



REPASO CSS



Css – Parte I

Cascade style sheet



CSS

• 1.-¿Qué es CSS?

- Lenguaje de hojas de estilo creado para controlar el aspecto o presentación de los documentos HTML.
- Puedes consultar el estado actual de cada componente de CSS 3 en <https://www.w3.org/Style/CSS/current-work>.
- Mejora el código HTML, elimina antigua etiqueta , separa los contenidos de la página y la información sobre su aspecto, se pueden aplicar los mismos estilos a muchas etiquetas con poco código.

CSS

• 2.-Incluir CSS en un documento HTML

• CSS en el documento HTML.

- Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta <style> de HTML y se deben incluir en la cabecera del documento.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Formularios</title>
8
9      <style>
10         p {
11             color: red;
12         }
13     </style>
14
15 </head>
16 <body>
```


CSS



- **2.-Incluir CSS en un documento HTML**

- CSS en un archivo externo. Es lo recomendado.
 - `<link rel="stylesheet" type="text/css" href="/css/estilos.css" media="screen" />`
 - **rel:** indica el tipo de relación que existe entre el recurso enlazado (en este caso, el archivo CSS) y la página HTML. Para los archivos CSS, siempre se utiliza el valor `stylesheet`
 - **type:** indica el tipo de recurso enlazado. 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.
 - También se puede usar la etiqueta `<style>` para añadir archivos CSS externos.

CSS

• 2.-Incluir CSS en un documento HTML

• Medios en CSS (atributo *media*)

Medio	Descripción
all	Todos los medios definidos
braille	Dispositivos táctiles que emplean el sistema braille
embosed	Impresoras braille
handheld	Dispositivos de mano: móviles, PDA, etc.
print	Impresoras y navegadores en el modo " <i>Vista Previa para Imprimir</i> "
projection	Proyectores y dispositivos para presentaciones
screen	Pantallas de ordenador
speech	Sintetizadores para navegadores de voz utilizados por personas discapacitadas
tty	Dispositivos textuales limitados como teletipos y terminales de texto
tv	Televisores y dispositivos con resolución baja

Ejemplos de uso:

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

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

```
@media print { estilos para impresoras y Vista Previa de navegadores }
```

```
@media screen { estilos para pantallas }
```

```
@media screen, print { estilos para ambos }
```

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

```
@import url("estilos_impresora.css") print;
```

Todos los métodos se pueden mezclar.

CSS

<> formularios.html > html > head > style

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Formularios</title>
8      <link rel="stylesheet" type="text/css" href="estilos/flex.css" media="screen" >
9      <style>
10         @import url('estilos/grid.css');
11     </style>
12 </head>
13 <body>
14     <form method="get" action="http://localhost/app/gestionDatos.php" id="formulario1">
15         <datalist id="datos">
16             <option value="123123123" label="Teléfono 1">
17             <option value="456456456" label="Teléfono 2">
18         </datalist>
19         <p><label>Teléfono: <input type="tel" name="telefono" list="datos"></label></p>
20     </form>
21     <input type="submit" name="envio" value="Enviar datos">
22
23 </body>
24 </html>
```



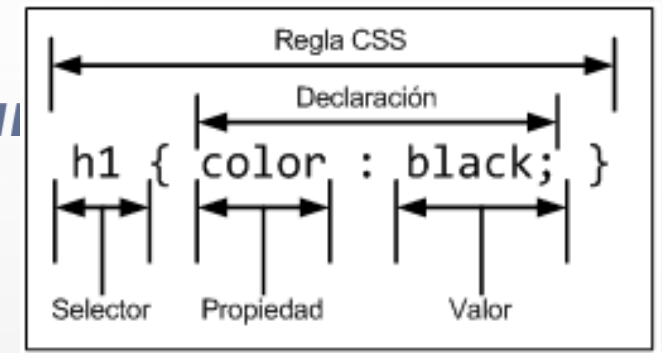
- **2.-Incluir CSS en un documento HTML**

- CSS en elementos HTML.

- Esta forma de incluir CSS directamente en los elementos HTML solamente se utiliza en determinadas situaciones en las que se debe incluir un estilo muy específico para un solo elemento concreto.

- `<p style="color:red">Este párrafo tiene fuente de color rojo</p>`

CSS



• 3.-Reglas CSS

- Los diferentes términos se definen a continuación:
 - **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:** indica el elemento o elementos HTML a los que se aplica la regla CSS.
 - **Declaración:** especifica los estilos que se aplican a los elementos. Está compuesta por una o más propiedades CSS.
 - **Propiedad:** característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, su color de fondo, etc.
 - **Valor:** establece el nuevo valor de la característica modificada en el elemento.

CSS

• 3.-Reglas CSS. Selectores.

- La declaración de una regla indica "qué hay que hacer" y el selector indica "a quién hay que hacérselo".
- Una misma regla puede aplicarse sobre varios selectores y un mismo selector se puede utilizar en varias reglas.

```
1 * {  
2   margin: 0;  
3   padding: 0;  
4 }
```

Selector universal.
Aplica las reglas a
todos los elementos
de la página.

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

Selector de etiqueta.
Aplica las reglas a todos
los elementos cuya
etiqueta coincide con el
selector

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

Si queremos asignar mismos estilos a
etiquetas distintas, lo separamos por
comas.



• 3.-Reglas CSS. Selectores.

```
p span {  
  color: ■ #AB0101;  
}
```

Selector descendiente.

Selecciona los elementos que se encuentran dentro de otros elementos.

- La sintaxis formal sería:
 - selector1 selector2 selector3 ... selectorN {..} //Siempre se aplica al último elemento
- No confundir:
 - El selector: p a span em { ... }
 - El selector: p, a, span, em { ... }

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

• 3.-Reglas CSS. Selectores. CSS

• Selector de clase:

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

```
<p class="destacado">Este párrafo es muy importante</p>
```

- Una clase se puede usar **n** veces en todo el documento. Aplica las reglas establecidas a todos los elementos que tengan el atributo *class* con valor igual que el nombre de clase definido.
- Un elemento puede tener varias clases definidas en su atributo *class*:
`<p class="destacado aviso importante">...</p>`
- Diferenciar entre los siguientes selectores:
 - `p.destacado {...}` -- `p .destacado { ... }` -- `p, .destacado { ... }`
- ¿`¿.error.destacado { color: blue; }?`

CSS

• 3.-Reglas CSS. Selectores.

• Selector de ID:

```
#cabecera {  
  background-color: aqua;  
}
```

```
<header id="cabecera">  
  <h1>Listado de cursos</h1>  
</header>
```

- Un ID se debe usar **una** vez en todo el documento. Aplica las reglas establecidas al elemento que tenga el atributo *id* con valor igual que el nombre de identificador definido.
- Un elemento puede tener varios ids definidos en su atributo *id*: `<header id="cabecera arriba">...</p>`
- Diferenciar entre los siguientes selectores:
 - `p#destacado {...}` -- `p #destacado { ... }` -- `p, #destacado { ... }`
- ¿`ul#menuPrincipal li.destacado a#inicio { ... }`?



- **3.-Reglas CSS. Selectores avanzados.**

- **Selector hijo directo:**

- Se utiliza para seleccionar un elemento que es hijo directo de otro elemento y se indica mediante el "signo de mayor que" (>):

- `p > a { color:green; }`

...

`<p>Párrafo de prueba</p>`

`<p>Párrafo de prueba</p>`

- ¿A qué elemento se aplicaría la regla?

CSS

• 3.-Reglas CSS. Selectores avanzados.

• Selector adyacente:

- El selector adyacente selecciona elementos que en el código HTML se encuentran justo a continuación de otros elementos. Su sintaxis emplea el signo **+** para separar los dos elementos:

```
p + p {  
    text-indent: 1.5em;  
}
```

- elemento1 + elemento2 { ... }
- Deben cumplir estas dos condiciones:
 - elemento1 y elemento2 deben ser elementos hermanos, por lo que su elemento padre debe ser el mismo.
 - elemento2 debe aparecer inmediatamente después de elemento1 en el código HTML de la página.

