

# 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;  
}
```

# 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;  
}
```

# 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;  
}
```

# 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%;  
}
```

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  
}
```

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)  
}
```

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;  
}
```

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)

# Media Query y responsive

El comando `@media` hace que se puedan ejecutar diferentes estilos para diferentes tipos de dispositivos.

Los comandos **de consulta para dispositivos** son útiles para :  
Conocer el ancho y el alto del dispositivo, ancho y alto de la vista del cliente, la orientación y la resolución.

La sintaxis general para colocar estilos CSS tomando en cuenta alguna de estas consultas es:

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

También se pueden colocar distintas hojas de estilo en el head del DOM con la sintaxis:

```
<link rel= "stylesheet" media="mediatype  
and|not|only (expressions)" href="estilosmobile2.css"
```

# Tipos de media

El **mediatype** se puede especificar como: all, print (para impresoras), screen (para pantallas) y speech (para páginas de lectura.)

El más usado es el mediatype screen. Una sintáxis general para estilos en un determinado tamaño de pantalla será:

```
@media screen and (min-width: 480px) {  
  body{  
    background-color: lightgreen;  
  }  
}
```

# Estilos de fuente e íconos especiales

Algunos estilos de fuente se pueden indexar a la página exportándolos desde google chrome.

`https://fonts.google.com/` dentro podemos encontrar variedad de fuentes que podemos exportar; primero copiando los enlaces al head y colocándo los estilos de fuente.

También se pueden obtener íconos variados del repositorio gratuito en Font Awesome(`https://fontawesome.com/`). Tienes que registrarte y puedes empezar a usarlo.