

# Posicionamiento

## # Posicionamiento Normal

- Cada elemento en bloque aparece en una línea nueva haciendo que el resto del contenido baje como se espera. Este es el comportamiento por default
- Para establecer el posicionamiento de los elementos utilizamos la propiedad **position**
- Si queremos que un elemento se comporte de manera normal no tenemos que agregar ninguna propiedad ya que como dijimos es la forma default
- En caso de tener que establecer que un elemento tiene que volver a tener posición normal utilizamos el valor **static**

```
div {  
  position: static;  
}
```

## Posicionamiento Relativo

- Esta opción hace que un elemento ocupe el lugar que ocuparía en el flow regular (normal) pero nos da la opción de mover su posición con valores de top, left, right y bottom.
- Para establecer que un elemento se posiciona de manera relativa tenemos que setear el valor de la propiedad **position** en **relative**
- Podemos decir que del lugar donde esta el elemento lo podemos mover en distintas direcciones pero siempre relativo al lugar que ocupa.
- El resto de los elementos ocupa su lugar normal
- Los valores top, left, right y bottom aceptan una unidad de medida que puede ser pixeles u otros valores de los vistos

```
p {  
  position: relative;  
  top: 10px;  
  left: 20px;  
}
```

## Posicionamiento absoluto

- Con este posicionamiento podemos establecer la posición absoluta que queremos que ocupe un elemento en relación a su contenedor. Este elemento es removido del flow normal por lo cual no afecta al resto de los elementos. Estos elementos posicionados de manera absoluta son scrolleados acompañando el scroll del usuario.

```
p {  
  position: absolute;  
  top: 10px;  
  left: 20px;  
}
```

## Posicionamiento Fixed

- Este posicionamiento es similar al absoluto pero en lugar de ser posicionado de forma absoluta al elemento padre lo hace en relación al browser, es decir que este elemento se posiciona usando las coordenadas del browser y no del elemento padre. El resto de los elementos siguen funcionando de la misma forma, es decir que hacen como si este elemento fixed no existiera.

```
p {  
  position: fixed;  
  top: 0;  
  left: 0;  
}
```

## z-index

- Al cambiar la forma en que se comportan o posicionan los elementos podemos controlar el nivel de profundidad que utiliza cada uno utilizando la propiedad **z-index**, es decir que si dos elementos estan posicionados en el mismo lugar podemos hacer que uno se vea arriba del otro aumentando el nivel de z-index. Por ejemplo si tengo 2 elementos en la misma posición fixed puedo cambiar el nivel de uno a z-index:1 y el otro a z-index:2 y de esta forma hacer que el que tiene mayor nivel de z-index se vea arriba de todo.

```
p {  
  z-index: 100  
}
```

## Elementos Flotantes

- Podemos establecer que un elemento flote hacia la izquierda o derecha de un elemento, el resto de los elementos se posicionan al costado de dicho elemento.

- Utilizamos la propiedad **float** y los valores **left** (izquierda) o **right** (derecha)
- Al utilizar elementos flotando conviene establecer un **width**

```
div {  
  float: left;  
  width: 200px;  
}
```

- Al tener elementos flotando el resto de los elementos se van a posicionar al costado del elemento flotante
- Si queremos que los elementos vuelvan a tener el flow normal podemos utilizar la propiedad **clear** que acepta los valores **left**, **right** o **both**
- Por medio de esta propiedad establecemos que no queremos tener elementos flotantes a la izquierda, derecha o ninguno de los dos

```
div {  
  clear: left;  
}
```

- En este ejemplo el div no va a tener a nadie flotando a su izquierda y se recupera el flow normal de los elementos
- Podríamos tener 3 secciones que esten flotando para hacer una parte del sitio en 3 columnas

```
div {  
  width: 300px;  
  float: left;  
}
```

- Podemos aprender más en el sitio de <http://es.learnlayout.com>
- Otra buena fuente es [Learn CSS Positioning in Ten Steps \(inglés pero se entiende\)](#)
- Actualmente se piensa el diseño web como si fuera imprenta (o los diarios) y existe el concepto de grilla
  - Podemos ver el sitio de [960.gs](#) que tiene buenos ejemplos
    - Básicamente la página se divide en pequeñas secciones donde el ancho máximo es 960px
    - Tenemos máximo 12 columnas
    - Si queremos usar toda una sección utilizamos las 12 columnas
    - Si queremos utilizar 2 columnas donde una es más grande y la otra más chica podemos elegir una columna de 4 espacios y otra de 8
    - Si queremos hacer una página de 2 lados nada más y ocupar la mitad para cada sección podemos hacer dos columnas de 6 y 6

- En caso de querer hacer una página a 3 columnas podríamos dividirlo en 3 secciones de 4 columnas cada una
- Lo importante es que la sumatoria de columnas de 12 como máximo
- Como vemos se pueden utilizar muchas opciones y la grilla nos abstrae de la forma de crear todo esto
  - Algunos sistemas de grillas son más flexibles
  - Se pueden conseguir grillas de 16 columnas
  - Bootstrap también trae su propio **sistema de grilla**
  - Foundation también: **grillas**

## Ejemplos de grillas:

### Ejemplo de distintas opciones



Distintas opciones

### Ejemplo de sitio



Ejemplo de sitio



Edit this page

Previous

« **Listas y Tablas**

Next

**Media Queries** »

Posicionamiento Normal

Posicionamiento Relativo

Posicionamiento absoluto

Posicionamiento Fixed

z-index

Elementos Flotantes

## Docs

Frontend

## Community

Discord

## Social

Linkedin

Instagram

Facebook

Copyright © 2023 Ada Itw.