



POSICIONAMIENTO

Universidad de Sevilla
IPO

CSS: POSICIONAMIENTO

Posicionamiento: Alteración del flujo de composición

Principios de diseño:
Simetría – Asimetría



[Netlify IPO: position](#)

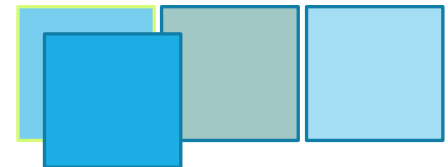
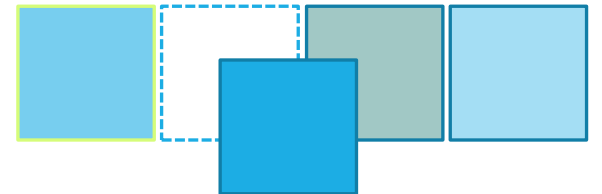
DEFINICIONES

- **Espacio inicial:** el que de partida asigna el flujo CSS a la hora de ubicar al elemento HTML)
- **Espacio final:** donde el elemento HTML se muestra
- **Elemento posicionado:** aquel cuyo espacio final puede ser distinto del inicial

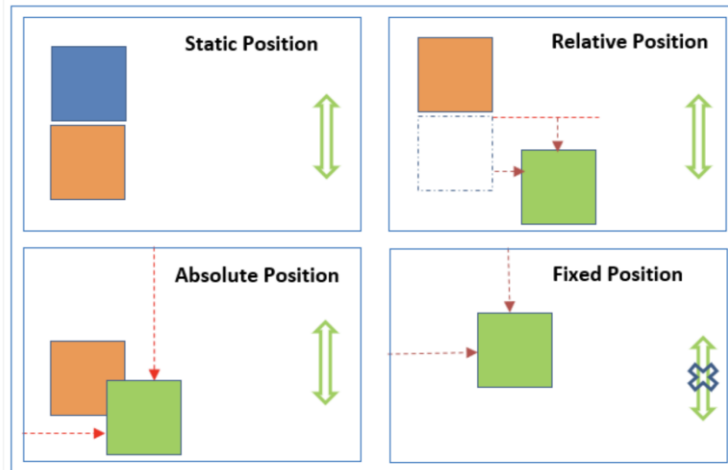
CARACTERIZACIÓN

En los elementos posicionados debemos tener en cuenta:

- **Conservación:** ¿se conservan (muestran) ambos espacios (inicial y final) o sólo el espacio final?
- **Referencia:** ¿respecto a que elemento se realiza el desplazamiento?
- **Visibilidad:** ¿se muestra el elemento aunque el uso de la barra de desplazamiento (scroll) lo oculte?.



CSS: POSITION



```
.ancestro-elemento {  
  position: relative;  
}  
  
.elemento {  
  position: absolute  
}
```

A la hora de posicionar un elemento de forma absoluta es necesario establecer cual será el elemento de referencia respecto del que se mueve. Ese elemento debe ser un elemento posicionado (no estático). Por tanto, aunque no se vaya a desplazar, deberá incluir la propiedad position con un valor distinto de static. Cuando el valor de position es indiferente, es frecuente elegir el valor relative para tal menester.

MDN: [position](#)

position: static | relative | absolute | fixed | sticky

No posicionado

Posicionado

| VALOR | CONSERVACIÓN | REFERENCIA | VISIBILIDAD |
|----------|---------------------------------------|---------------------------------------|--------------------------------------|
| relative | Espacio INICIAL y FINAL | Respecto a su POSICIÓN INICIAL | SI lo oculta el <i>scroll</i> |
| absolute | Solo espacio FINAL | Respecto a su 1º ANCESTRO NO ESTÁTICO | SI lo oculta el <i>scroll</i> |
| fixed | Solo espacio FINAL | Respecto a la PANTALLA | NO lo oculta el <i>scroll</i> |
| sticky | Espacio INICIAL y FINAL | Respecto a la PANTALLA | NO lo oculta el <i>scroll</i> |

sticky Se comporta de forma mixto: pasando de relative a fixed

absolute: Si no existiera ancestro estático explícito, se interpreta que es la pantalla

