

Posicionamiento y Layout (Parte II)

Elementos flotantes

Competencias

- Conocer una alternativa para alinear elementos en html.

La propiedad float nos permite sacar un elemento del flujo normal y llevarlo a la izquierda o derecha y hace que el resto de elemento se posicione al lado de este, lo que nos permitirá posicionar mejor nuestro contenido, revisemos el siguiente ejemplo:



The screenshot shows the Chrome DevTools Inspector. The HTML pane displays the following code structure:

```
<!DOCTYPE html>
<html lang="en">
  <head> ...
  </head>
  <body>
    <div class="container">
      
      <div class="texto"> ...
    </div>
  </body>
</html>
```

The CSS pane shows the styles for the `.img` element:

```
html > body > div.container > img.img
Rules Layout Computed Changes Fonts Animations
Filter Styles :hover .cls + inline
element { }
```

Imagen 1. Ejemplo de texto e imagen a la cual se le aplicará la propiedad float.

Tenemos un caja que contiene una imagen con texto, estos por defecto se despliegan como se ve en la imagen anterior (en bloque, según la estructura de nuestro html), ahora con float haremos que quede la imagen a la misma altura que nuestro texto, para eso aplicaremos un float right nuestra imagen.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



```

<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body>
    <div class="container">
      
      <div class="text">...</div>
    </div>
  </body>
</html>

```

html > body > div.container > img.float-right

Rules Layout Computed Changes Fonts Animations

Filter Styles :hov .cls + inline

element { } .float-right { float: right; }

Imagen 2. Resultado al aplicar float:right a la imagen.

Al dar float:right la imagen sale del flujo normal del html y se posiciona lo más a la derecha posible, obtuvimos el resultado esperado pero ¿Por qué al dar float a la imagen el fondo disminuyó y sólo cubre el tamaño del texto? Esto sucede por que la imagen al estar flotante (con la propiedad float) sale del flujo normal del html, entonces para el padre la imagen ya no existe por lo que el ancho y alto se lo está dando el div que contiene el texto.

¿Qué es la propiedad clear?

`clear` nos permitirá limpiar un elemento que se encuentra afectado por un float. Las opciones que tendremos para limpiar dependerán del lugar en el cual se encuentre el elemento float. Los valores que podremos usar son left, right o both. Para comprender mejor que hace la propiedad clear, revisemos el mismo ejemplo anterior, pero con una imagen un poco más grande y agregando otro bloque de texto.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```

<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body>
    <div class="container">
      
      <div class="text">...</div>
    </div>
  </body>
</html>

```

html > body > div.container > div.text

Rules Layout Computed Changes Fonts Animations

Filter Styles :hov .cls + inline

element { }

Imagen 3. Ejemplo de elementos afectados por la propiedad float.

Ahora si le damos clear:right al segundo bloque de texto, podemos ver que este volverá a su posición original sin ser afectado por el elemento con la propiedad float:right.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Inspector Console Network Debugger Style Editor ... X

Search HTML

```
<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body>
    <div class="container">
      
      <div class="texto" style="clear: right;"></div>
    </div>
  </body>
</html>
```

html > body > div.container > div.texto

Rules Layout Computed Changes Fonts Animations

Filter Styles

