PEC03-HTML CSS

Daniel Marcos Cuesta

# Preguntas Teóricas

## Document flow

Se trata del comportamiento de bloque por defecto de cualquier página, en la que no se hayan aplicado CSS que modifiquen este. Bajo estas circunstancias, los elementos inline aparecerán en una dirección lineal, y los elementos block aparecerán uno encima de otro verticalmente, comenzando en la parte superior de su contenedor.

## Posiciones static, relative, absolute, fixed y sticky.

static. Posición por defecto, no se encuentra posicionada de ninguna forma concreta sino simplemente como dictamine el document Flow.

relative. Permite posicionar el elemento afectado en función a su posición normal, aplicando las propiedades top, bottom, right y left para crear estas separaciones.

absolute. Nos permite posicionar con respecto al padre mas cercano en el documento, en caso de no tener padre se posicionará con respecto al document body.

fixed. Este posicionamiento nos permite posicionar de forma relativa al viewport, manteniendo su posición incluso al hacer scroll y pudiendo proporcionar con respecto a este las propiedades top, bottom, right y left.

sticky. En este caso el posicionamiento se realizará en función al scroll, manteniendo una posición inicial relative, pero quedando fixed al ejecutar scroll.

<https://codepen.io/dmarcoscu/pen/OJxozRv>

## Float y flexbox

Utilizar flex nos permite posicionar todos los elementos de un bloque únicamente aplicando propiedades al padre, con float esto se complica teniendo que posicionar cada elemento.

Utilizar flexbox nos permite variar entre orientación vertical u horizontal fácilmente, para hacer esto mismo con float necesitaríamos varias reglas.

Utilizando flexbox podemos controlar comportamientos al cambiar el ancho o alto de la pantalla, utilizando flex-wrap por ejemplo entre otros, incluso sin necesidad de media-queries, necesarias sin embargo con float.

<https://codepen.io/dmarcoscu/pen/NWaLXYo>

## Diseño responsivo

Con la aparición de distintos dispositivos con distintos anchos altos, o incluso dispositivos que cambian sus dimensiones en caliente, surge este concepto que trata de solventar el problema de que nuestro diseño se adapte de forma efectiva a las distintas dimensiones posibles de la forma más sencilla posible. Además todo esto debe hacerse teniendo en cuenta el browser support, que no siempre avanza a la velocidad que nos gustaría.

Aprovecho para utilizar una web que siempre me ha parecido muy llamativa, y mostrar un “problema” muy habitual, que es el menú de navegación.

<https://massivemonster.co/>

Como podemos ver el menú desktop tiene muchas opciones que en una versión mobile serían problemáticas, y una de las soluciones más habituales es convertirlo en la famosa “hamburguesa” que muestra un desplegable como en la página de ejemplo.

El segundo ejemplo que muestro es un cambio en la dirección del layout, o una redistribución de este, esto suele consistir en cambiar la dirección a vertical a medida que el dispositivo tiene menor anchura.

<https://github.com/dmarcosc>

En el propio github podemos ver este comportamiento (además del del ejemplo 1) donde tenemos este salto aproximadamente a partir de los 765px de anchura en adelante.

## Media-queries

Las media-queries son reglas CSS que se aplican condicionalmente, es decir, dada una condición se aplicarán unas reglas u otras, esto es muy potente cuando hablamos de diseño responsive porque nos permite aplicar reglas en función el tamaño del dispositivo, un ejemplo es el caso anterior de github donde se cambia la dirección del layout en función del ancho del viewport.

## Ejemplos media queries

@media and screen (min-width:400px) and (max-width: 580px) and (orientation: landscape) {

}

@media and screen (min-width:680px) or (orientation: portrait) {

}