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in condimentum augue </p>  
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in condimentum augue </p>  
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in condimentum augue, </p>
```



• **3.-Reglas CSS. Colisiones**

- El método seguido es:
 - Determinar todas las declaraciones que se aplican al elemento para el medio CSS seleccionado.
 - Ordenar las declaraciones según su origen (CSS de navegador, de usuario o de diseñador) y su prioridad (palabra clave !important).
 - Ordenar las declaraciones según lo específico que sea el selector. Cuanto más genérico es un selector, menos importancia tienen sus declaraciones.
 - Si después de aplicar las normas anteriores existen dos o más reglas con la misma prioridad, se aplica la que se indicó en último lugar.
- Simplificando:
 - Cuanto más específico sea un selector, más importancia tiene su regla asociada.
 - A igual especificidad, se considera la última regla indicada.



• 3.-Reglas CSS. Unidades de medida.

• Unidades de medida absoluta:

- **in**, pulgadas ("inches", en inglés). Una pulgada equivale a 2.54 centímetros.
- **cm**, centímetros.
- **mm**, milímetros.
- **pt**, puntos. Un punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros.
- **pc**, picas. Una pica equivale a 12 puntos, es decir, unos 4.23 milímetros.

• Unidades de medida relativas

- **em**, (no confundir con la etiqueta de HTML) relativa respecto del tamaño de letra del elemento.
- **ex**, relativa respecto de la altura de la letra x ("equis minúscula") del tipo y tamaño de letra del elemento.
- **px**, (píxel) relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.
- **%**, se refiere al porcentaje sobre el elemento padre.



- **3.-Reglas CSS. Unidades de medida.**
- Además en CSS3:
 - **rem**: parecido a em, pero toma como base el tamaño de fuente del elemento raíz (html).
 - **vw**: representa el 1% del ancho de la pantalla del dispositivo.
 - **vh**: representa el 1% de la altura de la pantalla del dispositivo.
 - **vmin**: 1% de la dimensión más pequeña de la ventana gráfica (entre vw y vh)
 - **vmax**: 1% de la dimensión más grande de la ventana gráfica.
 - **ch**: representa la anchura del carácter "0" (zero) del tipo de letra del elemento.
 - **lh**: altura de la línea del elemento. (line-height, no se debe usar para tamaños de fuente)

CSS



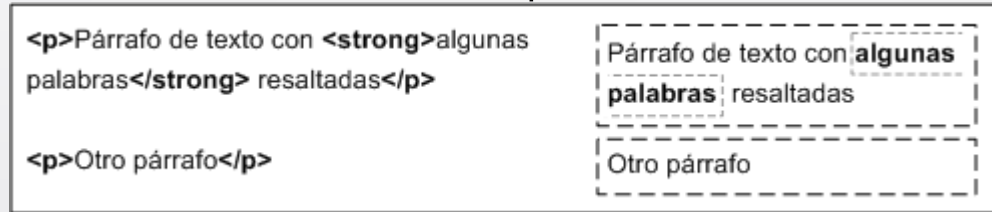
- **3.-Reglas CSS. Colores.**

- **Palabras clave:** aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow...
- **RGB decimal y porcentual.** definir un color indicando la cantidad de color rojo, verde y azul que se debe mezclar para obtener ese color.
 - `p { color: rgb (71, 98, 176); } /* Red=71, Green=98 y Blue=176*/`
- **RGB hexadecimal:** normalmente se usan 6 dígitos hexadecimales que representan RGB, precedido por el símbolo #.
 - `p { color: #4762B0; } /* Red= 47, Green=62 y Blue=B0 */`
- **HWB (hue, whiteness, blackness):** Recibe tres valores,
 - El primero representa en grados la posición del tono en la rueda de color.
 - El segundo expresado en porcentaje, definiendo la cantidad de blanco.
 - El tercero expresado en porcentaje, definiendo la cantidad de negro.
 - Por ejemplo `color: hwb(60, 0%, 0%);`

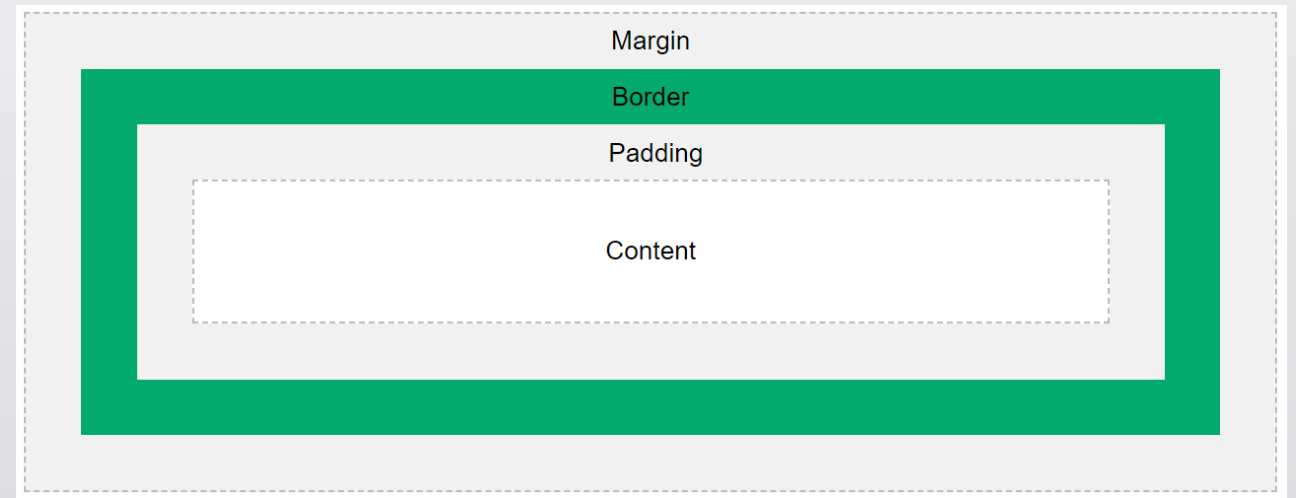
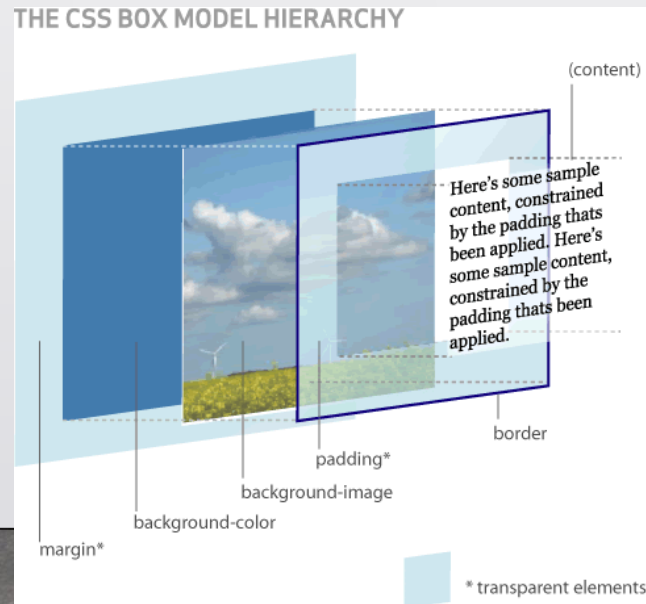
CSS

• 4.-Box Model.

- Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular



- Cada caja tiene una serie de propiedades:



CSS



- **4.-Box Model.**

- **Contenido (content):** se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
- **Relleno (padding):** espacio libre opcional existente entre el contenido y el borde.
- **Borde (border):** línea que encierra completamente el contenido y su relleno.
- **Imagen de fondo (background image):** imagen que se muestra por detrás del contenido y el espacio de relleno.
- **Color de fondo (background color):** color que se muestra por detrás del contenido y el espacio de relleno.
- **Margen (margin):** separación opcional existente entre la caja y el resto de cajas adyacentes.



- **4.-Box Model.**

- Si una caja define tanto un color como una imagen de fondo, **la imagen tiene más prioridad** y es la que se visualiza. No obstante, si la imagen de fondo no cubre totalmente la caja del elemento o si la imagen tiene zonas transparentes, también se visualiza el color de fondo.
- El **relleno y el margen son transparentes**, por lo que en el espacio ocupado por el relleno se muestra el color o imagen de fondo (si están definidos) y en el espacio ocupado por el margen se muestra el color o imagen de fondo de su elemento padre (si están definidos)

CSS

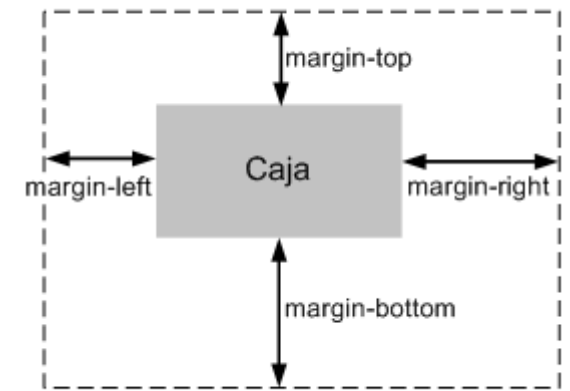
• 4.-Box Model. Propiedades.

Propiedad	width	Propiedad	height
Valores	unidad de medida porcentaje auto inherit	Valores	unidad de 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	Se aplica a	Todos los elementos, salvo los elementos en línea que no sean imágenes, las columnas de tabla y los grupos de columnas de tabla
Valor inicial	auto	Valor inicial	auto
Descripción	Establece la anchura de un elemento	Descripción	Establece la altura de un elemento

- Estas reglas no admiten valores negativos.
- *Inherit*: toma el valor del elemento padre.
- Si se establece el valor en %, se calcula a partir del elemento padre.

• 4.-Box Model. Propiedades: márgenes.

Propiedades	margin-top, margin-right, margin-bottom, margin-left
Valores	unidad de 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
Valor inicial	0
Descripción	Establece cada uno de los márgenes horizontales y verticales de un elemento

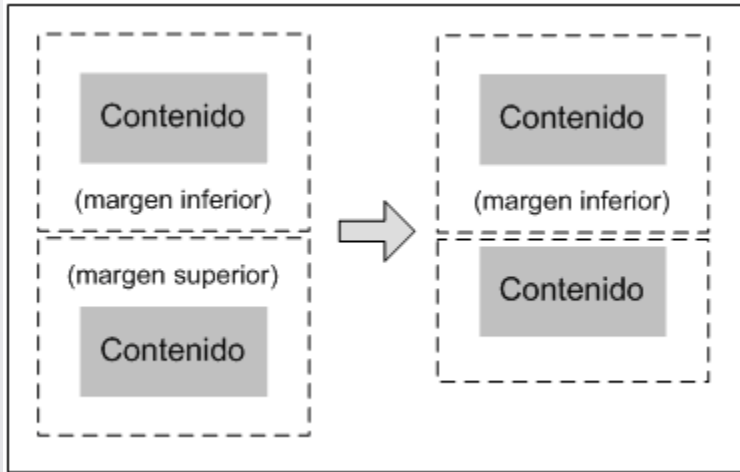


Propiedad	margin
Valores	{ unidad de medida porcentaje auto } {1, 4} inherit
Se aplica a	Todos los elementos salvo algunos casos especiales de elementos mostrados como tablas
Valor inicial	-
Descripción	Establece de forma directa todos los márgenes de un elemento

- Si solo se indica **un** valor, todos los márgenes tienen ese valor.
- Si se indican **dos** valores, el primero se asigna al margen superior e inferior y el segundo se asigna a los márgenes izquierdo y derecho.
- Si se indican **tres** valores, el primero se asigna al margen superior, el tercero se asigna al margen inferior y el segundo valor se asigna los márgenes izquierdo y derecho.
- Si se indican los **cuatro** valores, el orden de asignación es: margen superior, margen derecho, margen inferior y margen izquierdo.

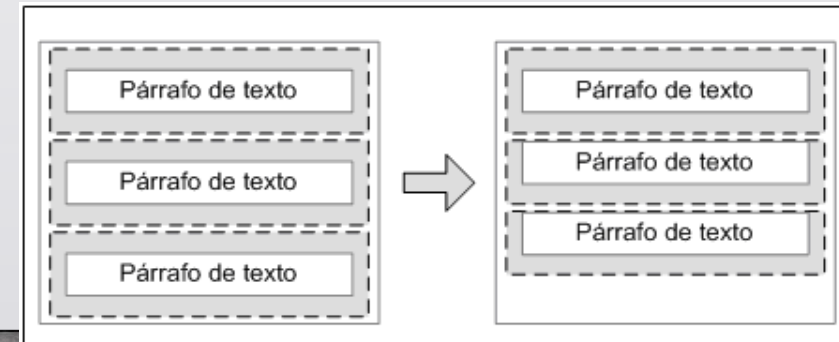
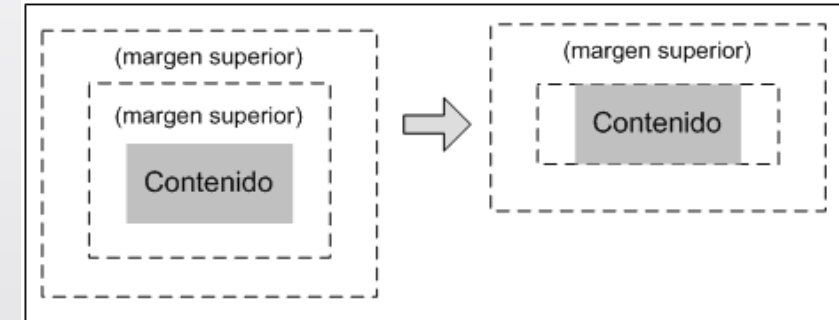
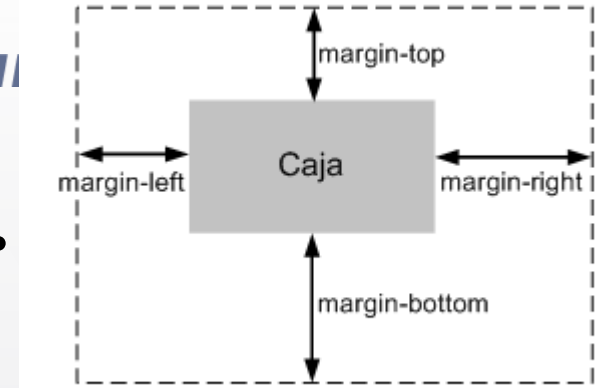
• 4.-Box Model. Propiedades: márgenes.

• Fusión automática de los márgenes

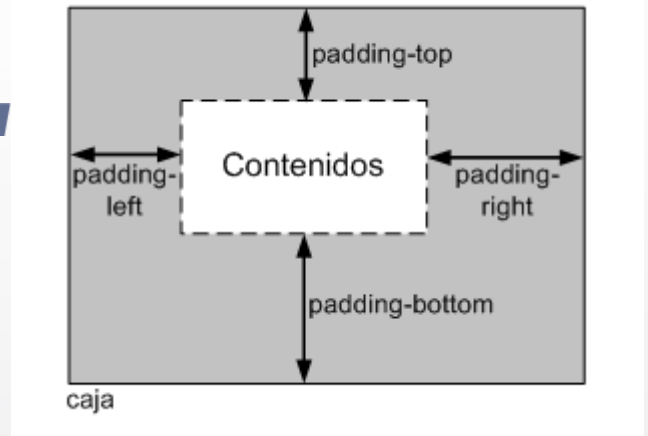


dos o más márgenes verticales, se fusionan de forma automática y la altura del nuevo margen será igual a la altura del margen más alto

si un elemento está contenido dentro de otro elemento, sus márgenes verticales se fusionan y resultan en un nuevo margen de la misma altura que el mayor margen de los que se han fusionado



• 4.-Box Model. Propiedades: Relleno.



Propiedades	padding-top, padding-right, padding-bottom, padding-left
Valores	unidad de medida porcentaje inherit
Se aplica a	Todos los elementos excepto algunos elementos de tablas como grupos de cabeceras y grupos de pies de tabla
Valor inicial	0
Descripción	Establece cada uno de los rellenos horizontales y verticales de un elemento

Propiedad	padding
Valores	(unidad de medida porcentaje) {1, 4} inherit
Se aplica a	Todos los elementos excepto algunos elementos de tablas como grupos de cabeceras y grupos de pies de tabla
Valor inicial	-
Descripción	Establece de forma directa todos los rellenos de los elementos

CSS



- **4.-Box Model. Propiedades: Bordes.**

- Podemos modificar:
 - Anchura: *border-width*
 - Color: *border-color*
 - Tipo de borde: *border-style*.
- De cada una de esas propiedades podemos modificar por separado el borde superior, izquierda, derecha e inferior:
 - *boder-top-width: 2px; border-left-color: red; border-bottom-style: dotted;*
- Normalmente establecemos todos los bordes usando una única regla abreviada:
 - *border: tamaño tipo color;*
 - *border: 1px solid black;*

CSS



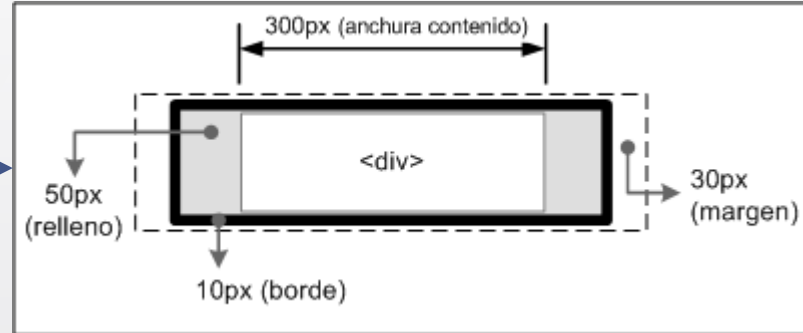
• 4.-Box Model. Propiedades: Bordes.

Propiedades	border-top, border-right, border-bottom, border-left
Valores	(unidad de 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

Propiedad	border
Valores	(unidad de medida_borde color_borde estilo_borde) inherit
Se aplica a	Todos los elementos
Valor inicial	-
Descripción	Establece el estilo completo de todos los bordes de los elementos

• 4.-Box Model. Tamaños de caja.

