



POLITÉCNICO COLOMBIANO  
JAIME ISAZA CADAVID

Educación para  
*vivir mejor*

# Construcción de elementos de software web 1

## Semana 5

Unidades y Posicionamiento

---

Media Técnica en Programación de Sistemas de Información  
Politécnico Jaime Isaza Cadavid - 2020



Politécnico Colombiano Jaime Isaza Cadavid



@PolitecnicoJIC



## Unidades

En la lección anterior vimos los píxeles (**px**), unidades absolutas, cuyo tamaño no varía. Un píxel siempre es un píxel. Veamos algunas unidades más.

### Unidades relativas al viewport

El viewport es la zona visible en una web. Mide **100vw** de ancho y **100vh** de alto siempre, en la pantalla pequeña de un móvil o en la grande de un portátil.

Las unidades **vw** y **vh** nos permiten ajustar ancho y alto de manera relativa al viewport.

```
1 .wrapper {  
2   height: 100vh;  
3 }
```

### Unidades de porcentaje

La unidad de % nos permite ajustar tamaños para que sean un porcentaje del tamaño de su etiqueta madre.

```
1 .hero {  
2   height: 33.3%;  
3 }
```

### Tamaños mínimos y máximos

Con max-width, min-width, max-height y min-height podemos controlar los tamaños de los elemento para que sean fluidos, y se comporten como nosotras queremos frente a diferentes tamaños de ventanas del navegador.

```
1 .box {  
2   width: 100%;  
3   max-width: 400px;  
4   min-width: 300px;  
5 }
```



Existen dos recursos fundamentales para modificar la apariencia de nuestra web: visualización (o display) y posicionamiento. Ambos permiten modificar cómo se muestran los elementos de la página ya sea modificando su tamaño, su posición o ambos a la vez.

Así estos elementos nos sirven para saber las diferentes opciones que tenemos de colocar los elementos según el diseño que nos manden.

Los vamos a utilizar cuando tengamos que darle un aspecto determinado al contenido, es decir, **siempre**. Algunos casos concretos son:

- Un listado de elementos distribuidos por columnas.
- El típico módulo que lleva el corazón de "like" en una esquina.
- Una galería donde las flechas de anterior/siguiente estén, una cada lado, y centradas verticalmente.
- El típico módulo de precios "desde \$ 5.000 al mes" donde el diseñador ha desplegado toda su creatividad compositiva con un diseño con elementos a diferentes tamaños.
- El menú que se mantiene en la parte superior del navegador al hacer *scroll*.
- El menú que aparece por uno de los laterales.

## Visualización (display)

Antes de entrar en detalle con el posicionamiento recordemos los modos de presentación que hemos visto con anterioridad.

La propiedad CSS display, se encarga de definir cómo se va a visualizar un elemento HTML, como va a colocarse en la página y cómo se colocarán el resto de elementos respecto a este. Según el valor que tenga asignado display, un elemento puede ocupar el ancho entero de su contenedor, ocupar solo el espacio que necesite para mostrar su contenido, mostrarse como si fuese una casilla de una tabla o directamente ocultarse.



Los navegadores web aplican por defecto un valor display a todos los elementos HTML de nuestra web. Hay muchos valores distintos para display pero, por el momento, nosotros solo veremos cuatro:

- **Block:** Los elementos en bloque se muestran ocupando el ancho completo de su contenedor. Los elementos en bloque siempre empiezan en una nueva línea y nunca van a tener más elementos a su misma altura dentro del mismo contenedor, estarán más arriba o más abajo. Se les puede aplicar margin, padding, alto y ancho.
- **Inline:** Los elementos en línea o inline son aquellos que ocupan lo que ocupa su contenido. En estos, el tamaño será exactamente el tamaño de su contenido y no podremos asignarle un tamaño diferente, ni tendrá efecto un margin vertical.
- **Inline-block:** Los elementos inline-block ocupan por defecto el ancho de su contenido y se comportan como si se tratase de un elemento en línea, pero permiten tener un ancho, un alto y relleno y márgenes verticales, como sucede con los elementos en bloque.
- **None:** Oculta por completo cualquier elemento al que se lo apliquemos, será como si ese elemento no existiese ya que no se mostrará y el resto de elementos de la página lo ignorarán.

## Posicionamiento

Aparte de modificar esa distribución, podremos hacer que cambien su comportamiento a la hora de hacer scroll en la página y que modifiquen la posición de otros objetos al modificar la suya propia.

