

Bootcamp Front End, conceptos básicos: Lectura 7

Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Febrero-2022

Float

La función de CSS float especifica cómo debería flotar un elemento en la pantalla.

La propiedad **float** permite posicionar y dar formato a un contenido. Por ejemplo cuando una imagen, se coloca a la izquierda de un texto.

Existen 4 tipos de float. left (El elemento flota a la izquierda de su contenedor), right (Flota a la derecha del contenedor), none (Por defecto), inherit (Flota según el valor de su contenedor.)

```
p img{  
  float: left;  
}
```

Float

La función de CSS float especifica cómo debería flotar un elemento en la pantalla.

La propiedad **float** permite posicionar y dar formato a un contenido. Por ejemplo cuando una imagen, se coloca a la izquierda de un texto.

Existen 4 tipos de float. left (El elemento flota a la izquierda de su contenedor), right (Flota a la derecha del contenedor), none (Por defecto), inherit (Flota según el valor de su contenedor.)

```
p img{  
  float: left;  
}
```

Clear

La propiedad `clear` permite decidir qué sucede con el elemento que sigue luego del que se ha definido como `float`.

Puede tener los siguientes valores: `none`(Por defecto), `left` (Es a la izquierda del elemento flotante), `right`(Se coloca a la derecha), `both`(El elemento flotante queda en el medio), `inherit`(Toma el valor de `clear` del padre).

Overflow: cuando un objeto flota y se desea controlar que no sobrepase los límites de su contenedor, se puede fijar al contenedor con `overflow: auto`. Este contenedor se suele llamar **clearfix**.

inline vs inline-block

Float también permite colocar width a los elementos, al igual que **inline-block**

Esta propiedad permite colocar los atributos de width y height a los elementos, que con inline, simplemente no podían tener atributos de ancho y alto.

```
p{  
    display:  inline-block;  
    width:    100px;  
    height:   100px;  
}
```

inline vs inline-block

Float también permite colocar width a los elementos, al igual que **inline-block**

Esta propiedad permite colocar los atributos de width y height a los elementos, que con inline, simplemente no podían tener atributos de ancho y alto.

```
p{  
    display:  inline-block;  
    width:    100px;  
    height:   100px;  
}
```

Alineamiento

Para centrar un elemento en bloque de manera horizontal, como por ejemplo un `div`, se puede usar `margin: auto`. Es recomendable en este caso tener el ancho del elemento definido, para evitar errores.

```
div{  
  margin: auto;  
  width: 50%;  
  padding: 10px;  
}
```

Alineamiento

Para centrar un elemento en bloque de manera horizontal, como por ejemplo un `div`, se puede usar `margin: auto`. Es recomendable en este caso tener el ancho del elemento definido, para evitar errores.

```
div{  
  margin:  auto;  
  width:    50%;  
  padding:  10px;  
}
```


Alineamiento

Si lo que se desea es centrar un texto dentro de un elemento, se usa la propiedad `text-align: center`

```
div{  
  text-align: center;  
  border: 3px solid green;  
}
```

Alineamiento

Si lo que se desea es centrar un texto dentro de un elemento, se usa la propiedad `text-align: center`

```
div{  
  text-align: center;  
  border: 3px solid green;  
}
```

Centrado de imágenes

Para centrar una imagen, por ejemplo, basta convertirla en un objeto **bloque**, y darle márgenes a la izquierda y a la derecha como **auto**

```
img{  
    display: block;;  
    margin-left: auto;  
    margin-right: auto;  
    width: 40%;  
}
```

Centrado de imágenes

Para centrar una imagen, por ejemplo, basta convertirla en un objeto **bloque**, y darle márgenes a la izquierda y a la derecha como **auto**

```
img{  
  display: block;;  
  margin-left: auto;  
  margin-right: auto;  
  width: 40%;  
}
```

Pseudoclasses

Una pseudoclase sucede en un estado especial del elemento: cuando se pasa el mouse, cuando un link es visitado, cuando un elemento es presionado. La sintaxis para las pseudoclasses es:

```
a:hover{  
  background-color:red  
  width: 40%;  
}
```

Pseudoclasses

Una pseudoclase sucede en un estado especial del elemento: cuando se pasa el mouse, cuando un link es visitado, cuando un elemento es presionado. La sintaxis para las pseudoclasses es:

```
a:hover{  
  background-color:red  
  width: 40%;  
}
```

Las pseudoclasses pueden ser: link, visited, hover y active.

Pseudoelementos

Se utilizan para aplicar estilos a una parte específica de un objeto.

Por ejemplo se usan en: Aplicar estilos a la primera letra, o en la primera línea de un elemento.

Insertar contenido antes o después del contenido de un elemento.

```
p::first-line{  
    color:  red  
}
```

Pseudoelementos

Se utilizan para aplicar estilos a una parte específica de un objeto.

Por ejemplo se usan en: Aplicar estilos a la primera letra, o en la primera línea de un elemento.

Insertar contenido antes o después del contenido de un elemento.

```
p::first-line{  
    color:  red  
}
```

Para **first-line** se aplica: propiedades de fuente, color, fondo, espacio entre palabras, etc.

Para **first-letter** se aplican las mismas propiedades que para first-line.

Pseudoelementos

También se pueden agregar pseudoelementos **after** y **before** para agregar contenido antes o después del elemento.

```
h1::before{  
  content url(img.png)  
}
```

Pseudoelementos

También se pueden agregar pseudoelementos **after** y **before** para agregar contenido antes o después del elemento.

```
h1::before{  
  content url(img.png)  
}
```

También está el pseudoelemento **selection** que da propiedades de color y background cuando se selecciona un elemento.

Flexbox

Para tener flexbox primero hay que definir un contenedor como tal.

```
div.flex-container{  
  display: flex;  
}
```

Flexbox

Para tener flexbox primero hay que definir un contenedor como tal.

```
div.flex-container{  
  display: flex;  
}
```

Las propiedades que se pueden manejar son: flex-direction(row o column), flex-wrap(wrap o nowrap), flex-flow (row wrap), justify-content(center,flex-start,flex-end,space-around,space-between), align-items(center,flex-start,flex-end,stretch,space-between)

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;  
}
```