```
div {
  width: 300px;
  padding-left: 50px;
  padding-right: 50px;
  margin-left: 30px;
  margin-right: 30px;
  border: 10px solid black;
}
```



Ancho total: $30\text{px} + 10\text{px} + 50\text{px} + 300\text{px} + 50\text{px} + 10\text{px} + 30\text{px} = 480 \text{ píxel}$

Podemos modificar el tratamiento que se hace con el cálculo de la anchura.

Box-sizing → indica como se debe calcular el ancho de la caja.

Propiedades	box-sizing
Valores	content-box border-box
Se aplica a	Todos los elementos
Valor inicial	content-box
Descripción	Establece cómo se calcula la anchura de la caja.

content-box: cálculo por defecto, al valor width se le suma el relleno y el margen

border-box: incluye margen y relleno en el cálculo.
 En el ejemplo anterior, la caja ocupa 300px en total:
 $\text{Ancho total} = 300\text{px} = 30\text{px} + 10\text{px} + 50\text{px} + \text{anchoCaja} + 50\text{px} + 10\text{px} + 30\text{px}$
 $\text{AnchoCaja} = 300 - 60 - 20 - 100 = 120\text{px}$

CSS



• 4.-Box Model. Fondos.

Propiedad	background-color
Valores	color transparent inherit
Se aplica a	Todos los elementos
Valor inicial	transparent
Descripción	Establece un color de fondo para los elementos

Propiedad	background-repeat
Valores	repeat repeat-x repeat-y no-repeat inherit
Se aplica a	Todos los elementos
Valor inicial	repeat
Descripción	Controla la forma en la que se repiten las imágenes de fondo

Propiedad	background-image
Valores	url none inherit
Se aplica a	Todos los elementos
Valor inicial	none
Descripción	Establece una imagen como fondo para los elementos

Propiedad	background-position
Valores	((porcentaje unidad de medida left center right) (porcentaje unidad de medida top center bottom)?) ((left center right) (top center bottom)) inherit
Se aplica a	Todos los elementos
Valor inicial	0% 0%
Descripción	Controla la posición en la que se muestra la imagen en el fondo del elemento

• 4.-Box Model. Fondos.

Propiedad	background-attachment
Valores	scroll fixed inherit
Se aplica a	Todos los elementos
Valor inicial	scroll
Descripción	Controla la forma en la que se visualiza la imagen de fondo: permanece fija cuando se hace scroll en la ventana del navegador o se desplaza junto con la ventana

Propiedad	background
Valores	(background-color background-image background-repeat background-attachment background-position) inherit
Se aplica a	Todos los elementos
Valor inicial	-
Descripción	Establece todas las propiedades del fondo de un elemento

CSS



- **4.-Box Model. Posicionamiento y visualización.**
 - Todos los elementos son representados como cajas en el navegador. Para ello se tiene en cuenta:
 - Las propiedades *width* y *height* de la caja (si están establecidas).
 - El tipo de cada elemento HTML (elemento de **bloque** o elemento en **línea**).
 - Posicionamiento de la caja (**normal**, **relativo**, **absoluto**, **fijo** o **flotante**).
 - Las relaciones entre elementos (dónde se encuentra cada elemento, elementos descendientes, etc.)
 - Otro tipo de información, como por ejemplo el tamaño de las imágenes y el tamaño de la ventana del navegador.

• 4.-Box Model. Posicionamiento y visualización.

Propiedad	position
Valores	static relative absolute fixed inherit
Se aplica a	Todos los elementos
Valor inicial	static
Descripción	Selecciona el posicionamiento con el que se mostrará el elemento

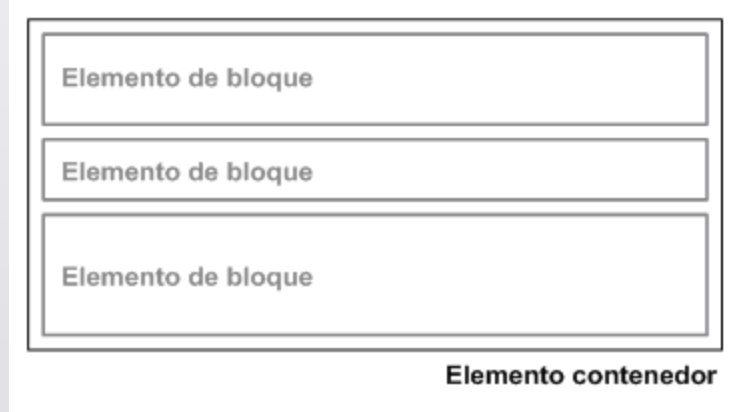
Dependiendo del tipo de posicionamiento podemos aplicar:

Propiedades	top, right, bottom, left
Valores	unidad de medida porcentaje auto inherit
Se aplica a	Todos los elementos posicionados
Valor inicial	auto
Descripción	Indican el desplazamiento horizontal y vertical del elemento respecto de su posición original

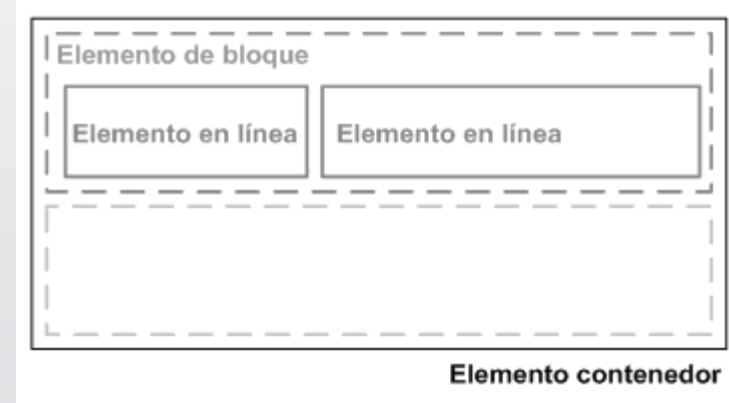
• 4.-Box Model. Posicionamiento y visualización.

• Posicionamiento normal (static).

- es el modelo que utilizan por defecto los navegadores para mostrar los elementos de las páginas. En este modelo, sólo se tiene en cuenta si el elemento es de bloque o en línea, sus propiedades width y height y su contenido.



La distancia entre elementos de bloque con márgenes verticales

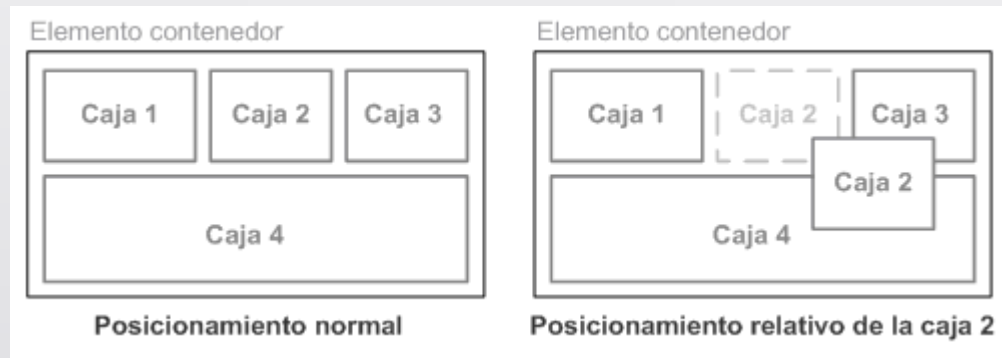


La distancia entre elementos de línea con márgenes horizontales

• 4.-Box Model. Posicionamiento y visualización.

• Posicionamiento relativo (relative).

- desplaza una caja respecto de su posición original establecida mediante el posicionamiento normal. El desplazamiento de la caja se controla con las propiedades *top*, *right*, *bottom* y *left*.



Al mover una caja el resto de cajas respetan el espacio libre

Si tanto **left** como **right** tienen valores distintos de *auto*, uno de los dos valores se tiene que ignorar porque son mutuamente excluyentes.

CSS



• 4.-Box Model. Posicionamiento y visualización.

• Posicionamiento absoluto (absolute).

- La nueva posición de la caja se indica mediante las propiedades top, right, bottom y left y dependen del elemento contenedor.
- El resto de elementos ven afectado su posicionamiento y modifican su posición.



Cálculo origen del desplazamiento:

- Se buscan todos los elementos contenedores de la caja hasta llegar al elemento <body> de la página.
- Se recorren todos los elementos contenedores empezando por el más cercano a la caja y llegando hasta el <body>
- El primer elemento contenedor que esté posicionado de cualquier forma diferente a position: **static** se convierte en la referencia que determina la posición de la caja posicionada de forma absoluta.
- Si ningún elemento contenedor está posicionado, la referencia es la ventana del navegador, que no debe confundirse con el elemento <body> de la página.



- **4.-Box Model. Posicionamiento y visualización.**

- **Posicionamiento fijo (fixed).**

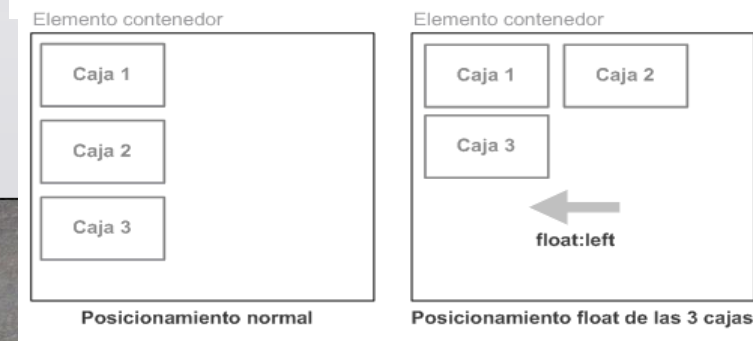
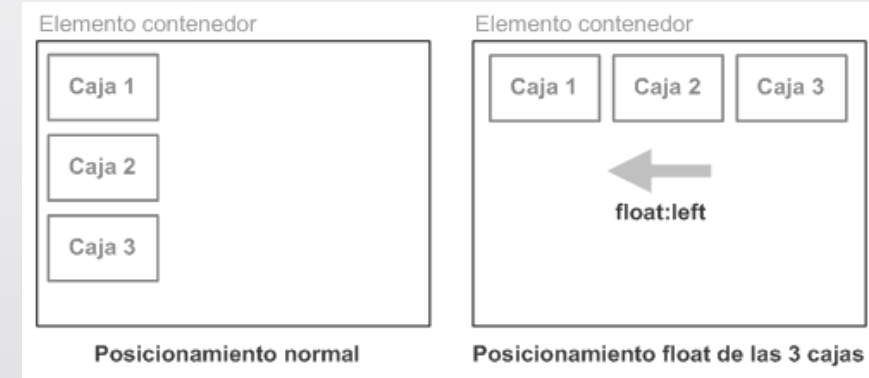
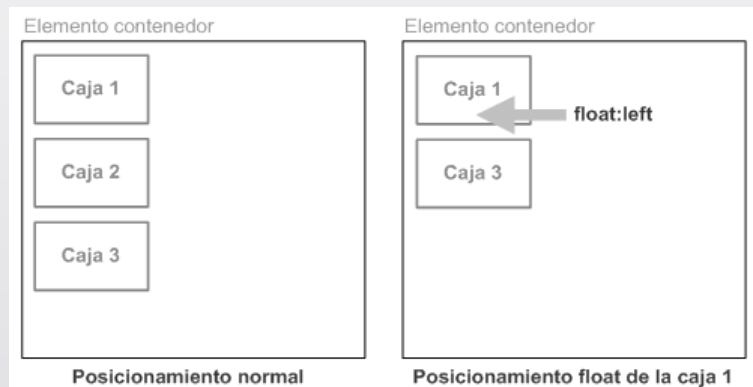
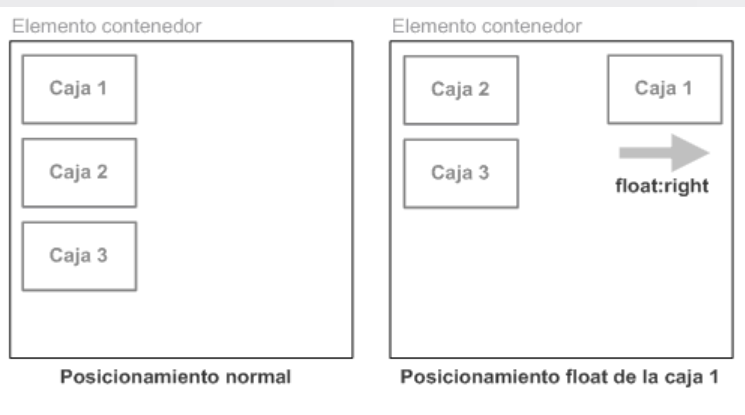
- Posicionamiento particular de posicionamiento absoluto. La forma de obtener origen de coordenadas para el desplazamiento es igual que *absolute*.
- La principal característica de una caja posicionada de forma fija es que su posición es inamovible dentro de la ventana del navegador. El posicionamiento fijo hace que las cajas no modifiquen su posición ni aunque el usuario suba o baje la página en la ventana de su navegador.

CSS

• 4.-Box Model. Posicionamiento y visualización.

• Posicionamiento flotante (float:left, float:right).

- Cuando establecemos esta propiedad a un elemento, éste 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.

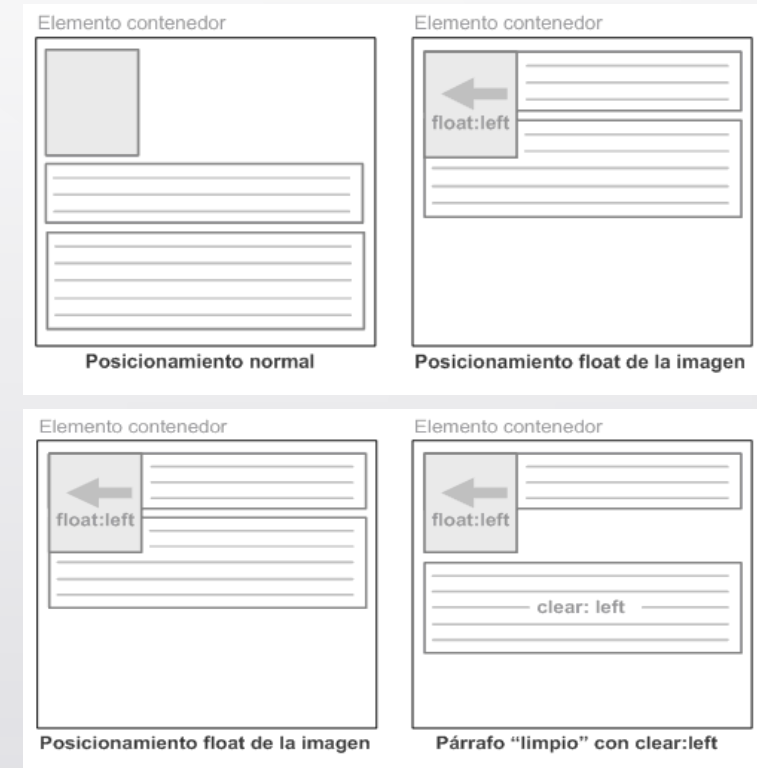


Si no cabe cambia de línea

• 4.-Box Model. Posicionamiento y visualización.

• Posicionamiento float. Clear

Propiedad	clear
Valores	none left right both inherit
Se aplica a	Todos los elementos de bloque
Valor inicial	none
Descripción	Indica el lado del elemento que no debe ser adyacente a ninguna caja flotante



Si se indica la regla `clear:left`, el elemento se desplaza de forma descendente hasta que pueda colocarse en una línea en la que no haya ninguna caja flotante en el lado izquierdo. Igual para `clear:right`.

CSS



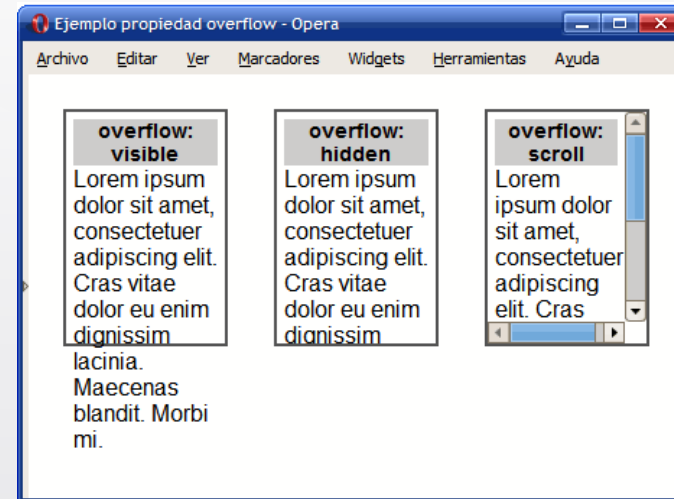
- **4.-Box Model. Posicionamiento y visualización.**
 - **Propiedades display y visibility.**
 - La propiedad *display* permite ocultar completamente un elemento haciendo que desaparezca de la página. Como el elemento oculto no se muestra, el resto de elementos de la página se mueven para ocupar su lugar. Además modifica el comportamiento de la caja si queremos usando valores como: inline, block, flex, grid...
 - la propiedad *visibility* permite hacer invisible un elemento, lo que significa que 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.

CSS

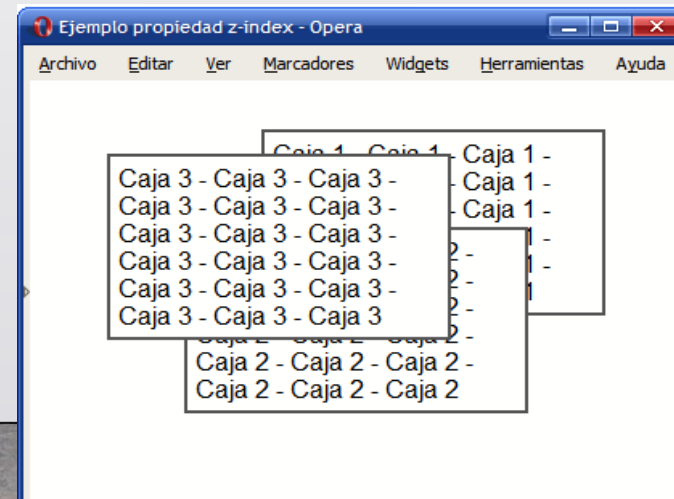
• 4.-Box Model. Posicionamiento y visualización.

• Overflow y z-index.

Propiedad	overflow
Valores	visible hidden scroll auto inherit
Se aplica a	Elementos de bloque y celdas de tablas
Valor inicial	visible
Descripción	Permite controlar los contenidos sobrantes de un elemento



Propiedad	z-index
Valores	auto numero inherit
Se aplica a	Elementos que han sido posicionados explícitamente
Valor inicial	auto
Descripción	Establece el nivel tridimensional en el que se muestra el elemento



• 5.-Texto.

• Tipografía.

Propiedad	color
Valores	color inherit
Se aplica a	Todos los elementos
Valor inicial	Depende del navegador
Descripción	Establece el color de letra utilizado para el texto

Propiedad	font-family
Valores	((nombre_familia familia_generica) (,nombre_familia familia_generica)*) inherit
Se aplica a	Todos los elementos
Valor inicial	Depende del navegador
Descripción	Establece el tipo de letra utilizado para el texto

Para evitar el problema común de que el usuario no tenga instalada la fuente que quiere utilizar el diseñador, CSS permite indicar en la propiedad *font-family* más de un tipo de letra. El navegador probará en primer lugar con el primer tipo de letra indicado. Si el usuario la tiene instalada, el texto se muestra con ese tipo de letra. Probará con el resto si el usuario no tiene instalada, hasta que finalmente, si no encuentra ninguna, cogerá un tipo de letra de la familia genérica lo más parecido:

```
p {
    font-family: "Times New Roman", Times, serif;}
h1 {
    font-family: Arial, Helvetica, sans-serif;}
```

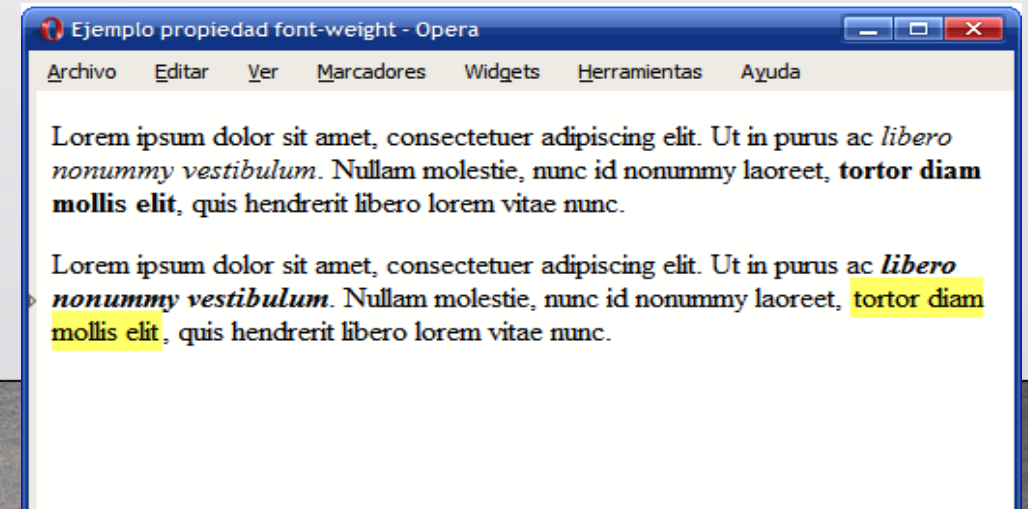
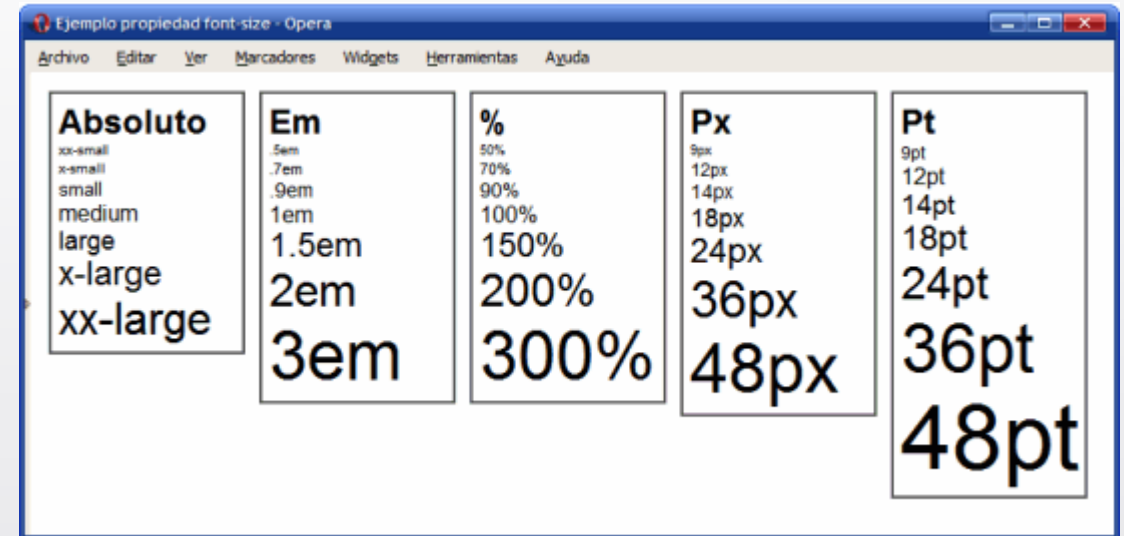
CSS

• 5.-Texto.

• Tipografía.

Propiedad	font-size
Valores	tamaño_absoluto tamaño_relativo unidad de medida porcentaje inherit
Se aplica a	Todos los elementos
Valor inicial	medium
Descripción	Establece el tamaño de letra utilizado para el texto

Propiedad	font-weight
Valores	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Establece la anchura de la letra utilizada para el texto



CSS

• 5.-Texto.

• Tipografía.

Propiedad	font-style
Valores	normal italic oblique inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Establece el estilo de la letra utilizada para el texto

Propiedad	text-align
Valores	left right center justify inherit
Se aplica a	Elementos de bloque y celdas de tabla
Valor inicial	left
Descripción	Establece la alineación del contenido del elemento

Propiedad	font-variant
Valores	normal small-caps inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Establece el estilo alternativo de la letra utilizada para el texto

Propiedad	line-height
Valores	normal numero unidad de medida porcentaje inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Permite establecer la altura de línea de los elementos

CSS

- 5.-Texto.

- Tipografía.

Propiedad	text-decoration
Valores	none (underline overline line-through blink) inherit
Se aplica a	Todos los elementos
Valor inicial	none
Descripción	Establece la decoración del texto (subrayado, tachado, parpadeante, etc.)

Propiedad	letter-spacing
Valores	normal unidad de medida inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Permite establecer el espacio entre las letras que forman las palabras del texto

Propiedad	text-transform
Valores	capitalize uppercase lowercase none inherit
Se aplica a	Todos los elementos
Valor inicial	none
Descripción	Transforma el texto original (lo transforma a mayúsculas, a minúsculas, etc.)

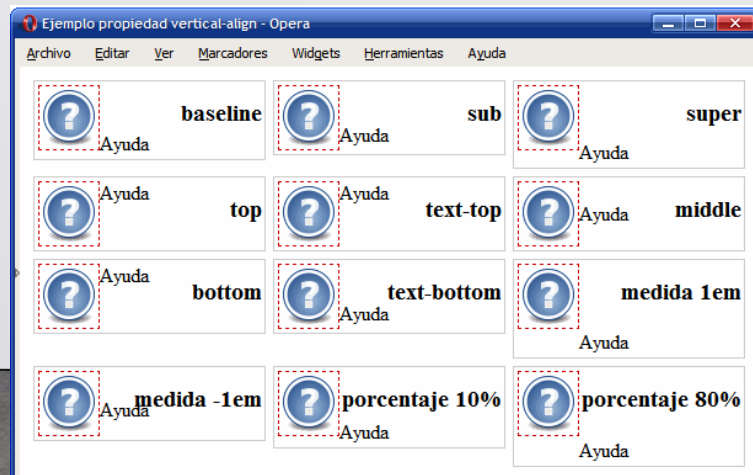
Propiedad	word-spacing
Valores	normal unidad de medida inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Permite establecer el espacio entre las palabras que forman el texto

CSS

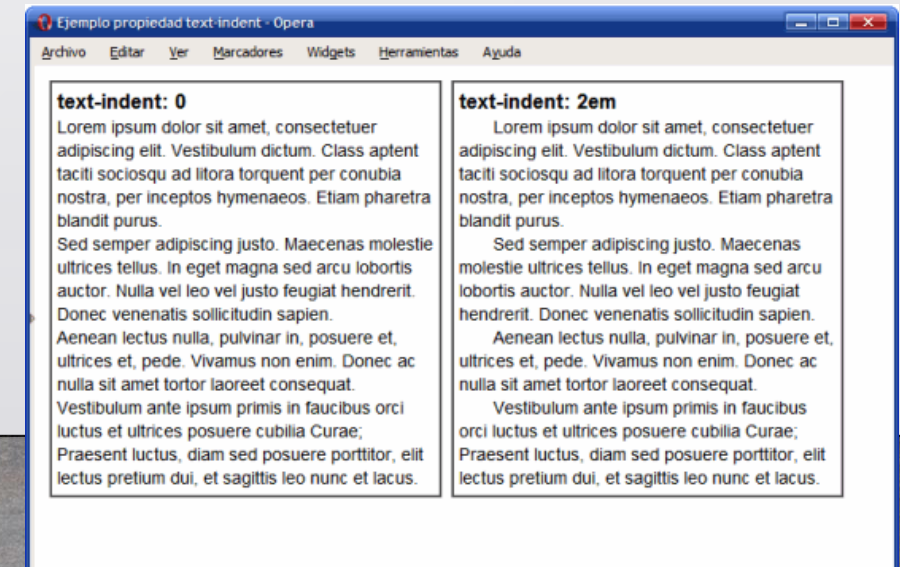
• 5.-Texto.

• Tipografía.

Propiedad	vertical-align
Valores	baseline sub super top text-top middle bottom text-bottom porcentaje unidad de medida inherit
Se aplica a	Elementos en línea y celdas de tabla
Valor inicial	baseline
Descripción	Determina la alineación vertical de los contenidos de un elemento



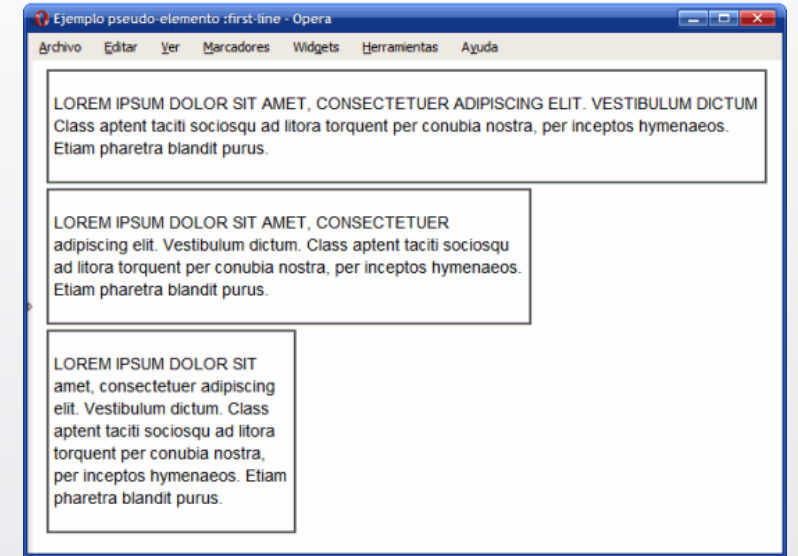
Propiedad	text-indent
Valores	unidad de medida porcentaje inherit
Se aplica a	Los elementos de bloque y las celdas de tabla
Valor inicial	0
Descripción	Tabula desde la izquierda la primera línea del texto original



CSS

• 5.-Texto. Pseudoelementos

- CSS define unos elementos especiales llamados "pseudo-elementos" que permiten aplicar estilos a ciertas partes de un texto o a una parte concreta del documento.
- Sintáxis:
 - selector::pseudo-elemento { código CSS }
- El pseudo-elemento **::first-line** permite aplicar estilos a la primera línea de un texto.



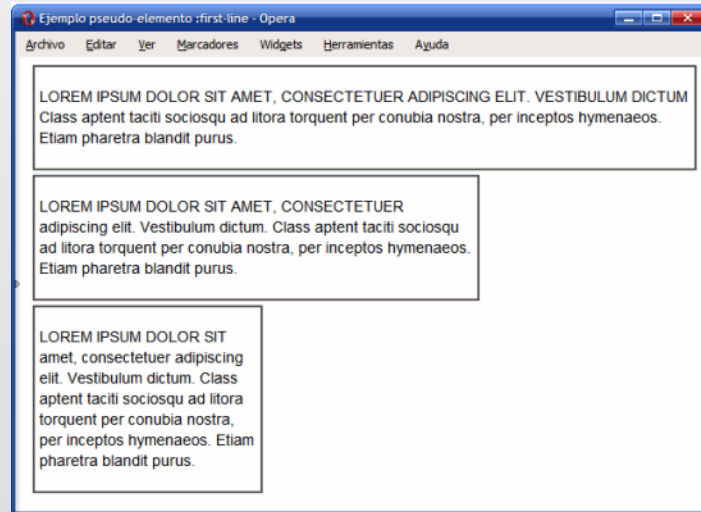
```
p::first-line {  
    text-transform: uppercase;  
}
```


CSS

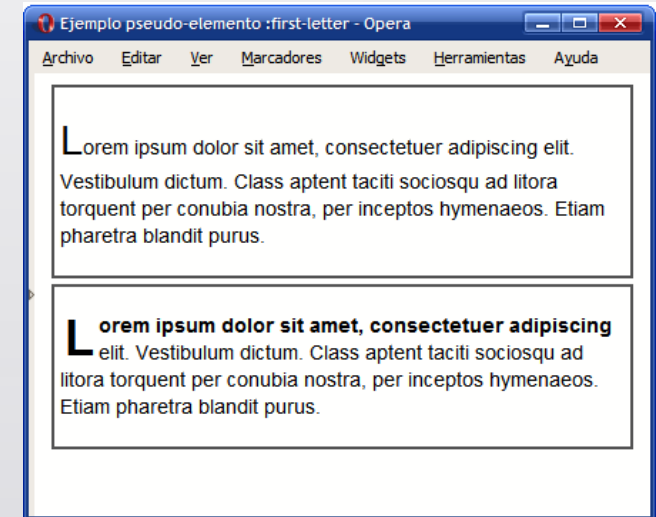
• 5.-Texto. Pseudoelementos

- CSS define unos elementos especiales llamados "pseudo-elementos" que permiten aplicar estilos a ciertas partes de un texto.
- El pseudo-elemento **::first-line** y **::first-letter** permite aplicar estilos a la primera línea de un texto y a la primera letra del párrafo.

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



```
#normal p::first-letter {  
  font-size: 2em;  
}  
#avanzado p::first-letter {  
  font-size: 2.5em;  
  font-weight: bold;  
  line-height: .9em;  
  float: left;  
  margin: .1em;  
}...
```





- **6.-Otros Pseudo elementos.**

- Otros pseudo elementos

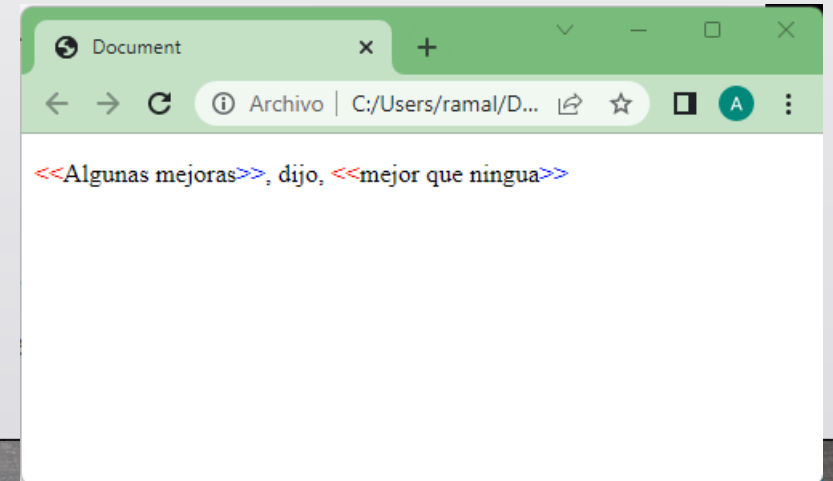
- ::after → crea un pseudo-elemento que es el último hijo del elemento seleccionado. Normalmente se usa para añadir elementos decorativos
- ::before → igual que ::after pero para el primer hijo del elemento seleccionado.
- Ejemplos a continuación.

CSS

• 6.-Otros Pseudo elementos.

```
<style type="text/css">
  span::after{
    content: ">>";
    color: blue;
  }
  span::before {
    content: "<<";
    color: red;
  }
</style>
</head>
<body>
  <p><span id="inicio">Algunas mejoras</span>, dijo, <span id="ultima">mejor que ninguna</span></p>
</body>
```

<https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-elements>



• 7.-Pseudoclasses en enlaces.

- Aplicar estilos a un enlace (<a...>)según su estado:
 - **:link**, aplica estilos a los enlaces que apuntan a páginas o recursos que aún no han sido visitados por el usuario.
 - **:visited**, aplica estilos a los enlaces que apuntan a recursos que han sido visitados anteriormente por el usuario. El historial de enlaces visitados se borra automáticamente cada cierto tiempo y el usuario también puede borrarlo manualmente.
 - **:hover**, aplica estilos al enlace sobre el que el usuario ha posicionado el puntero del ratón.
 - **:active**, aplica estilos al enlace que está pinchando el usuario. Los estilos sólo se aplican desde que el usuario pincha el botón del ratón hasta que lo suelta, por lo que suelen ser unas pocas décimas de segundo.
 - Por ejemplo: `a:visited { color: red; }`
 - Más en: <https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes>



Pseudoelementos y Pseudoclasses

Cascade style sheet

CSS

- **Pseudoclasses.**

- Es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado. Ya hemos visto en hiperenlaces: *link*, *active*, *visited* y *hover*.
- **:first-child** → el primer elemento entre un grupo de elementos hermanos.

- /* Selecciona cualquier <p> que sea el primer elemento entre sus hermanos */

```
p:first-child {  
    color: lime;  
}
```

Nulla mauris ante, dignissim et porta id, blandit dapibus quam. Proin dignissim lectus a tellus mollis porttitor. Ut est urna, molestie non ullamcorper id, euismod in tellus. Praesent nec neque in nibh pellentesque volutpat vitae vel diam. Morbi posuere maximus consectetur. Quisque euismod, libero vel elementum interdum, enim nunc dictum magna, id tempor eros elit in eros. Proin nec commodo felis, sed iaculis velit. Etiam quis sem a sem blandit luctus. Morbi molestie, sapien in lacinia vehicula, nulla nibh molestie magna, et tincidunt turpis neque tempor ipsum. Integer eu finibus est.

Ut vestibulum et erat eget finibus. Maecenas in tempor eros, eu blandit lacus. Aenean dolor quam, tristique in nisi eu, pulvinar mollis diam. Aliquam eu aliquam ligula, quis mattis nisl. Nulla cursus orci felis, tempor commodo quam condimentum eget. Cras quis vestibulum nunc. Aliquam ac magna ut est dignissim facilisis a eu orci. Aenean sem dolor, condimentum luctus ligula non, ultrices vehicula enim. Etiam eu libero non erat pharetra tristique ac a arcu. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur lacinia mauris quis turpis tristique, non interdum eros facilisis.

Nullam commodo sapien ut nunc mollis, ut suscipit elit rhoncus. Suspendisse in neque et ex pellentesque vulputate. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec consectetur orci nisi, sed vulputate lacus feugiat sed. Nulla facilisi. Fusce vel vehicula dui. Pellentesque nisi nisi, eleifend vel commodo eu, feugiat tempus arcu. Pellentesque vel sem felis. Curabitur eget neque sed nulla sollicitudin efficitur eu a nisi. Duis ornare rutrum nulla ac gravida. Integer id tincidunt est. Nulla facilisi. Vivamus viverra porttitor scelerisque. Nulla consectetur diam vel massa maximus, vitae tristique mi suscipit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.



• Anexo I. Pseudoclasses.

- **:last-child** → el último elemento entre un grupo de elementos hermanos.
 - /* Selecciona cualquier que esté dentro de y que sea el último elemento entre sus hermanos */

```
ul li:last-child {  
    color: lime;  
}
```

- Primer elemento
- Segundo elemento
- Tercer elemento
- Cuarto elemento

• Anexo I. Pseudoclasses.

- **:nth-child(n)** → coincide con uno o más elementos en función de su posición entre un grupo de hermanos.
 - /* Selecciona cada cuarto elemento entre cualquier grupo de hermanos */
:nth-child(4n) {
 color: lime;
}
- *odd* → impares
- *even* → pares
- $An+B$ → elementos cuya posición numérica en una serie de hermanos coincide con el patrón $An+B$, para cada entero positivo o valor cero de n . Por ejemplo, valores impares: $2n+1$.

- Primer elemento
- Segundo elemento
- Tercer elemento
- Cuarto elemento

CSS

• Pseudoclasses.

- **:last-of-type** → representa el último elemento de su tipo entre un grupo de elementos hermanos..

- /* Selecciona cualquier que sea el último elemento de su tipo entre sus hermanos */

