

16. Cajas en CSS

Podemos maquetar asumiendo que cualquier elemento es una caja con:

- *content*: contenido *width* y *height* (en px o %) `p { width:90px; height:50px; }`
 - O establecer máximos o mínimos `p { min-width:100px; }`
- *margin*: margen exterior (en px o %) `p { margin: 20px; }`
- *padding*: margen interior (en px o %) `p { padding: 10px; }`
- *border*: línea que bordea el contenido
 - Línea sólida: `p { border-style: solid; }`
 - Línea discontinua: `p { border-style: dashed; }`
 - Línea de puntos: `p { border-style: dotted; }`
 - Línea doble: `p { border-style: double; }`
 - Surco: `p { border-style: groove; }`
 - Cresta: `p { border-style: ridge; }`
 - Relieve hundido: `p { border-style: inset; }`
 - Relieve saliente: `p { border-style: outset; }`
 - Color del borde: `p { border-color: blue; }`
 - Ancho del borde: `p { border-width: 5px; }`
 - Bordes redondeados `p { border-radius: 5px; }`

Podemos alterar los márgenes, bordes y paddings independientemente

<pre>p { border-top-style: dotted; border-right-style: solid; border-bottom-style: dotted; border-left-style: solid; }</pre>	Equivale a: <pre>p { border-style: dotted solid; }</pre>
--------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------

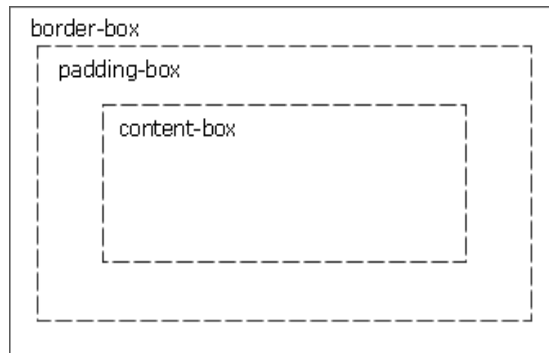
Podemos abreviar agrupando los valores, según esta pauta:

- 4 valores → uno para cada lado en el sentido de las agujas del reloj
- 3 valores → arriba [derecha izquierda] abajo
- 2 valores → [arriba abajo] [derecha izquierda]
- 1 valor → todos iguales

La pauta de agrupación funciona igual para los valores de márgenes y paddings

Formato de cajas

- Sombra `p { box-shadow: 5px 5px 5px red; }`
- Imagen como borde `p { border-image:url(border.png) 30 30 round;}`
- Color de fondo `p { background-color: red; }`
- Imagen de fondo, puede llenar sólo el contenido, hasta el padding o hasta el borde



```
.imagen { background-image: url(fondo.jpg);
          background-repeat: no-repeat;
          background-origin: content-box;
          background-clip: content-box;
          background-size: auto; }
```

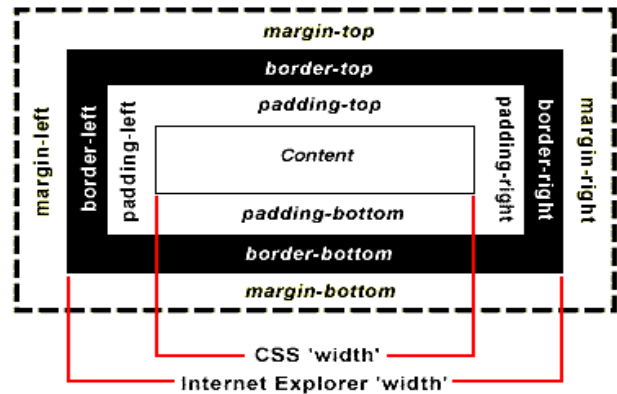
- Degradados lineales
 - Hacia una dirección `#d1 {background: linear-gradient(green, blue); }`
 - Diagonales `#d2 {background: linear-gradient(to left, green, blue); }`
 - Diagonales con grados `#d3 {background: linear-gradient(to top right, green, blue);}`
 - De más de dos colores `#d4 {background: linear-gradient(30deg, green, blue);}`
 - Con transparencia `#d5 {background: linear-gradient(red, yellow, green);}`
 - Con transparencia `#d6 {background: linear-gradient(to left, rgba(255,0,0,0),
rgba(255,0,0,1));}`
- Degradados radiales `#d7 {background: radial-gradient(orange 5%, red 20%,
black 60%);}`

Si detectamos errores de compatibilidad con algún navegador, debemos añadir su prefijo antes del valor que causa el conflicto

-webkit-	para Chrome o Safari
-o-	para Opera
-moz-	para Firefox

¿Cuánto mide una caja?

```
p {
  margin: 10px;
  padding: 10px;
  width: 250px;
  border: 5px solid gray;
  background-color: red;
}
```