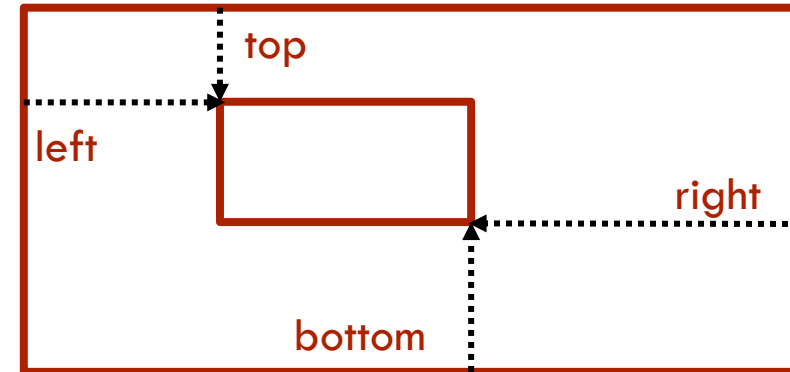
sticky: El elemento se desplaza con la operación de scroll hasta mantenerse en la posición fija (respecto a la pantalla) determinada por el valor top (o bottom).

CSS: TOP, LEFT, RIGHT Y BOTTOM

Propiedades habilitadas para indicar el desplazamiento de los elementos **no estáticos**

top: **auto** | *<longitud>* | *<porcentaje>*
left: **auto** | *<longitud>* | *<porcentaje>*
right: **auto** | *<longitud>* | *<porcentaje>*
bottom: **auto** | *<longitud>* | *<porcentaje>*

CSS-Tricks: [top](#), [right](#), [bottom](#), [left](#)



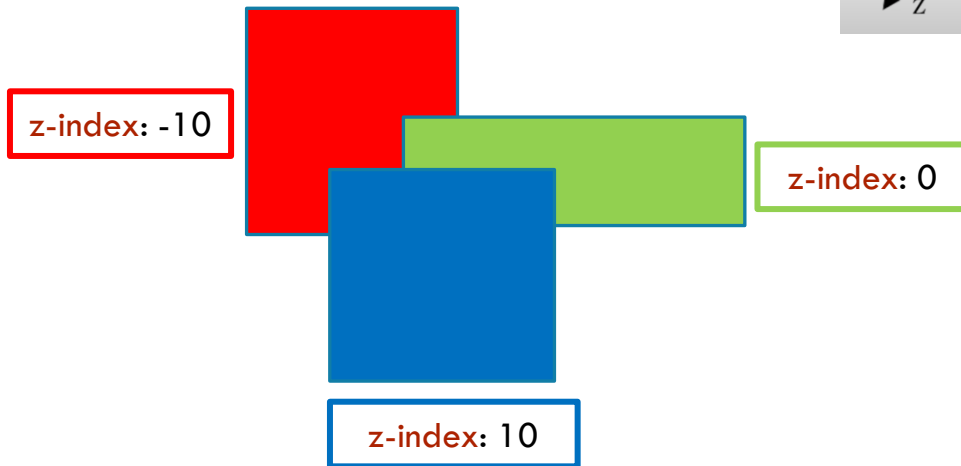
NOTAS:

- Se admiten valores negativos
- Los valores pueden desplazar el elemento a una posición no visible (fuera de la pantalla)
- Las combinaciones más frecuentes son:
 - **top** y **left** (el punto de referencia para el desplazamiento es la esquina superior izquierda)
 - **bottom** y **right** (el punto de referencia para el desplazamiento es la esquina inferior derecha)
 - Ante conflictos, prevalecen los valores de las propiedades **top** y **left**

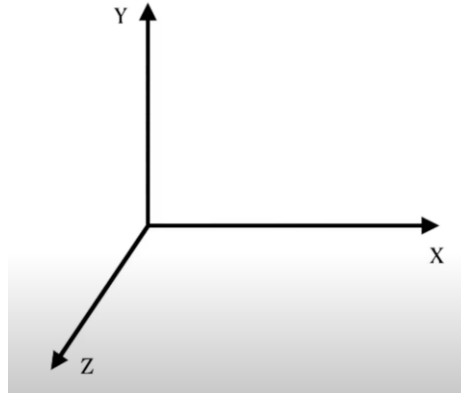
CSS: Z-INDEX

MDN: z-index

z-index: **auto** | *<numero-entero>*



- Convenio respecto a los valores de z-index: evitar valores continuos. Mejor usar 10, 20, 30, ... que 1,2,3, ...
- El valor de z-index puede ocultar el elemento al extremo de no ser visible.