```
li:last-of-type {  
    color: lime;  
}
```

- Primer elemento
- Segundo elemento
- Tercer elemento
- Cuarto elemento

- Para el primer elemento de su tipo usaremos **:first-of-type** de la misma forma.

CSS



- **Pseudoelementos.**

- Los pseudoelementos permiten añadir estilos a una parte concreta del documento.
- Notación.
 - Con la especificación CSS3 se quiere diferenciar entre pseudo-elementos y pseudo-clases por lo que define:
 - El uso de `:` para selectores de pseudo-clases.
 - El uso de `::` para selectores de pseudo-elementos.
 - De esta forma hay una diferenciación visual.
 - Aún así, en CSS2 se usaba la notación de `“:”` para pseudoelementos también, por lo que aún sigue siendo compatible con la mayoría de navegadores.

CSS

• Anexo I. Pseudoelementos. Quotes, after y before

- La propiedad **quotes**, que define las comillas que acompaña a un elemento, sólo tiene sentido cuando se utiliza junto con la propiedad content y los pseudo-elementos **::after** y **::before**.
 - **::after** → junto con la propiedad content, normalmente lo usaremos para insertar contenido después de un elemento
 - **::before** → junto con la propiedad content, normalmente lo usaremos para insertar contenido antes de un elemento.
 - Por ejemplo:

```
<style type="text/css">
  h1::before {
    content: "Capítulo";
  }
</style>
```

```
<h1>1</h1>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vulputate
  porta laoreet. Nunc nulla purus, bibendum a sem quis, vehicula euismod
  tellus. Aliquam lobortis nunc et massa feugiat, in egestas quam maximus.
  Cras elementum enim quis venenatis congue. Aenean aliquam euismod magna
  non mattis. Sed maximus tellus eu lectus eleifend, sit amet sagittis
  turpis placerat. Morbi porta leo in dapibus rutrum. Sed pulvinar in nisi
  non molestie. Pellentesque eget maximus sem. Fusce pellentesque tempor
  fringilla. Morbi euismod, massa sit amet vehicula elementum, ligula dui
  porttitor nulla, in porta felis velit ac est. Nulla pharetra est odio, in
  dapibus odio sollicitudin egestas. Duis iaculis tellus commodo diam
  suscipit, in hendrerit enim consectetur. Sed nec sagittis sapien, id
  aliquam felis. In viverra consequat tincidunt.
</p>
<h1>2</h1>
<p>
  Etiam dapibus erat urna, sit amet auctor felis facilisis vel. Quisque sed
  eleifend ipsum. Cras at tortor ante. Phasellus eget maximus risus.
  Pellentesque placerat, libero nec laoreet finibus, tellus urna egestas
  turpis, id posuere nulla felis sit amet velit. Suspendisse et velit
  molestie, maximus massa ac, scelerisque enim. Maecenas tortor sapien,
  maximus ac egestas quis, iaculis in metus. Duis tristique congue erat in
```

Capítulo1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vulputate porta laoreet. Nunc nulla purus, bibendum a sem quis, vehicula euismod tellus. Aliquam lobortis nunc et massa feugiat, in egestas quam maximus. Cras elementum enim quis venenatis congue. Aenean aliquam euismod magna non mattis. Sed maximus tellus eu lectus eleifend, sit amet sagittis turpis placerat. Morbi porta leo in dapibus rutrum. Sed pulvinar in nisi non molestie. Pellentesque eget maximus sem. Pellentesque tempor fringilla. Morbi euismod, massa sit amet vehicula elementum, ligula dui porttitor nulla, in porta felis velit ac est. Nulla pharetra est odio, in dapibus odio sollicitudin egestas. Duis iaculis tellus commodo diam suscipit, in hendrerit enim consectetur. Sed nec sagittis sapien, id aliquam felis. In viverra consequat tincidunt.

Capítulo2

Etiam dapibus erat urna, sit amet auctor felis facilisis vel. Quisque sed eleifend ipsum. Cras at tortor ante. Phasellus eget maximus risus. Pellentesque placerat, libero nec laoreet finibus, tellus urna egestas turpis, id posuere nulla felis sit amet velit. Suspendisse et velit molestie, maximus massa ac, scelerisque enim. Maecenas tortor sapien, maximus ac egestas quis, iaculis in metus. Duis tristique congue erat in viverra consequat tincidunt. Duis dictum fringilla velit ut pharetra.

CSS

• Pseudoelementos. Quotes, after y before

- Quotes.
- Permite indicar un número ilimitado de pares de comillas. Cuando sólo se define un par, se muestran siempre que se utilice *open-quote* y *close-quote*.

```
<style>
blockquote,
span {
  quotes: "«»" "»»" /
}
blockquote:before {
  content: open-quote;
}

blockquote:after {
  content: close-quote;
}
```

```
<blockquote>
  El éxito es la suma de pequeños esfuerzos repetidos día tras día
</blockquote>
```

«» El éxito es la suma de pequeños esfuerzos repetidos día tras día »»

CSS

• Anexo I. Pseudoelementos. Quotes, after y before

- Quotes.
 - Cuando se definen varios pares, se van alternando cada vez que se muestran unas comillas dentro de otro elemento que ya tiene comillas.
 - Cada comilla se separa de las demás mediante un espacio en blanco, por lo que no debe utilizarse una coma o cualquier otro carácter de separación.

```
<style>
blockquote,
span {
  quotes: "«»" "»»" "—" "—";
}
blockquote::before, span::before {
  content: open-quote;
}

blockquote::after, span::after {
  content: close-quote;
}
```

Cuando hay anidamiento, se aplicará el segundo par de comillas.

```
<blockquote>
  El éxito es la suma de pequeños esfuerzos repetidos día tras día
</blockquote>

<blockquote>
  Con diez cañones por banda, <span>viento en popa</span> a toda vela,
  <span>no corta el mar</span>, sino vuela un velero bergantín;
</blockquote>
```

««« El éxito es la suma de pequeños esfuerzos repetidos día tras día »»»

««« Con diez cañones por banda, --viento en popa-- a toda vela, --no corta el mar--, sino vuela un velero bergantín; »»»

• Anexo I. Pseudoelementos. Quotes, after y before

• Quotes.

- En el ejemplo anterior, tanto `<blockquote>` como `` establecen dos pares de comillas.
- Cuando uno de estos dos elementos se encuentre dentro de otro elemento que ya muestra comillas, la propiedad **quotes** indica que se debe utilizar el segundo par de comillas en vez del primero.
- El primer y tercer valor corresponderán con **open-quote** y el segundo y cuarto como **close-quote**.

```
<blockquote>
| El éxito es la suma de pequeños esfuerzos repetidos día tras día
</blockquote>

<blockquote>
| Con diez cañones por banda, <span>viento en popa</span> a toda vela,
| <span>no corta el mar</span>, sino vuela un velero bergantín;
</blockquote>
```

««« El éxito es la suma de pequeños esfuerzos repetidos día tras día »»»

««« Con diez cañones por banda, --viento en popa-- a toda vela, --no corta el mar--, sino vuela un velero bergantín; »»»



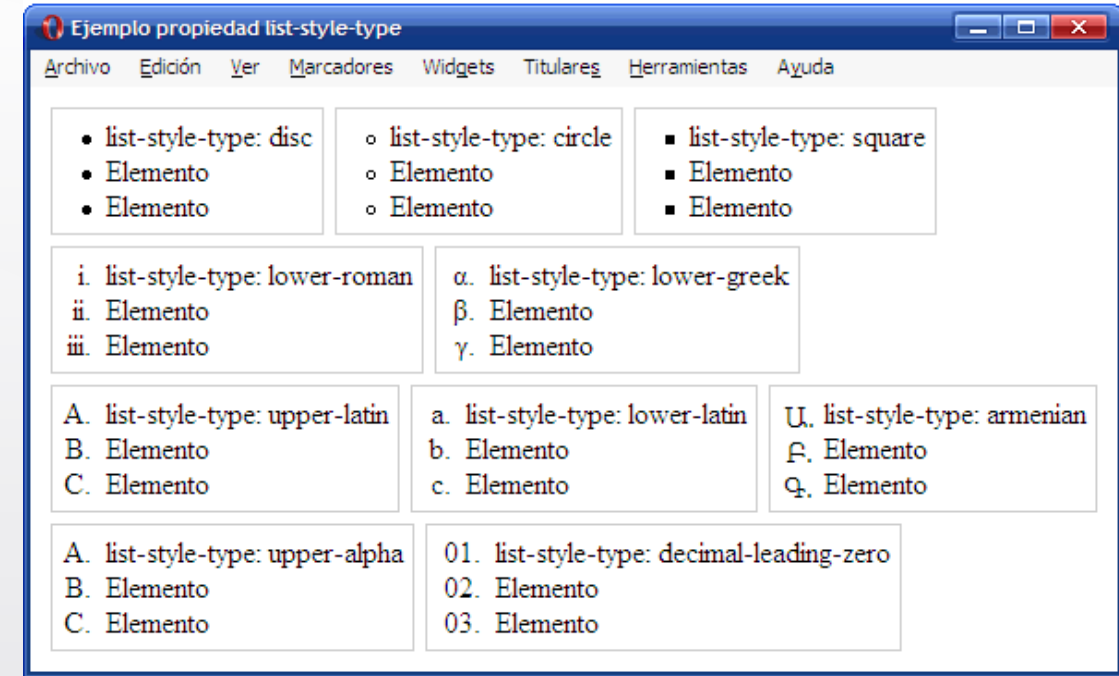
Css – Parte II

Cascade style sheet

• 8.-Listas.

Propiedad	list-style-type
Valores	disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian lower-alpha upper-alpha none inherit
Se aplica a	Elementos de una lista
Valor inicial	disc
Descripción	Permite establecer el tipo de viñeta mostrada para una lista

Propiedad	list-style-position
Valores	inside outside inherit
Se aplica a	Elementos de una lista
Valor inicial	outside
Descripción	Permite establecer la posición de la viñeta de cada elemento de una lista



Propiedad	list-style-image
Valores	url none inherit
Se aplica a	Elementos de una lista
Valor inicial	none
Descripción	Permite reemplazar las viñetas automáticas por una imagen personalizada

CSS

• 9.-Tablas.

Propiedad	border-collapse
Valores	collapse separate inherit
Se aplica a	Todas las tablas
Valor inicial	separate
Descripción	Define el mecanismo de fusión de los bordes de las celdas adyacentes de una tabla

Propiedad	border-spacing
Valores	unidad de medida unidad de medida? inherit
Se aplica a	Todas las tablas
Valor inicial	0
Descripción	Establece la separación entre los bordes de las celdas adyacentes de una tabla

Propiedad	empty-cells
Valores	show hide inherit
Se aplica a	Celdas de una tabla
Valor inicial	show
Descripción	Define el mecanismo utilizado para el tratamiento de las celdas vacías de una tabla

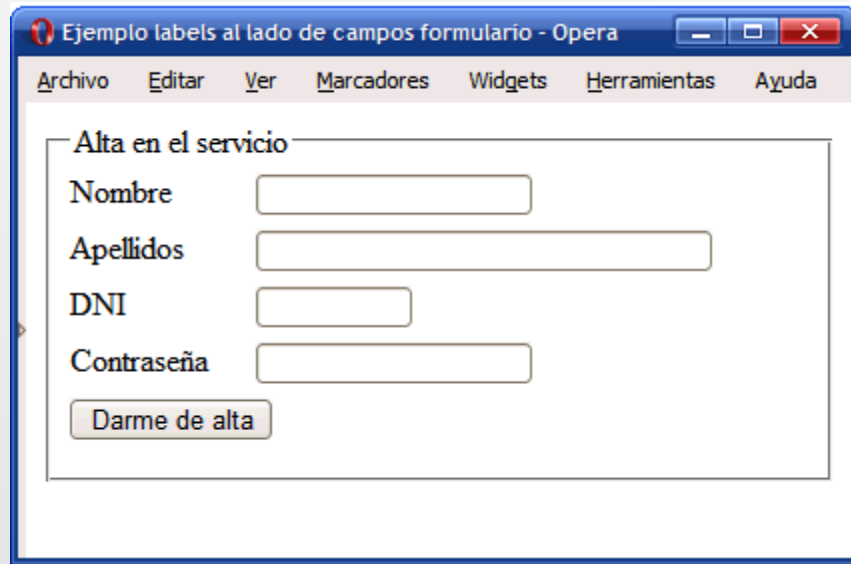
Propiedad	caption-side
Valores	top bottom inherit
Se aplica a	Los elementos caption
Valor inicial	top
Descripción	Establece la posición del título de la tabla

Puedes probar la pseudoclase **:hover** en tablas.

CSS

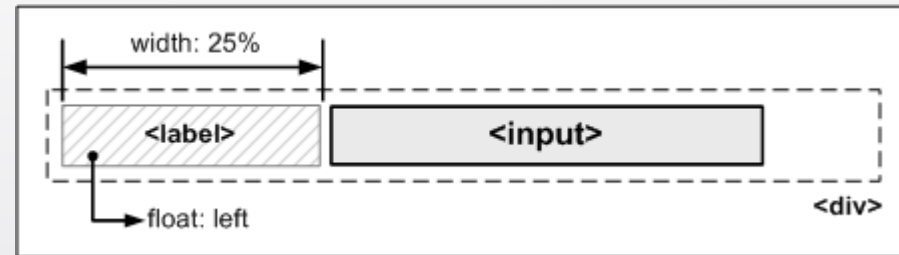
- 10.-Formularios.

- Para alinear formularios por ejemplo:



Pseudoclase → **:focus**. Cuando obtiene el foco

```
input:focus {  
    border: 2px solid black;  
    background-color: #F3F3F3;  
}
```



HTML

```
<form>  
<fieldset>  
  <legend>Alta en el servicio</legend>  
  
  <div>  
    <label for="nombre">Nombre</label>  
    <input type="text" id="nombre" />  
  </div>  
  
  <div>  
    <label for="apellidos">Apellidos</label>  
    <input type="text" id="apellidos" size="35"/>  
  </div>...  
</fieldset>  
</form>
```

CSS

```
div {  
    margin: .4em 0;  
}  
div label {  
    width: 25%;  
    float: left;  
}
```

CSS

• 11.-Layout.

- Para evitar los grandes espacios en blanco se crean páginas con una anchura fija adecuada y centramos la página horizontalmente respecto de la ventana del navegador.