El ancho total del elemento será:

$$250\text{px} + (10\text{px} * 2) + (5\text{px} * 2) + (10\text{px} * 2) = 300\text{px}$$

Posición

- Fija
- Relativa, respecto al flujo normal
- Absoluta, se salta el flujo normal
- Superpuestas, el valor más alto encima

```
.footer {position: fixed; }
.c1 {position: relative; top: 20px; }
.c2 {position: absolute; top: 20px; }
.c3 {position: absolute; z-index:2; }
```

Flotación

- Alinear a la izquierda:
- Alinear a la derecha:
- Romper la flotación

```
img { float: left; }
img { float: right; }
div { clear: both; }
```

Visibilidad

- Elemento invisible, sí ocupa espacio:

```
.escondido {visibility: hidden; }
```

Visualización

- Elemento invisible, no ocupa espacio:
- En línea
- De bloque
- Bloque que se agrupa en línea
- Flexible, los elementos se adaptan al espacio

```
.nulo { display: none; }
.linea { display: inline; }
.bloque {display:block; }
.acumulable {display: inline-block; }
.contenedor { display: flex; }
```

Permite más posibilidades de forma más simple:

Propiedades del contenedor *flex*

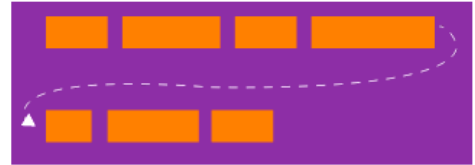
Dirección

`flex-direction: row | row-reverse | column | column-reverse ;`



Ajuste

`flex-wrap: nowrap | wrap | wrap-reverse ;`



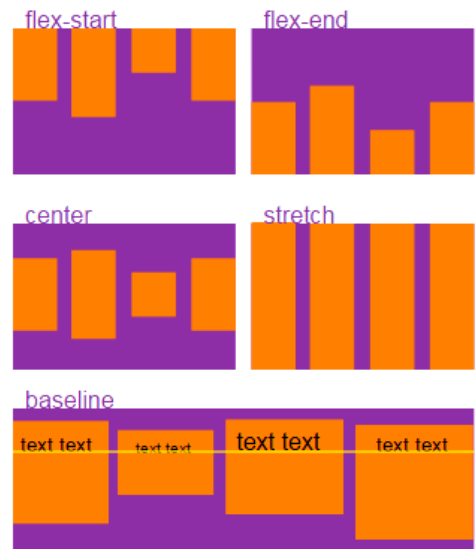
Alinear elementos en horizontal

`justify-content: flex-start | flex-end | center | space-between | space-around ;`



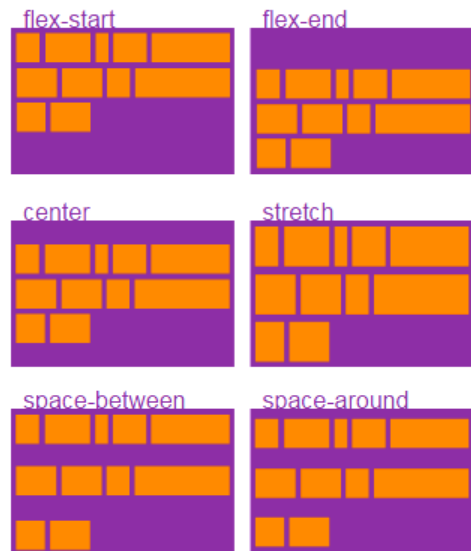
Alinear elementos en vertical

`align-items: flex-start | flex-end | center | baseline | stretch ;`



Alinear filas de elementos

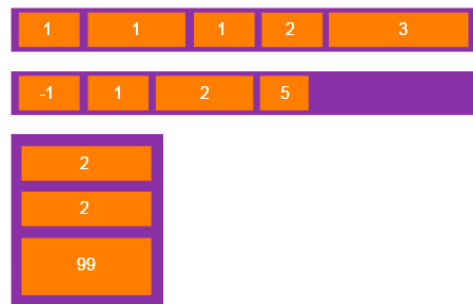
align-content: flex-start | flex-end | center | space-between | space-around | stretch;



Propiedades de los elementos de un contenedor flex

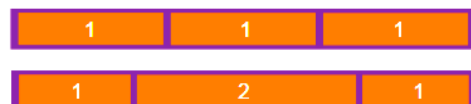
Orden de los elementos

order: <número entero>;



Tamaño

flex: <número entero>;



Alinear elementos individualmente

align-self: auto | flex-start | flex-end | center | baseline | stretch;



<http://css-tricks.com/snippets/css/a-guide-to-flexbox/> (en inglés)