

Bootcamp Front End, conceptos básicos: Lectura 6

Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Febrero-2022

Bordes

La sintaxis general para colocar bordes a un contenedor o a un elemento con css es :

```
p{  
    border: 5px solid red;  
}
```

Donde el primer valor indica el ancho del pixel, el estilo es indicado como segundo, y el tercer valor indica el color.

El ancho puede ser indicado como valores absolutos o relativos. **El estilo** puede ser solid o dotted.

Luego se puede colocar bordes redondeados como:

```
p{  
    border: 5px solid red;  
    border-radius: 3px;  
}
```

Márgenes

Los márgenes crean un espacio alrededor del elemento, fuera de sus bordes. Su sintaxis es :

```
p{  
  margin: 25px 50px 75px 100px;  
}
```

Donde los valores son en sentido antihorario.

Border-collapse es la propiedad para fusionar bordes, puede ser **inherit**: como lo define del elemento padre, **separate**: separación de borde, **collapse**: fusión de borde.

Padding

El padding crea espacio alrededor del elemento, dentro de sus bordes definidos.

```
p{  
  padding: 25px 50px 75px 100px;  
}
```

Donde los valores van en sentido antihorario.

Border-box es la propiedad por la que el borde de un elemento queda exacto, sin que influya el padding.

```
div{  
  width: 300px;  
  padding: 25px;  
  box-sizing: border-box;  
}
```

Dimensiones

Las dimensiones de un objeto estarán dadas por sus propiedades de ancho y alto. Su sintaxis se expresa:

```
div{  
    width: 50%;  
    height: 200px ;  
    background-color: powderblue;  
}
```

Se puede fijar un máximo ancho, cuando la ventana es menor al ancho de un contenedor, el navegador añade una barra de scrool. Por lo que al incluir la propiedad **max-width** se maneja mejor para ventanas pequeñas.

```
div{  
    max-width: 500px;  
    height: 200px ;  
    background-color: powderblue;  
}
```


Modelo de caja para CSS

Por ejemplo en :

```
div{  
    width: 300px;  
    padding: 20px ;  
    border: 2px solid black ;  
    margin: 5px;  
}
```

La suma sería :

$$Tamaño_{total} : 300 + 40 + 4 + 10 = 354px$$

Desliegue

El estilo para controlar el despliegue es **Display**. Display tiene como posibilidades 3 tipos de posición básica. **Block** e **inline**.

block: el estilo block hace que el elemento use el ancho entero de su contenedor o padre.

inline: El elemento se ajusta horizontalmente.

none: El elemento no es renderizado en lo absoluto.

```
div{  
  display: inline;  
}
```


Posición

Existen 5 tipos de posición para los objetos del body.

Posición estática: al declarar `position:static;`, el elemento se posiciona de manera normal, por lo que no se puede definir su posición respecto a nada. Seguirá el flujo normal de la página.

Posición relativa: Al declarar `position:relative;`, el objeto toma su posición como origen de coordenadas y sobre esa posición, se puede establecer una ubicación arriba(top), abajo(bottom), a la izquierda(left) y a la derecha(right). Dejará un espacio en blanco donde iba a ir y ningún otro objeto puede tomar esta posición que ha dejado.

Posición ajustada: Al declarar `position:fixed;` El objeto es posicionado de acuerdo a la pantalla, y se quedará en esta posición así se haga scrool en la pantalla.

Posición

Posición absolute: al declarar `position:absolute;` Su eje de coordenadas se fija en relación a su ancestro, y respecto a el se puede mover con libertad sobre todo la pantalla. Puede superponerse a otros elementos y su contenedor ya no lo contiene obligatoriamente.

Posición pegajosa: al declarar `position:sticky;`. Su eje de coordenadas es el lugar normal del elemento. Pero permanece ahí a pesar de hacer scroll.

Ejemplo de sintaxis:

```
div{  
    display:  block;  
    position:  sticky;  
    position:  -webkit-sticky;  
    top:0px;  
}
```

Z-index

Refleja la posición en el eje z de los valores de superposición para distintos elementos superpuestos. Determinará que elemento superpone a que. Su sintaxis:

```
div{  
    position: absolute;  
    left: 20px;  
    top: 0px;  
    z-index:-1;  
}
```

Cuando varios objetos dentro de un contenedor no tienen la posición z index, el último elemento definido en html será el que aparezca arriba.

Textos

Se pueden definir la alineación de textos, su decoración, en el caso se necesite, y algunas propiedades más que veremos.

Alineación: Un texto puede alinearse al centro, es decir, estar centrado, a la izquierda, a la derecha o justificado. Ejemplo de sintaxis:

```
p span.arriba{
  text-align: justify;
  color: 4CAF50;
}
```

También se puede colocar una alineación vertical, con la sintaxis.

```
p.a{
  vertical-align: baseline;
}
```

Las consideraciones para vertical-align son: baseline, text-top, text-bottom, sub, super

Decoración de texto

Para decorar el texto se puede tener: `overline` (sobrerayado), `line-through` (línea cruzada), `underline` (subrayado normal) y `overline underline` (subrayado y sobrerayado), `none` (sin línea).

También se puede especificar el estilo de la línea: `solid`, `double` (doble línea), `dotted` (punteada), `dashed` (con guiones), `wavy` (con ondas).

Por último se puede especificar el ancho del subrayado. Esto se expresa con la siguiente sintaxis:

```
p {  
  text-decoration: underline red double 5px;  
  color: 4CAF50;  
}
```