```
#contenedor {  
  width: 300px;  
  margin: 0 auto;  
}
```

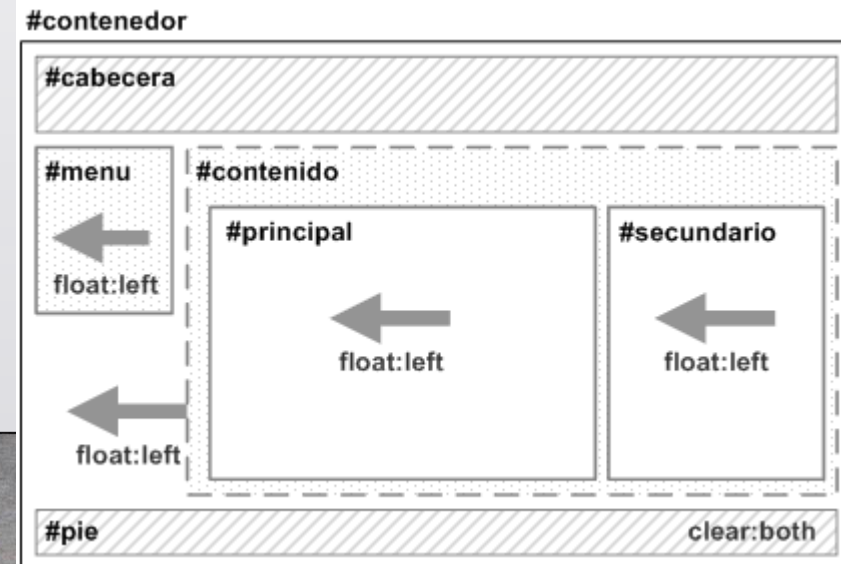
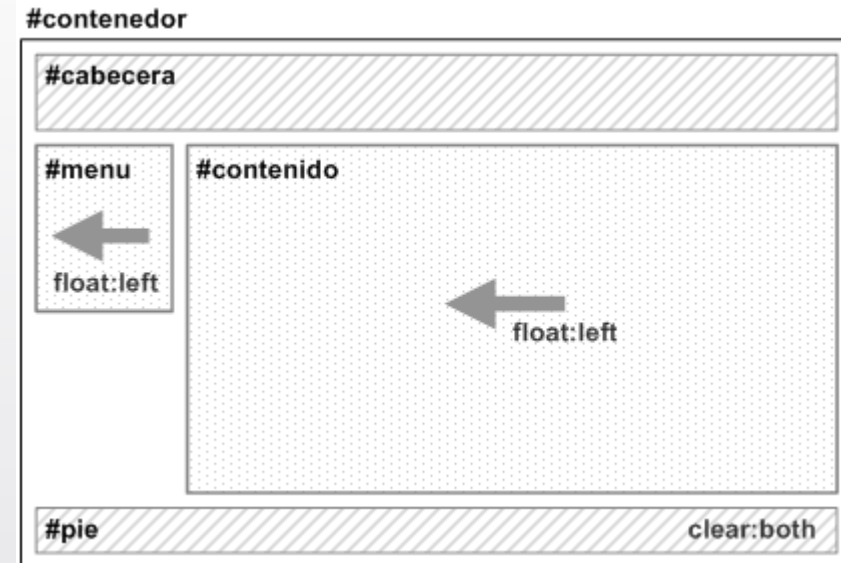
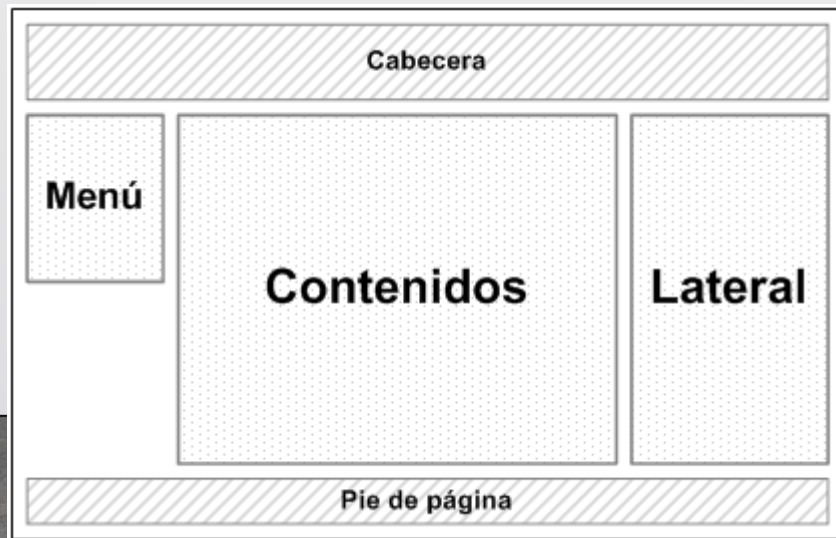
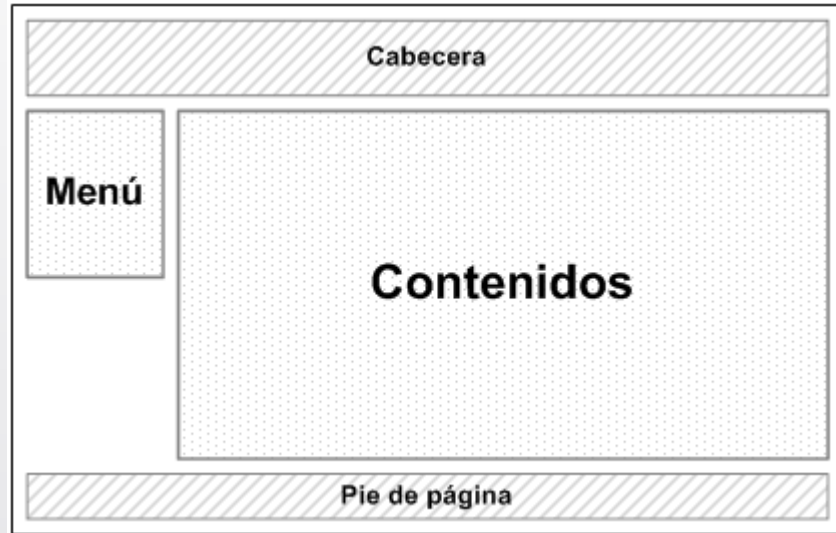
O podemos hacerlo fluido:

```
#contenedor {  
  width: 70%;  
  margin: 0 auto;  
}
```

```
<body>  
  <div id="contenedor">  
    <header>....  
    </header>  
    ...  
  </div>
```



• 11.-Layout. Diseño común.





- **11.-Layout. Media queries.**

- Técnica que permite aplicar diferentes estilos al mismo documento HTML en función de las características del dispositivo o de la ventana del navegador.
- Sintaxis:
 - @media condición { ... código CSS }
- Condiciones:
 - @media screen and (max-width: 600px) { ... código CSS ... }
 - @media (min-width: 700px) { ... código CSS ... }
 - @media (min-width: 250px) and (max-width: 550px) { ... código CSS ... }

CSS



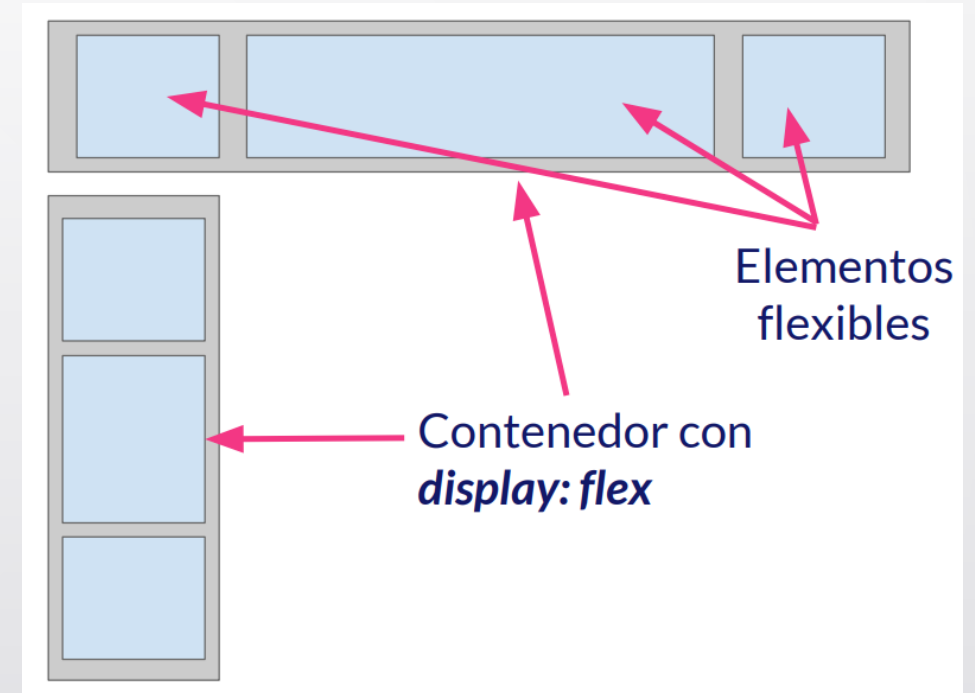
- **10.-Layout. Media queries.**

- **min-width y max-width:** Estas condiciones permiten definir estilos para diferentes rangos de anchura de la ventana del navegador.
- **min-height y max-height:** Estas condiciones permiten definir estilos para diferentes rangos de altura de la ventana del navegador.
- **orientation:** Esta condición permite definir estilos diferentes para dispositivos en modo vertical (con la pantalla en posición vertical) y en modo horizontal (con la pantalla en posición horizontal).
- **aspect-ratio:** Esta condición permite definir estilos para diferentes proporciones de aspecto de la pantalla. Por ejemplo, se pueden definir estilos específicos para pantallas panorámicas con una proporción de aspecto de 16:9.
- **color:** Esta condición permite definir estilos diferentes para dispositivos con diferentes capacidades de color. Por ejemplo, se pueden definir estilos específicos para dispositivos en blanco y negro, o para dispositivos con pantallas de alta calidad de color.
- **resolution:** Esta condición permite definir estilos diferentes para diferentes resoluciones de pantalla. Por ejemplo, se pueden definir estilos específicos para dispositivos de alta resolución (como las pantallas retina de Apple).

CSS

• 10.-Layout. Flex.

- Con la aparición de CSS3 aparece un nuevo valor para la propiedad display. (**display: flex;**).
- La idea es tener un elemento que va a poder controlar las propiedades de los elementos que contiene.
- Dos elementos:
 - El contenedor con **display: flex**.
 - Los elementos flexibles que son **hijos** del contenedor.





- **10.-Layout. Flex.**

- Propiedades de los elementos:
 - La altura
 - La anchura
 - El orden
 - La alineación vertical
 - La alineación horizontal
 - La distribución a lo largo del contenedor

CSS



- **10.-Layout. Flex.**

- Propiedades de los elementos:

- **Dirección de los elementos**

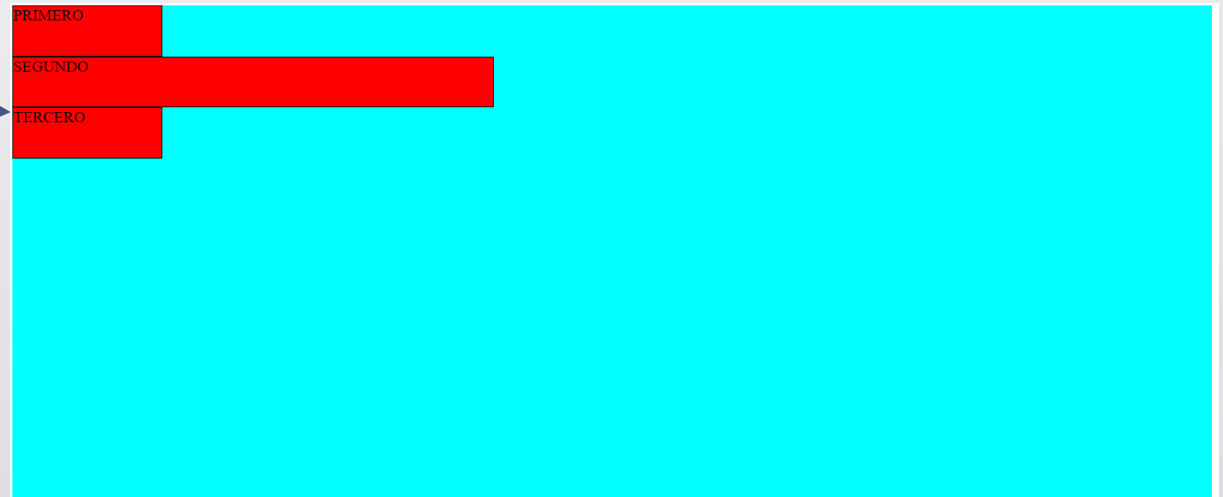
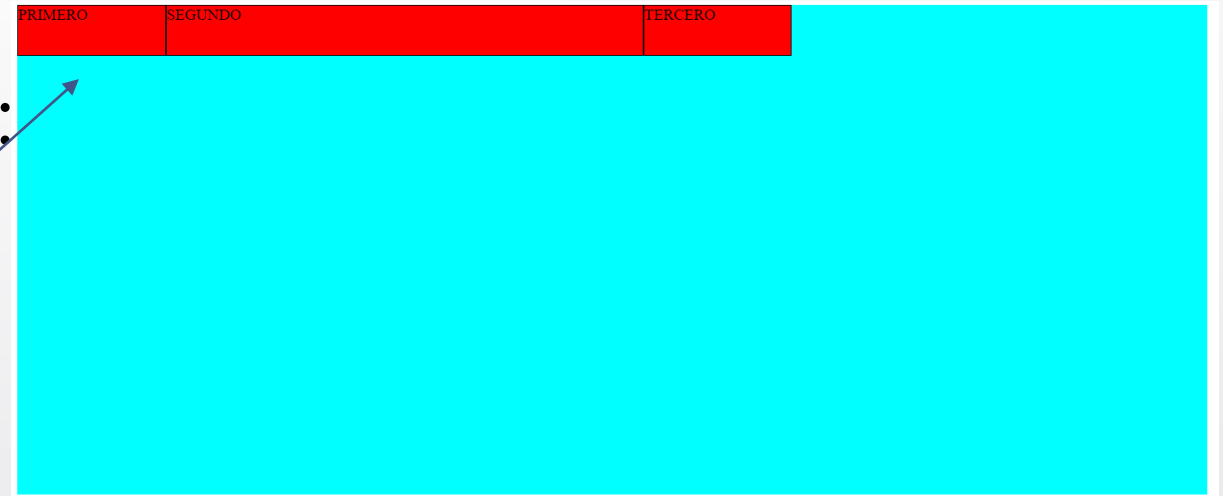
- Propiedad: **flex-direction**

- row (valor por defecto)

- row-reverse

- column

- column-reverse

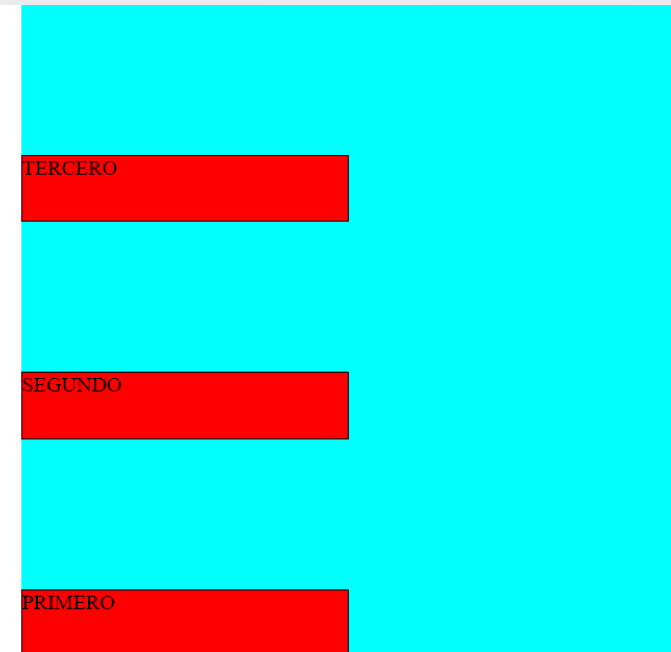
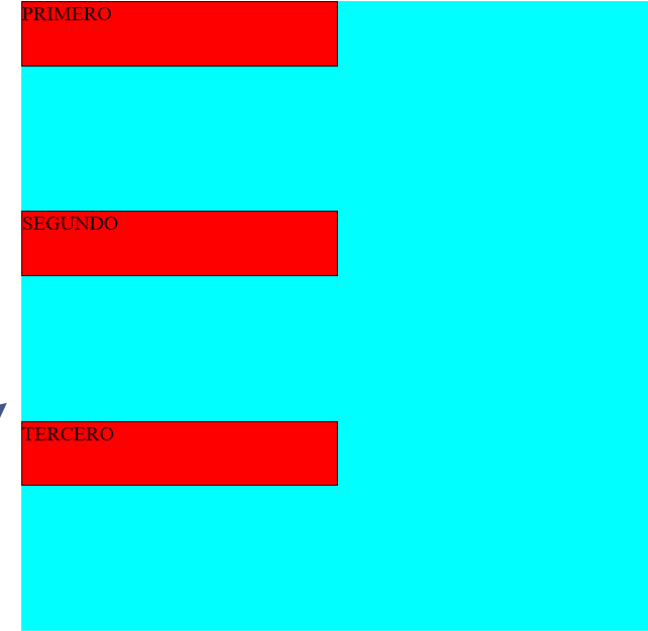


• 10.-Layout. Flex.

- Propiedades de los elementos:

- **Ajuste de los elementos**(si no caben):

- Propiedad: *flex-wrap*.
 - nowrap. Por defecto, aparecen en una sola línea.
 - **wrap**. Los elementos se disponen en varias líneas si no caben. De arriba abajo
 - **wrap-reverse**. Igual que antes pero de abajo a arriba.

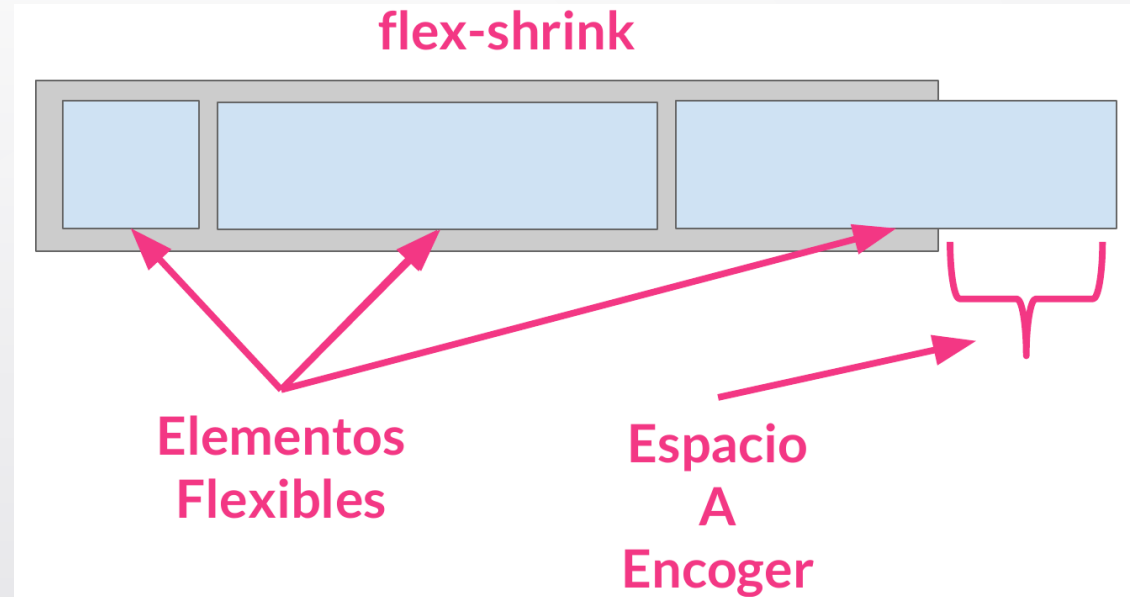
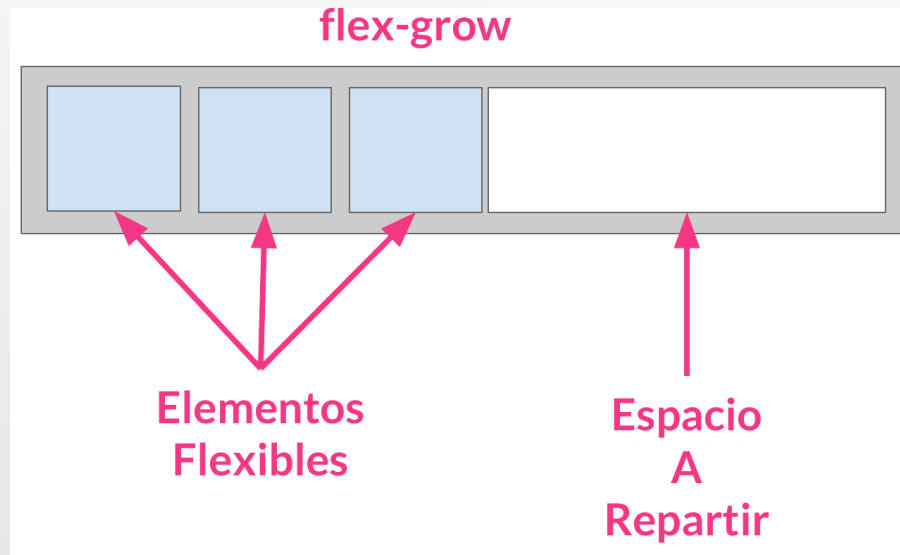




- **10.-Layout. Flex.**

- Podemos unir las dos reglas anteriores en una:
 - **flex-flow:** flex-direction flex-wrap
 - Por ejemplo:
 - flex-flow: column wrap-reverse
 - flex-flow: column-reverse wrap

• 10.-Layout. Flex.



- Propiedad uniendo las tres propiedades anteriores:
 - `flex: flex-grow flex-shrink flex-basis`



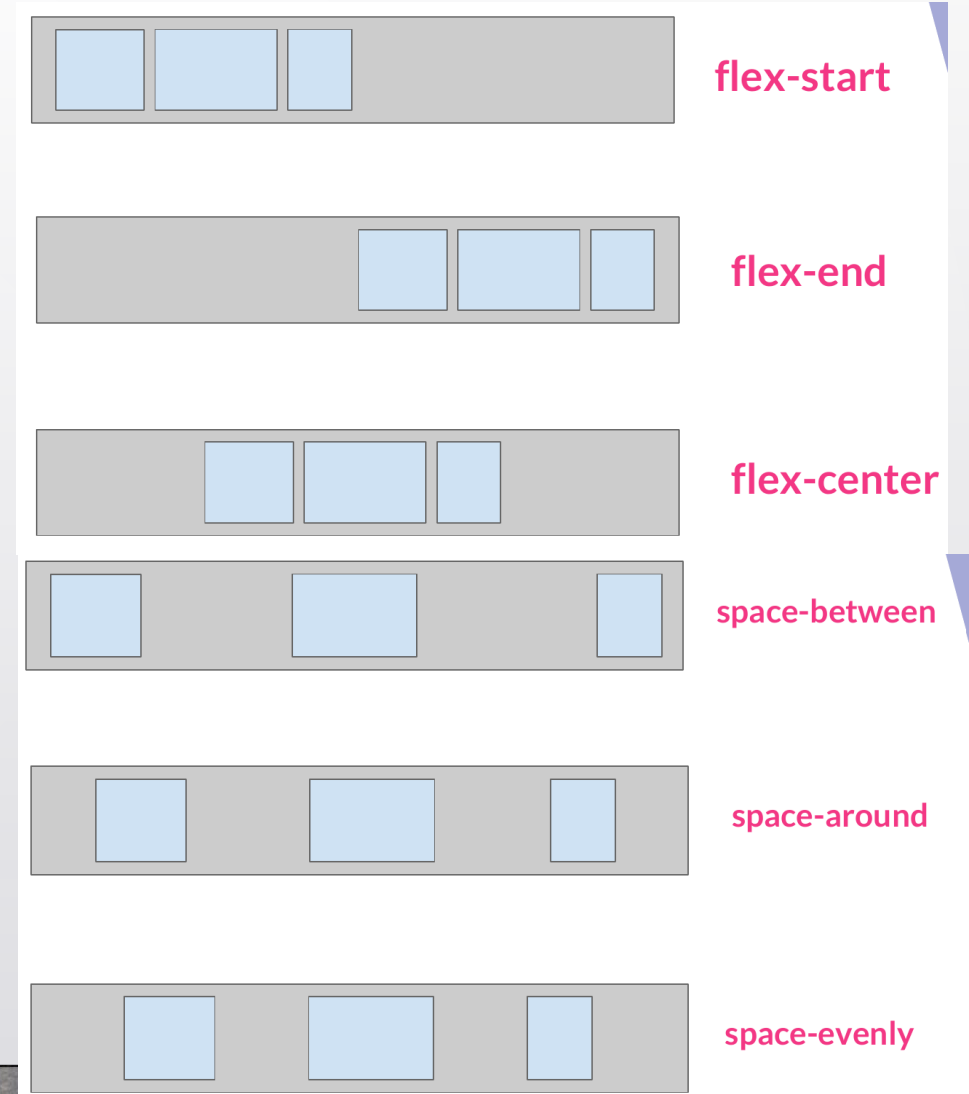
- **10.-Layout. Flex.**

- Para modificar el **tamaño** de los elementos flexibles:
 - Propiedades de elementos dentro del contenedor:
 - **flex-grow**: el factor de crecimiento de un elemento flexible cuando se distribuye el espacio restante. Es decir, si hay espacio disponible, cuanto se le asigna a cada elemento del contenedor flex.(por defecto: 0)
 - **flex-shrink**: el factor de contracción de un elemento flexible cuando sobrepasan el tamaño del contenedor. Es decir, cuanto puede encoger un elemento en relación con los demás elementos. Cuánto más valor, más encoje (por defecto: 1).
 - **flex-basis**: Especifica la base flexible, la cual es el tamaño inicial de un elemento flexible. Si el elemento no cabe se adapta al contenido. (por defecto: auto)

• 10.-Layout. Flex.

• Alineación horizontal de elementos flex.

- Propiedad: **justify-content**.
- Valores:
 - flex-start
 - flex-end
 - center
 - space-between
 - space-around
 - space-evenly.

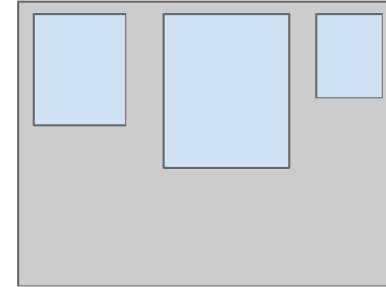


10.-Layout. Flex.

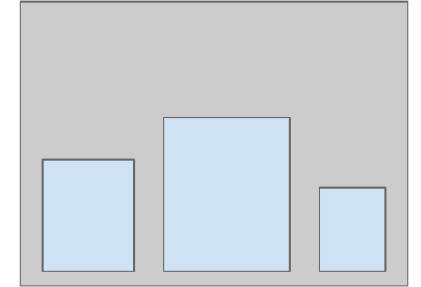
- Alineación vertical de elementos flex

- Propiedad: **align-items**
- Valores:
 - flex-start
 - flex-end
 - center
 - stretch
 - baseline

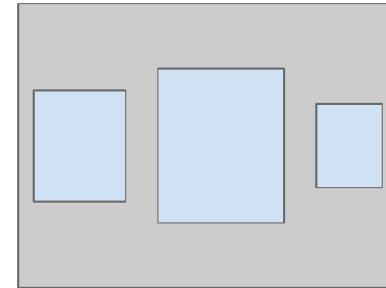
flex-start



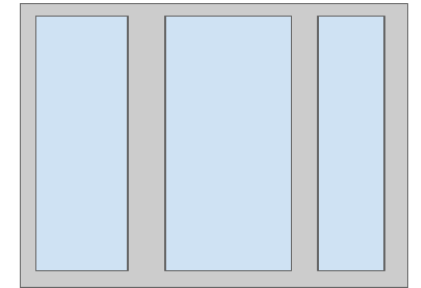
flex-end



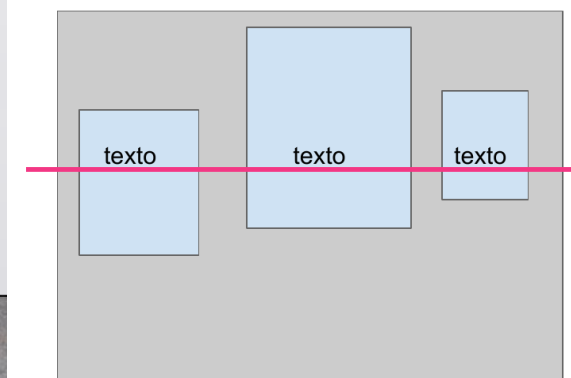
center



stretch (no height)



baseline



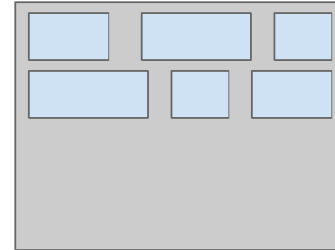
• 10.-Layout. Flex.

CSS

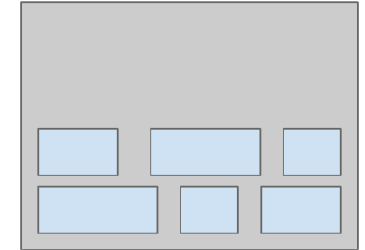
• Alineación vertical de wrap

- Propiedad: **align-content**
 - Tengo varias líneas de elementos flexibles y quiero controlarlos.
- Valores:
 - flex-start
 - flex-end
 - center
 - stretch
 - space-between
 - space-around

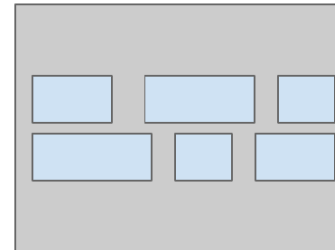
flex-start



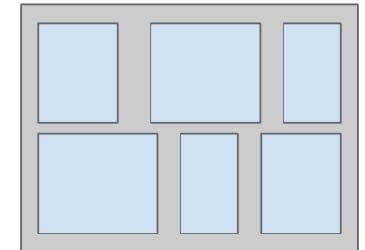
flex-end



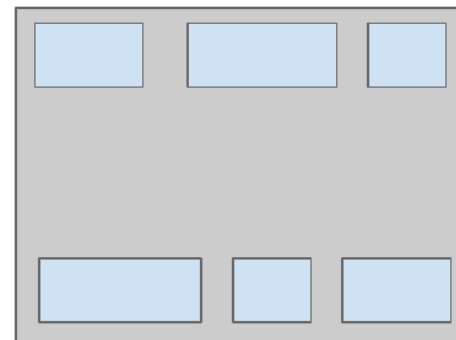
center



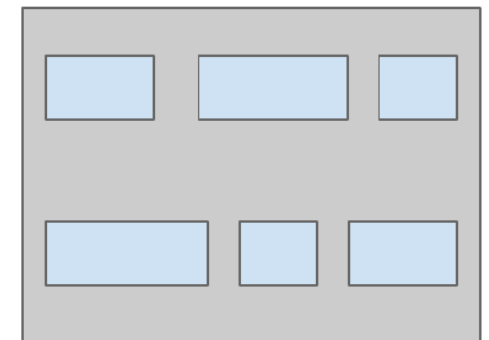
stretch (no height)



space-between



space-around

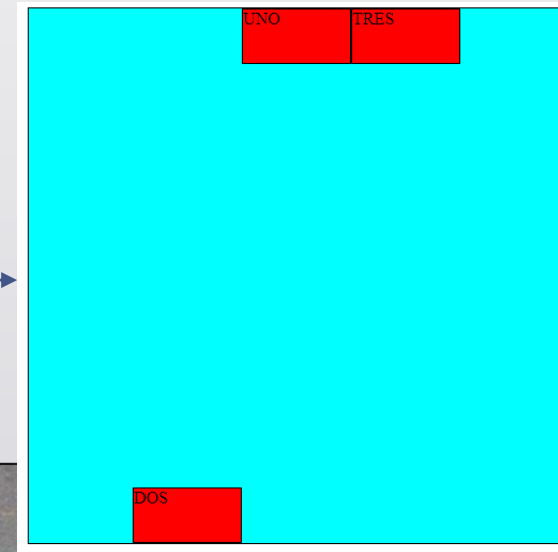


CSS

• 10.-Layout. Flex.

- Se puede modificar el **orden** en el que se van a mostrar los elementos flexibles.
- Propiedad de elemento flex-> **order**: *número*; /*Por defecto es 0*/
- Cuanto menor es el valor, antes se muestra el elemento, en caso de empate, el que esté antes en el código.

