para pantallas, print para impresoras, o all para todos los dispositivos.
- Las "características" son condiciones que determinan cuándo se aplicarán los estilos.

# @MEDIA QUERIES

- Características:

- Ancho de Pantalla: @media screen and (max-width: 600px) { ... } aplicará estilos cuando el ancho de la pantalla sea igual o inferior a 600 píxeles.
- Orientación: @media screen and (orientation: landscape) { ... } aplicará estilos cuando la pantalla esté en modo horizontal.
- Resolución: @media print and (min-resolution: 300dpi) { ... } aplicará estilos cuando la resolución de impresión sea de al menos 300 píxeles por pulgada.
- Dispositivos: @media screen and (device-width: 768px) { ... } aplicará estilos si el ancho del dispositivo es de 768 píxeles.

- Operadores Lógicos:

- Los operadores lógicos and, or, y not se utilizan para combinar múltiples condiciones en una Media Query.

```
@media screen and (max-width: 600px) {  
  section {  
    margin: 0.5em;  
    padding: 0.5em;;  
  }  
  header{  
    background-color: darkorchid;  
  }  
  main{  
    flex-direction: column;  
  }  
}
```

# BOOSTRAP

- Serie de herramientas para maquetar en front-end
- Dota a los sitios web de responsividad

# BOOTSTRAP

• <https://getbootstrap.com/>



## Include via CDN

When you only need to include Bootstrap's compiled CSS or JS, you can use [jsDelivr](#). See it in action with our simple [quick start](#), or [browse the examples](#) to jumpstart your next project. You can also choose to include Popper and our JS [separately](#).

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" rel="script">
```

Figures

### Forms

Overview  
Form control  
Select  
Checks & radios  
Range  
Input group  
Floating labels  
Layout  
Validation

### Components

#### Accordion

Alerts  
Badge  
Breadcrumb  
Buttons  
Button group  
Card  
Carousel  
Close button  
Collapse  
Dropdowns

## How it works

The accordion uses [collapse](#) internally to make it collapsible. To render an accordion that's expanded, add the `.open` class on the `.accordion`.

The animation effect of this component is dependent on the `prefers-reduced-motion` media query. See the [reduced motion](#) section of our [accessibility documentation](#).

## Example #

Click the accordions below to expand/collapse the accordion content.

Accordion Item #1

Accordion Item #2

Accordion Item #3

**This is the third item's accordion body.** It is hidden by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the `.accordion-body`, though the transition does limit overflow.

HTML

