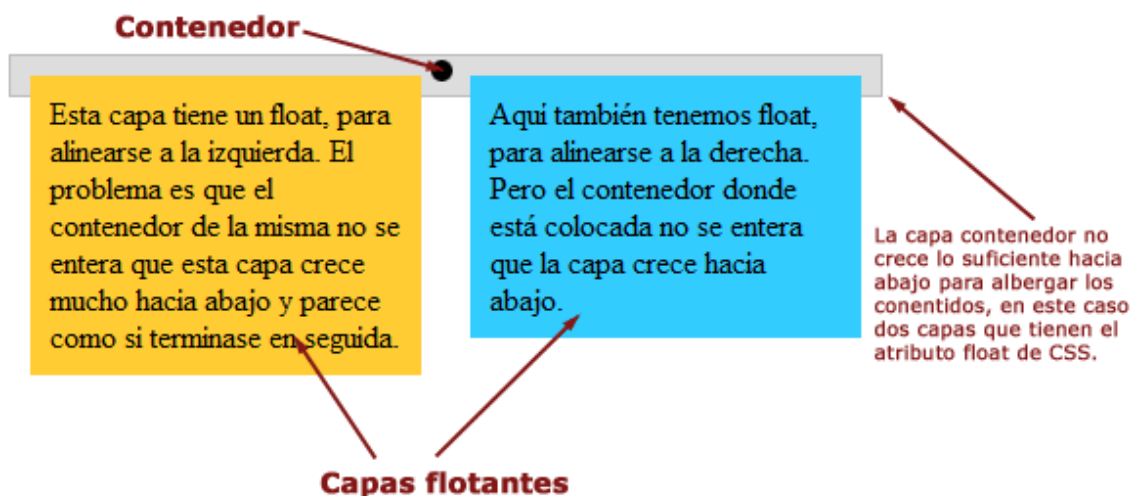


Buenas

Hoy vengo a preguntar una serie de cuestiones que me lleva trayendo de cabeza desde que empecé a ver el tema 4 de diseño y que va a terminar haciendo que me líe para los exámenes de lenguajes de marcas y de diseño. Son dudas un poco “offtopic” del tema, pero relacionadas. Solo que se sale del tipo de dudas que uno podría tener normalmente.

En el tema 2 de Lenguajes de marcas vimos el sistema de maquetación Flex (basado en el Flexbox o caja Flex) como maquetar una página web entera con Flex. Y vimos que maquetar era muy fácil de hacer.

Pero ahora cuando hemos llegado al tema 4 de diseño, hemos visto otras formas de maquetar, grid y con float, y los problemas que acarrea, como el que se comenta [aquí](#) sobre esta situación tan común:



Cuando esta misma situación, con flex no ocurriría, porque tu puedes posicionar los elementos con justify-content, poniendo space-between, o en su defecto, mediante una etiqueta entre medias de ambas capas, hacerle un flex-grow: 1 para que crezca el máximo espacio disponible entre ambas capas. Por lo que realmente, no haría falta usar float.

**Por otro lado:** usar float: left para posicionar un elemento detrás de otro con display: inline... ¿Para qué hacerlo? Si con aplicar al contenedor que contiene a todos esos elementos la propiedad flex-row y flex-wrap, consigues el mismo efecto, sin tener que tocar cada uno de los elementos individualmente. En resumidas cuentas, se ahorra tiempo de codificación.

**Otra cosa que no entiendo:** si no es porque tengamos claro que queremos usar tamaños fijos (por ejemplo, tal vez, se me ocurre, para labels dentro de un formulario, para que los textboxes queden alineados), ¿por qué usamos width todo el rato para aplicar tamaños? ¿No sería mejor usar paddings/margins para rellenar espacios? En lugar de comerse la cabeza haciendo los cálculos para que width + padding dé lo que queremos que dé, o usar este hack que quizás no queramos aplicar a todos los elementos:

```
* {  
  box-sizing: border-box;  
}
```