```
✓ .contenedor div {  
  background-color: red;  
  width: 100px;  
  height: 50px;  
  border: 1px solid black;  
  order: 2;  
}  
✓ #dos {  
  align-self: flex-end;  
  order: 1;  
}
```



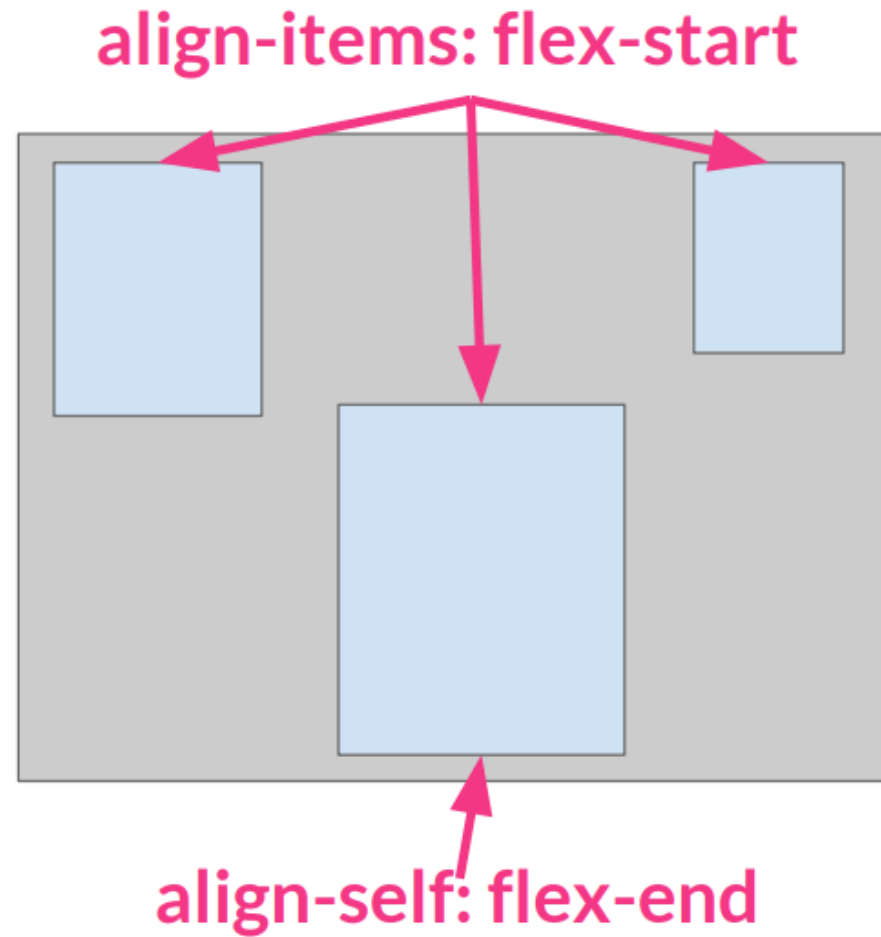


- **10.-Layout. Flex.**

- Desde el contenedor FLEX podíamos controlar la alineación vertical de todos los elementos flexibles a la vez con la propiedad **align-items**.
- Si queremos que un elemento tenga una alineación propia debemos usar la propiedad **align-self**.
- Valores propiedad: **align-self**
 - flex-start
 - flex-end
 - center
 - stretch (no height)
 - baseline
- Los elementos con esta propiedad no se ven afectados por vertical-align

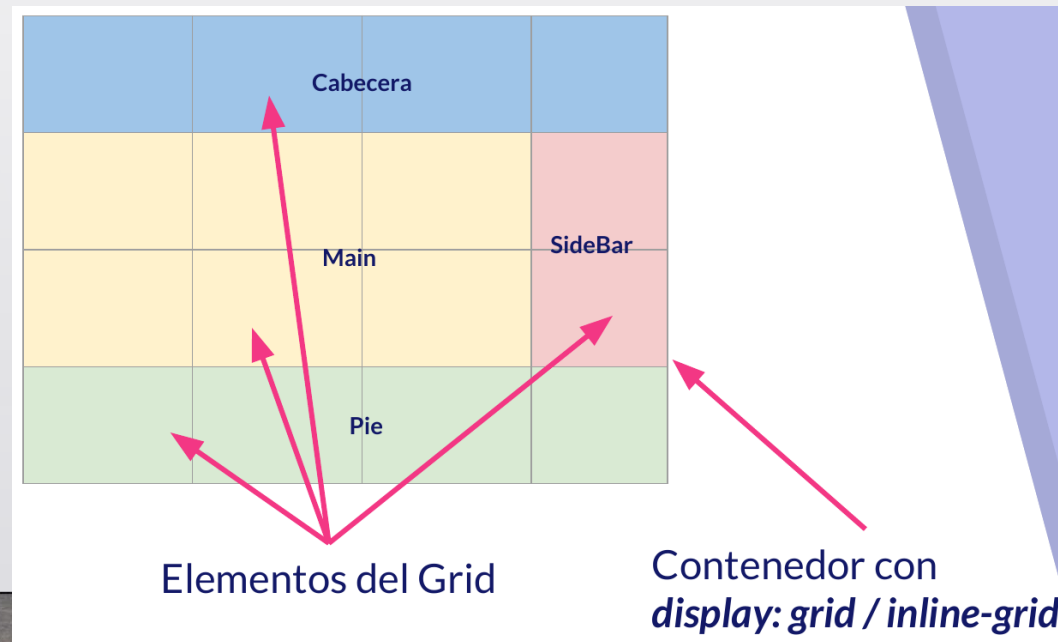
• 10.-Layout. Flex

- align-self



11.-Layout. Grid. CSS

- Con la aparición de CSS3 aparece un nuevo valor para la propiedad *display*. (*display:grid* o *display:inline-grid*).
- La idea es crear una rejilla o tabla donde definiremos el número de filas y columnas. Además podremos controlar el comportamiento de los elementos que lo componen.





- **11.-Layout. Grid.**

- Propiedades del contenedor.
 - **Estructura en filas y columnas y la separación entre ellas.**
 - **grid-template-columns** : Número de columnas del grid y tamaño de las mismas. Debemos especificar tantos valores como columnas queramos en la plantilla.
 - **grid-template-rows**: Número de filas del grid y tamaño de las mismas. Debemos especificar tantos valores como filas queramos en la plantilla.
 - **row-gap**: Separación entre las filas.
 - **column-gap**: Separación entre las columnas.

CSS



- **11.-Layout. Grid.**

- Propiedades del contenedor.

- Ejemplos:

- grid-template-columns: 20% 50% 30%
 - grid-template-columns: 100px auto 100px 100px
 - grid-template-columns: auto auto auto auto auto
 - grid-template-columns : [id] 100px [nombre] 300px [apellidos] auto
 - En esta última hemos definido 3 columnas. La primera se llamará id y ocupa 100px, la segunda nombre y ocupa 300px y la última se llama apellidos y obtiene el espacio automáticamente.

- Para las filas igual usando grid-template-rows.

• 11.-Layout. Grid.

• Elementos GRID.

- Los elementos hijos directos del elemento con la propiedad CSS display: grid.
- Propiedades de dichos elementos:
 - Área del GRID que va a ocupar
 - Alineación horizontal y vertical de forma individual.
 - Colocación implícita.
 - No les afecta:
float, display:inline-block, display-table-cell, vertical-align o column-*

CSS

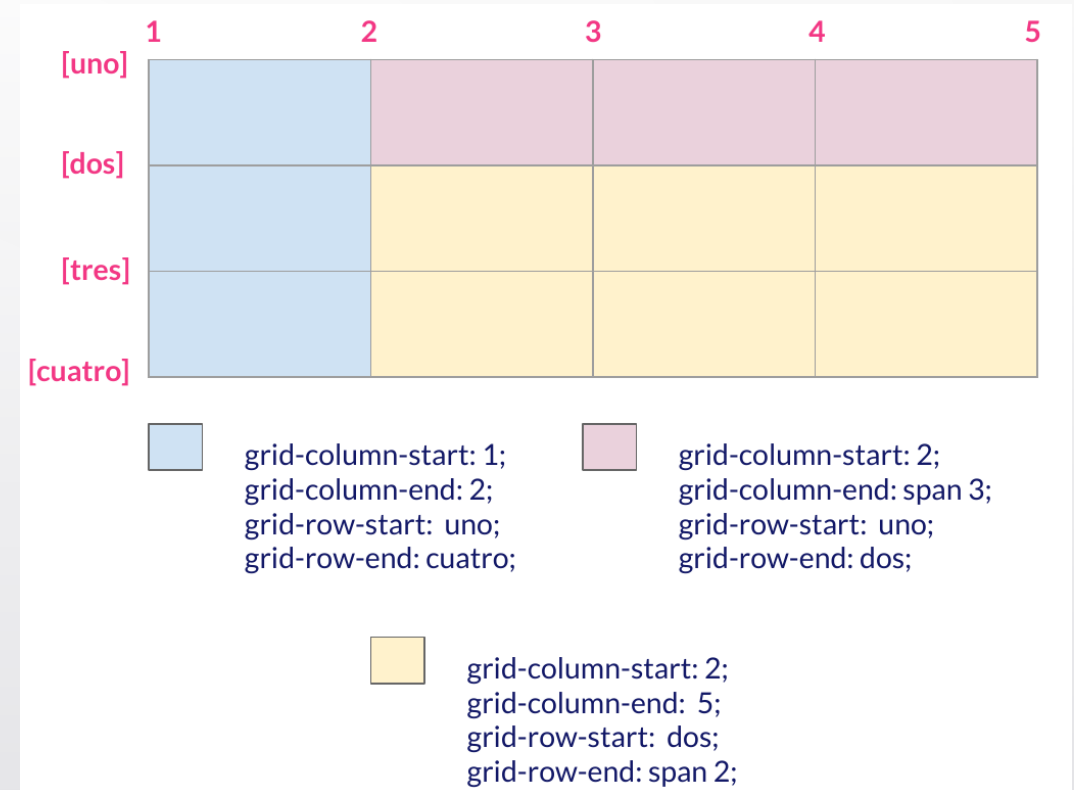
• 10.-Layout. Grid.

- Elementos GRID.

- Especificar el área que ocupan:
 - grid-column-start y grid-column-end
 - grid-row-start y grid-row-end

Uniendo propiedades:

- grid-column: start/end; → grid-column:1/2
- grid-row: start/end; → grid-row:uno/cuatro
- grid: row-start column-start row-end column-end;
grid: uno 1 cuatro 2

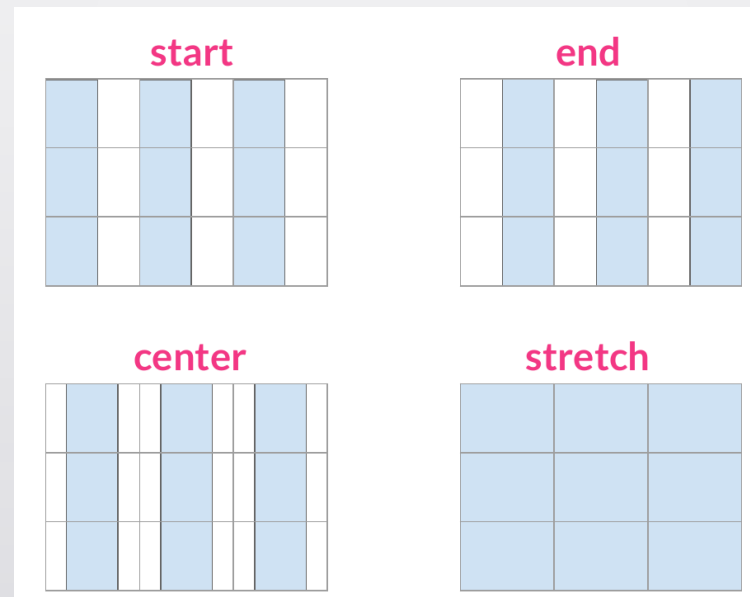


CSS

- 11.-Layout. Grid.

- Propiedades del contenedor. **Alineación horizontal del contenido.**

- Para alinear horizontalmente el contenido que hay dentro de cada celda usamos:
 - **justify-items:** start | end | center | stretch (por defecto)

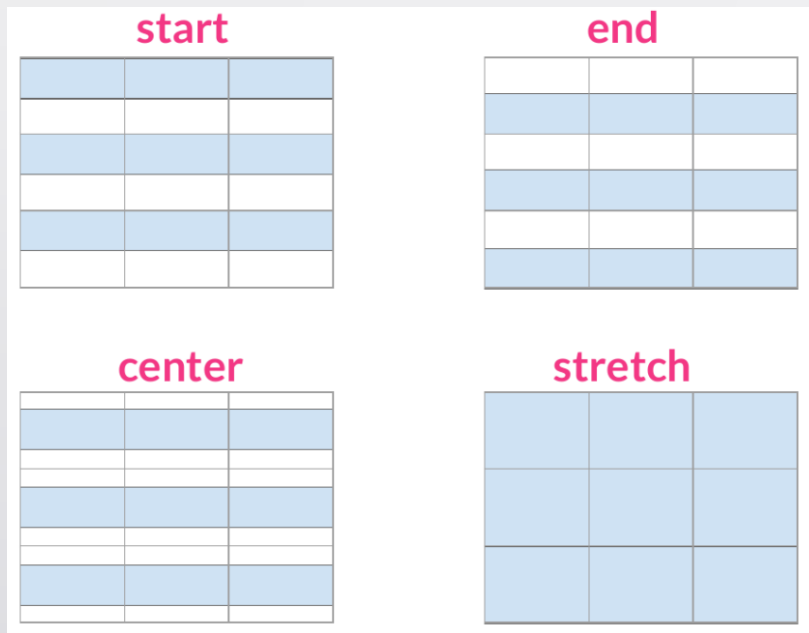


CSS

• 11.-Layout. Grid.

- Propiedades del contenedor. **Alineación vertical del contenido.**

- Para alinear verticalmente el contenido que hay dentro de cada celda usamos:
 - **align-items:** start | end | center | stretch (por defecto)



Puedo unir las propiedades `justify-items` y `align-items` en una sola:

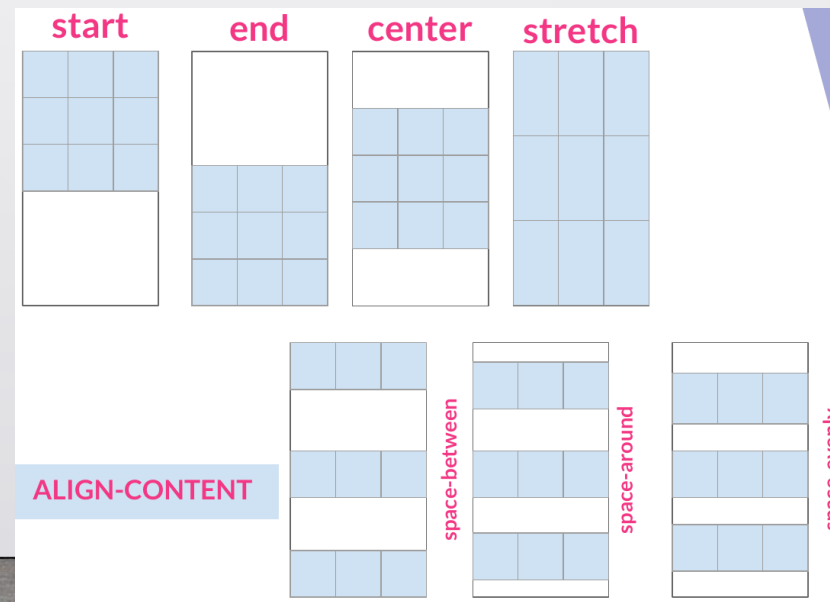
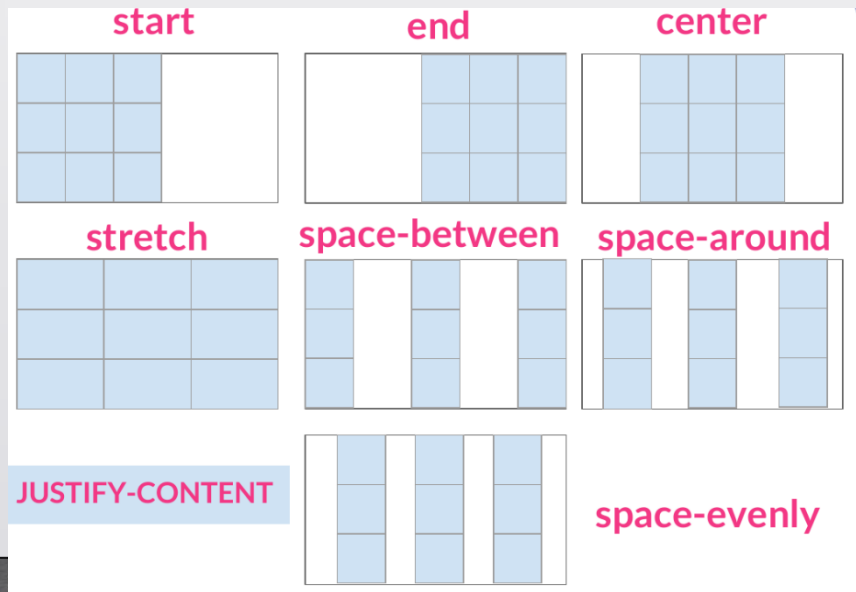
place-items: vertical horizontal
place-items: end start

CSS

- **11.-Layout. Grid.**

- Propiedades del contenedor. **Contenedor GRID.**

- En el caso de que el contenedor GRID no ocupe todo el espacio podemos distribuirlo con las propiedades CSS **justify-content** (horizontal) y **align-content** (vertical)



Las dos propiedades con:

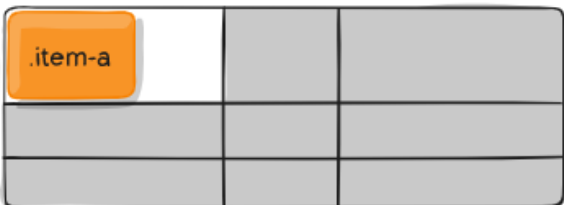
place-content: vertical horizontal
place-content: start center

CSS

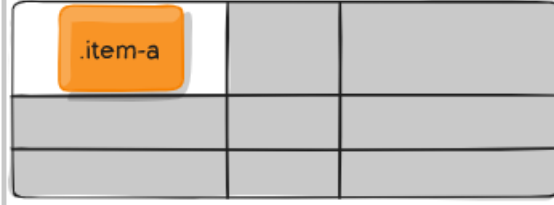
• 10.-Layout. Grid.

- Propiedad alineación horizontal de elementos Grid: **justify-self**
 - Valores
 - start
 - end
 - center
 - stretch (por defecto)

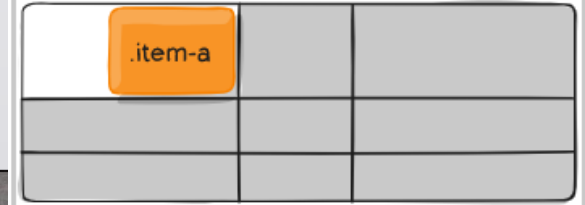
```
.element {  
  justify-self: start;  
}
```



```
.element {  
  justify-self: center;  
}
```



```
.element {  
  justify-self: end;  
}
```



CSS

• 10.-Layout. Grid.

• Propiedad alineación vertical de los elementos Grid: **align-self**

• Valores

- start
- end
- center
- stretch (por defecto)