MDN: Stacking context

El posicionamiento origina escenarios en los que existen elementos que ocupan el mismo espacio de dibujo. En ese caso: ¿quién se muestra delante o detrás?

Las capas de dibujo (**stacking context**) simulan una tercera dimensión (profundidad) cuya utilidad es esclarecer los dibujos resultantes cuando hay elementos superpuestos.

De partida, todos los elementos de la página se muestra en una primera capa de dibujo creada por el elemento raíz (**html**). Pero también crean capas de dibujo los:

- Elementos posicionados con **fixed** o **sticky**
- Elementos posicionados con **absolute** o **relative** siempre que el valor de **z-index** sea distinto de **auto**

CONSIDERACIONES

- Las capas se superponen atendiendo al valor numérico de la propiedad z-index. Cuanto menor sea el valor de z-index mayor profundidad
- Los elementos situadas en la misma capa de dibujo se muestran siguiendo el orden de aparición en la página HTML
- El valor de z-index afecta al elemento y a sus descendientes.

CSS: PSEUDOELEMENTOS

Pseudoelementos: Son términos que se añaden como sufijos a los selectores y que permiten dar estilos a ciertas partes de un elemento.

- [MDN: pseudoelements](#)
- Ejemplos ::after ::before ::first-letter ::selection

Pseudoclasses: Son términos que se añaden como sufijos a los selectores para dar estilo a los elementos en un determinado estado o que cumplen una cierta condición

- [MDN: pseudo-classes](#)
- Ejemplos: :hover :not() :nth-of-type() :root

NOTA: Se recomienda usar :: (prefijo con doble dos puntos) en vez de : (prefijo dos puntos) para los pseudoelementos

CSS: ::BEFORE Y ::AFTER

[MDN: ::before](#)

[MDN: ::after](#)

Los pseudoelementos **::before** y **::after** añaden al elemento seleccionado nuevos elementos con el contenido procedente del valor de la propiedad **content** y que son añadidos como primer (resp. último) hijo.

```
<div class="item">Lorem ipsum</div>
```

```
.item::before {  
  content: "<< ";  
}  
  
.item::after {  
  content: " >>";  
}
```

```
<div class="item">  
  ::before  
  "Lorem ipsum"  
  ::after  
</div>
```

<< Lorem ipsum >>

Los pseudoelementos **::before** y **::after** son usados frecuentemente para enriquecer los elementos de HTML con contenido de carácter auxiliar o estético. En muchas ocasiones, el contenido es reubicado con **position**.

Algunos casos típicos de uso son:

- decorar el elemento seleccionado
- diseño de notas emergentes (*tooltips*)

Consideraciones:

- Las reglas **::before** o **::after** deben incluir necesariamente la propiedad **content** (en otro caso, no se crea ningún elemento)
- En el flujo CSS normal, los elementos creados por **::before** y **::after** son en línea
- Si **::before** o **::after** se aplican a un contenedor Flexbox (o Grid), los elementos creados son considerados items Flex (o Grid) ordinarios

CSS: CONTENT

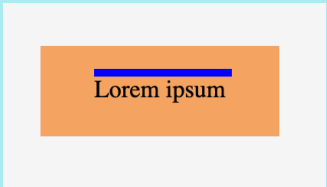
[MDN: content](#)

content: none | **normal** | `<string>` | `<image>` | **attr**(x)

Cadena con contenido textual
Puede ser vacía lo que facilita la decoración gráfica

```
/* Lorem ipsum */
.item {
  position: relative;
}

/* Barra superior azul */
.item::before {
  content: "";
  width: 90px;
  height: 5px;
  background-color: blue;
  position: absolute;
  top: 15px;
  left: 35px;
}
```



Lorem ipsum

Función **attr**()

Permite añadir contenido procedente del valor de un atributo del elemento seleccionado

```
<div class="item" data-marca="** ">
  Lorem ipsum
</div>
```

```
.item::before {
  content: attr(data-marca);
}
```



** Lorem ipsum

A blue L-shaped graphic element consisting of a vertical line on the left and a horizontal line extending to the right, positioned to the left of the 'IPO' text.

IPO

Universidad de Sevilla