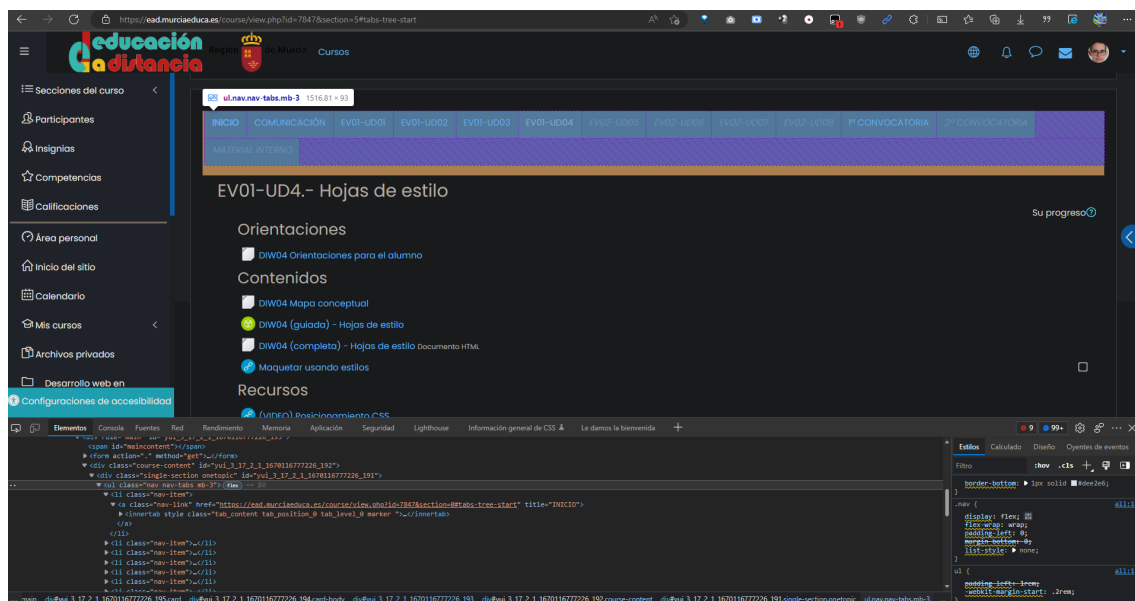
**Por ejemplo:** ¿para qué hacer todo eso de las clases para 12 columnas grid responsive, si puedes conseguir una web igual de responsive con menos de la mitad de tiempo invertido

usando la propiedad flex-wrap que proporciona el sistema de Flexbox y ocupándote de una única dimensión en todo momento (la vertical, o la horizontal) en función del contenedor en el que te encuentres?

Además, con grid estás condicionado a tener un tamaño de wrapper fijo por usar width, mientras que con Flex, puedes aplicar márgenes a izquierda y derecha y obtener un diseño adaptativo al ancho de la pantalla desde el propio wrapper.

Según el profesor de lenguajes de marcas que tenemos (Jose Luis Cruz), Flex es ya el presente, y además que era su sistema preferido de posicionamiento con CSS, entonces, no entiendo como en el tema 4, no se le da continuidad a lo aprendido en lenguajes de marcas. Son dos filosofías de maquetar webs muy diferentes, pero a pesar de que ambas permiten (así a priori, con mis escasos conocimientos en materia por estar aún aprendiendo) hacer las mismas cosas, Flex te permite ahorrar muchísimo tiempo, entiendo que esta manera de maquetar que se enseña en el tema 4 de diseño, se sigue enseñando porque en las empresas Flex no ha tenido buena tasa de adopción y aún se continúa usando el sistema de posicionamiento basado en grid (rejilla). Pero tampoco me cuadra. Y expongo mis motivos:

En la propia plataforma de educación a distancia, si nos metemos en su código, vemos que sí que usa Flex, por ejemplo, para el sistema de navegación por los distintos temas de un curso de Moodle:



Esto mismo, podrían haberlo hecho aplicando float: left a cada item del .nav. Pero es más fácil aplicarle al contenedor una propiedad flex-row y flex-wrap, ya que los datos de ese nav supongo que vienen escritos en la página mediante programación del lado del servidor, extrayéndolos de una BB.DD, con lo que resulta mucho más cómodo hacerlo así con Flex.

Además, también nos dijo Jose Luis que nos enseñaba Flex y su “manera” de crear clases comunes para aplicar a las clases en función de como queríamos que creciesen las “cajas” div, como método de posicionamiento porque es lo que usan tecnologías como Bootstrap. ¿Entonces? ¿Por qué enseñarnos grid como única forma de maquetar en este tema 4, si al final lo que se suele ver que demandan las empresas en las ofertas de trabajo es el manejo de Bootstrap?

En fin, espero que me puedas resolver estas dudas porque la verdad, estoy hecho un lío y no sé qué pensar... ¿La forma correcta de maquetar es Flex, o Grid? ¿O la mezcla de las dos (¿se podría?)?