La propiedad position de CSS será la que nos permita modificar la forma en la que se distribuyen los objetos a través de una página web.



El atributo position es fundamental en las webs actuales porque sin él todo el contenido sería lineal, es decir, no podríamos tener ciertas superposiciones y no podríamos sacar a un elemento del flujo de la página para que el resto de elementos (contenedor y elementos hermanos) no lo tengan en cuenta y lo ignoren.

La propiedad position tiene cuatro valores posibles:

- **static:** Es el tipo de posición por defecto en todos los elementos HTML.
- **relative:** Permite modificar la posición de un elemento en función de su posición actual en la página.
- **absolute:** Saca al elemento del flujo de la página, es decir, hace que su contenedor y los elementos de antes y después no lo tengan en cuenta a la hora de posicionarse y definir su tamaño y por otro lado posiciona el elemento en función de la posición del body o en su defecto del primer elemento contenedor que tenga una posición diferente a static (posición por defecto).
- **fixed:** Saca a un elemento del flujo normal de la página y permite posicionarlo en función de la ventana del navegador. Aparte, este tipo de elementos mantienen su posición cuando hacemos *scroll* en la página (como si se mantuvieran anclados en un mismo punto), de ahí su nombre fixed (fijo).

Como hemos visto, cuando posicionamos una caja con cualquier valor que no sea static y modificamos su posición horizontal y/o vertical (**top, right, bottom, left**) esta se puede superponer visualmente por encima de otras.

Esto ocurre porque, adicionalmente a sus posiciones horizontales y verticales, las cajas se apilan a lo largo de un "eje z".

Cuando las cajas se superponen por encima de otras, se están posicionando en capas adicionales a la capa normal de renderizado (capa 0).



La posición Z de cada capa representa el orden de apilamiento. Podemos modificarlo con la propiedad **z-index**. Números más grandes significan mayor cercanía a la observadora.

## Transform

Otra forma interesante de modificar la posición de un elemento HTML es la propiedad transform con la que podemos realizar una serie de ajustes al elemento respecto a sí mismo. Sin embargo no saca al elemento del flujo de la página como position: absolute o position: fixed, y el resto de elementos de la página se comporta como si no hubiésemos aplicado una transformación a uno de ellos.

Vamos a ver varias transformaciones:

- Translate
- Scale
- Rotate

## Translate

Permite desplazar el elemento, una cantidad dada, horizontal y/o verticalmente:

```
1 .element {  
2   transform: translate(-100px, 100px);  
3 }
```

## Scale

Permite agrandar o reducir el elemento, proporcionalmente o indicando la deformación horizontal y vertical independientemente:



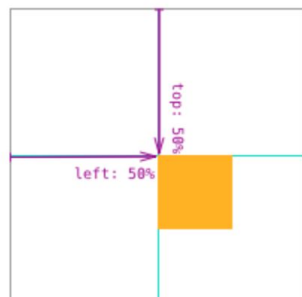
```
1 .element {  
2   transform: scale(2);  
3 }
```

## Rotate

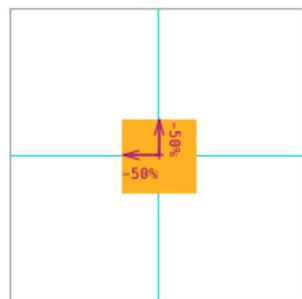
Permite rotar el elemento:

```
1 .element {  
2   transform: rotate(45deg);  
3 }
```

Transform se suele usar mucho junto con posiciones fijas o absolutas para centrar elementos horizontal y verticalmente:



```
.orange {  
  left: 50%;  
  position: absolute;  
  top: 50%;  
}
```



```
.orange {  
  left: 50%;  
  position: absolute;  
  top: 50%;  
  transform: translate(-50%, -50%);  
}
```



## Taller: Ejercicios Práctico

1. **Listas Horizontales:** Crea una lista de cinco elementos que se muestre en línea y con espacios entre cada elemento de 12 píxeles.