```
element { :hov .cls + inline
  clear: right;
```

Imagen 4. Resultado al aplicar clear:right al segundo texto.

Layout estático v/s fluído

Competencias

- Conocer los diferentes layout que existen.
- Entender cómo transformar un layout estático en fluido.

Aunque parezca que hayamos terminado nuestra maqueta y que esta se vea genial, aún nos falta resolver un tema muy importante a la hora crear maquetas.

Desde hace un buen tiempo hasta ahora el uso de dispositivos móviles ha aumentado al nivel de superar a los computadores personales haciendo que los tamaños de pantalla sean variados.

Por ejemplo, el contenido de un sitio dispuesto en un smartphone se verá diferente al verlo en un computador.



Imagen 5. Sitio web visto desde un computador.



Imagen 6. El mismo sitio web visto desde un dispositivo móvil.

Así mismo las resoluciones de pantalla también conllevan un problema a la hora de definir una resolución exacta, ya que existen casos en que las pantallas de dispositivos móviles tienen mayor resolución que dispositivos de escritorio.

Esto en concreto crea un gran problema a la hora de decidir cuál será el tamaño adecuado para el diseño de la maqueta, sin embargo existen estrategias que nos ayudarán a construir maquetas que se adegúen a estas diferencias de tamaño y resoluciones.

Estas estrategias hacen relación al tipo de unidad de medida que contiene el layout, o sea, si creamos un diseño con unidades de medida absolutas como los píxeles obtendremos un diseño estático que se mantendrá fijo y si usamos unidades de medidas relativas como los porcentajes, obtendremos cajas que serán relativas al tamaño del viewport del navegador, de modo que es importante identificar el tipo de diseño que usaremos.

Los tipos de diseño más básicos que podremos crear son los:

Diseños estáticos

En este tipo de diseño los elementos que lo componen tienen un ancho fijo que no varía y se mantiene fijo aún cuando reduzca o aumente el tamaño del viewport.

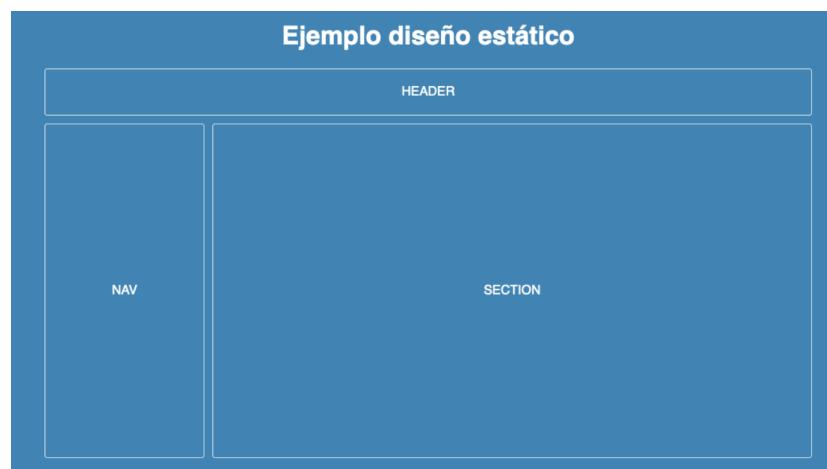


Imagen 7. Ejemplo diseño estático visto desde un computador.

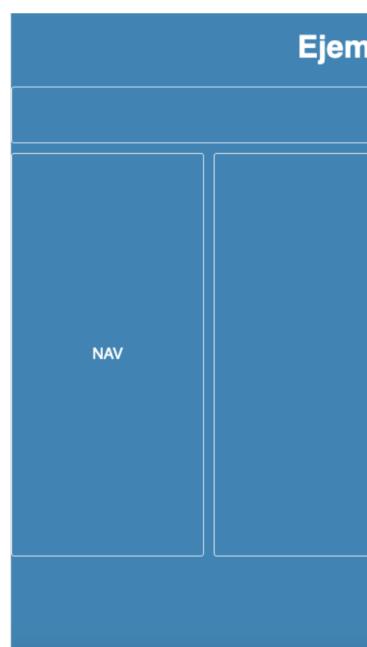


Imagen 8. Mismo ejemplo visto desde un dispositivo móvil.

Por lo general este tipo de diseño está construido pensando en un tamaño específico. Sin ir más lejos, antes de que aparecieran los smartphone los diseños de los sitios web eran totalmente estáticos, definidos por un ancho especificado en el .

Ventajas de los diseños estáticos

Las ventajas que tiene este tipo de diseño son:

- El control del tamaño y posición de los elementos es más preciso.
 - Ya que sabremos realmente la cantidad de píxeles tendrán.
- El control de la cantidad de espacio usado por el contenido.
 - Al igual que el punto anterior, tendremos un control preciso de cuánto espacio existe entre un elemento y otro.
- El control del tamaño de las imágenes.
 - Bajo el mismo contexto, sabremos cual es el tamaño exacto de las imágenes.

Desventajas de los diseños estáticos

Las desventajas que tienen este tipo de diseño son:

- Problemas si el usuario tiene un dispositivo con mayor resolución a la definida por el maquetador.
 - Si ocurre el usuario verá el contenido pequeño haciendo que la usabilidad del sitio baje.
- Problemas si el usuario aumenta el tamaño de la fuente usando el navegador.
 - Al usar unidades estáticas el texto se mantendrá fijo, haciendo que estos no puedan ser modificados por el usuario, bajando la funcionalidad de la página.
- La página será totalmente usable sólo si el usuario navega en un dispositivo con el mismo tamaño al definido por el maquetador.
 - Si por el contrario el dispositivo del usuario tiene una resolución mayor o menor a la definida existirán problemas en la legibilidad del contenido.

Diseños Fluidos

Los diseños fluidos o líquidos son aquellos que usan unidades de medidas relativas como lo porcentajes para estirarse o contraerse dependiendo del tamaño del viewport.



Imagen 9. Ejemplo diseño fluido visto desde un computador.

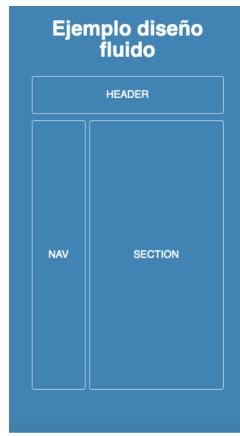


Imagen 10. Mismo ejemplo visto desde un dispositivo móvil.

Ventajas de los diseños fluidos

Algunas ventajas de los diseños fluidos son:

- El contenido del diseño se ajustará al tamaño del viewport.
 - O sea, que el contenido se contraerá si la pantalla es de un dispositivo móvil y no se agregará un scroll horizontal como si lo hacen los diseños estáticos.
- El diseño puede ser configurable por el usuario mejorando la usabilidad del sitio.
 - Esto en la práctica tiene que ver con el tamaño de la fuente.

Desventajas de los diseños fluidos

Algunas desventajas de los diseños fluidos son:

- Si el usuario tiene pantalla ancha el texto se verá largo.
 - Esto provocará que el texto sea poco legible.
- Si tenemos una imagen con tamaño fijo esta se desbordará del diseño provocando que la diagramación no funcione.

El entender cómo funcionan estos dos tipos de diseños hará que la elección de las unidades sea base al tipo de layout que requiere la representación visual. Esto a grandes rasgos nos permitirá crear mejores elementos o cajas con tamaños absolutos estos no fluirán en base al viewport.

Como vemos el trabajar con uno u otro diseño conlleva entender cómo funcionan estos, así como también entender cuales son sus ventajas y desventajas a modo de usarlos de manera adecuada en cualquier tipo de maquetas.

¿Cómo pasar de un layout estático a fluido?

Por la definición de ambos layout la gran diferencia es la unidad de medida que se usa entre uno y otro, por lo que para poder hacer la transformación de un layout estatico a fluido lo que debemos hacer es por un lado migrar las unidades de medida estaticas a relativas donde corresponda, ya que no es necesario hacerlo en todos los lugares, sólo debemos enfocarnos en los más relevantes para el diseño.