- a. Vista de la lista sin estilos

- Home
- Contact
- Product
- About
- More

- b. Vista de la lista con estilos

Home Contact Product About More

2. **Desplazando divs relativamente:** Define un documento HTML con un div padre (divPadre), dentro del cual existan otras 3 cajas contenedoras div (div1, div2 y div3), cada una de ellas con unas dimensiones de 300x300px, 40 píxeles de margin en todas direcciones, 30 píxeles de padding en todas direcciones y un background color diferente. Usando posicionamiento relativo genera un desplazamiento de los div de la siguiente manera:

- El div 1 deberá desplazarse 100 píxeles a la derecha y 50 píxeles hacia abajo respecto a lo que sería su posición normal.
- El div 2 deberá desplazarse 150 píxeles a la izquierda y 320 píxeles hacia arriba respecto a lo que sería su posición normal.
- El div 3 deberá desplazarse 180 píxeles a la derecha y 240 píxeles hacia arriba respecto a lo que sería su posición normal.

3. **Cajas sobre cajas:** Vamos a definir un documento HTML con 3 cajas contenedoras div (div1, div2 y div3):

- La primera con unas dimensiones de 500x500px y color de fondo amarillo.
- La segunda con dimensiones 300x300px y color de fondo verde.





- c. La tercera con dimensiones 150x150px y color de fondo azul.

Usando posicionamiento absoluto establecemos para el div2 y el div3 el mismo origen que para el div1, de modo que las cajas se superpongan y el efecto generado sea ver cuadrado azul sobre uno verde que a su vez está sobre uno amarillo.

Ahora tenemos que hacer que las cajas estén centradas vertical y horizontalmente añadiendo:

- 40px de padding y 2px de borde al div1.
- 75px de padding al div2.
- 20px de borde de puntos al div3.

Para esto te recomendamos usar **box-sizing: border-box;**

4. **Mi artículo:** Crea una composición similar a la de la imagen.





Las dimensiones de esta composición serían las siguientes:

- a. El body tendrá un borde de 8px
- b. El contenido estará centrado dentro del body e irá dentro de un div que tendrá 106px de margen superior
- c. En el primer texto irá el nombre del autor con una fuente de 18px y un margen inferior de 40px
- d. El titular irá después con un tamaño de fuente de 32px y un margen inferior de 32px
- e. Cada párrafo tendrá un tamaño de fuente de 18px y un margen inferior de 27px
- f. El enlace tendrá un padding superior e inferior de 8px y otro izquierdo y derecho de 16px y un margen izquierdo de -16px

**5. Tecnologías web:** La web que vamos a crear consta de las siguientes características:

- Toda la web usa una tipografía sin serifa (sans-serif)
- Tiene como título "Tecnologías web"
- Tiene un párrafo que describe qué son las tecnologías web
- Al final del párrafo, tiene un listado de tecnologías compuesto por: HTML, CSS y JavaScript, cada una de las cuales aparece subrayada para indicar que se puede interactuar
- Al poner el ratón sobre cualquiera de ellas:
  - i. el cursor cambia para indicar que estamos obteniendo ayuda
  - ii. aparece un tooltip (recuadro flotante de 400px por 200px) de color blanco, con el nombre de la tecnología como título del tooltip y una breve descripción de la misma



POLITÉCNICO COLOMBIANO  
JAIME ISAZA CADAVID

Educación para  
*vivir mejor*

## Quiero saber más..

- <https://www.youtube.com/watch?v=13CbCpAnvYI>
- <https://www.youtube.com/watch?v=whqnlupzpNk>
- <https://www.youtube.com/watch?v=X6lG1biGJa8>
- <https://www.youtube.com/watch?v=wpuP2kCN6QE>
- [https://www.youtube.com/watch?v=u2O\\_ys4X1cQ](https://www.youtube.com/watch?v=u2O_ys4X1cQ)
- <https://devdocs.io/css/transform>
- <https://www.youtube.com/watch?v=AoAy4jCFDDw>
- <https://uniwebsidad.com/libros/css/capitulo-5/tipos-de-elementos?from=librosweb>
- <https://uniwebsidad.com/libros/css/capitulo-5/posicionamiento?from=librosweb>
- <https://uniwebsidad.com/libros/css/capitulo-5/posicionamiento-absoluto?from=librosweb>
- <https://uniwebsidad.com/libros/css/capitulo-5/posicionamiento-fijo?from=librosweb>
- <https://uniwebsidad.com/libros/css/capitulo-5/posicionamiento-relativo?from=librosweb>
- <https://uniwebsidad.com/libros/css/capitulo-5/posicionamiento-flotante?from=librosweb>



Politécnico Colombiano Jaime Isaza Cadavid

@PolitecnicoJIC