Resumen y datos importantes sobre lectura de CSS y sus unidades:

Un grupo de unidades que funcionan en CSS son:

- Unidades absolutas: Son medidas fijas, no cambian jamás. Se recomiendan solo para archivos con objetivo de ser impreso (Evitar utilizarlas para archivos que serán publicados en la web).

Unidad	Significado	Medida aproximada
in	Pulgadas	1in = 25.4mm
cm	Centímetros	1cm = 10mm
pc	Picas	1pc = 4.23mm
mm	Milímetros	1mm = 1mm
pt	Puntos	1pt = 0.35mm

- Unidades relativas: Son más comunes en CSS, su uso es ideal para trabajar en dispositivos con diferentes tamaños, puesto que estas unidades son muy flexibles.

  La unidad em hace referencia al tamaño actual de la fuente del elemento en cuestión, se puede jugar con una constante para variar su tamaño. Por otro lado, la unidad ex es aproximadamente la mitad de la unidad em (En la práctica ex no se suele utilizar).

Unidad	Significado	Medida aproximada
em	«M»	1em = tamaño fuente
ex	«X» (~0.5em)	1ex = ~tamaño fuente/2
rem	«root M»	1rem = tamaño fuente general
ch	«zero width»	1ch = ancho del cero
~~px~~	~~Pixel~~	~~1px = 1 pixel~~
%	Porcentaje	Relativa a herencia

- Unidades flexibles de viewport (ventana gráfica): estas unidades dependen de la ventana gráfica, se trabaja con el porcentaje del tamaño específico que tengamos en la ventana del navegador, independientemente si se redimensiona o no.
  La unidad vw hace referencia al ancho del viewport, mientras que vh hace referencia al alto. Por ejemplo, si utilizamos 100vw estaremos haciendo referencia al 100% del ancho del navegador, o sea, todo lo que se está viendo de ancho en pantalla, mientras que si indicamos 50vw estaremos haciendo referencia a la mitad del ancho del navegador.

Unidad	Significado	Medida aproximada
vw	viewport width	1vw = 1% ancho de navegador
vh	viewport height	1vh = 1% alto de navegador
vmin	viewport minimum	1vmin = 1% de alto o ancho (el mínimo)
vmax	viewport maximum	1vmax = 1% de alto o ancho (el máximo)

→El uso de pixeles no es una buena medida para el diseño de páginas web.

Recomiendan el uso de:

- Porcentajes: Si trabajamos con ellos en la estilización de la página web está se podrá adaptar a los distintos tamaños pues siempre usará el ancho total visible. Se recomienda hacer uso de un máximo de ancho (este sí en pixeles), a su vez podemos aplicarlo en elementos internos del HTML para que los objetos sean adaptables a sus contenedores.
- Em: Su unidad es equivalente a 16px, es escalable y siempre depende de su elemento padre. Es recomendable usar la unidad de medida em para definir los tamaños de fuente, los altos de línea y también para elementos de diseño que no requieran ser muy exactos o que requieran una medida que tenga relación con el tamaño del texto, como por ejemplo el margen entre párrafos, etc.
- Rem: La unidad de medida rem es muy similar a em, con la única diferencia de que no es escalable, esto quiere decir que no depende del elemento padre, sino del elemento raíz del documento, el elemento HTML. Rem significa “Root Em “, o sea, es un “em” basado en la raíz.

Selectores:

<introducción teórica>

http://flukeout.github.io/

CSS GRID LAYOUT O CSS GRID:

Es una técnica que facilita la colocación de elementos en el documento HTML, es decir, enfocado al diseño de la página, se pueden diversificar la manera de presentación de los objetos, desde posicionamientos simétricos y otros asimétricos.

Algunas de sus ventajas:

- Para diseños 2D (filas y columnas).
- CSS Grid es perfecto para construir una imagen más grande. Hace que sea realmente fácil de manejar el diseño de la página e incluso puede manejar diseños poco ortodoxos y asimétricos.
- Requiere menos intervención de consulta, lo que hace que sea realmente potente en funcionalidades del estilo auto layout: minmax(), repeat(), y auto-fill.

Vienen a sustituir las tablas en CSS.

UNIDAD FR (de fracción): indica el espacio disponible que tengo, lo que sobre, lo que quede.

Divide mi espacio disponible entre el número que le anteponga, basándose en la sumatoria total de elementos que tenga y usen FR, por ejemplo esto lo puedo utilizar en los TEMPLATE-ROWS / COLUMNS, y me estaría refiriendo al espacio de tales bloques disponibles…

De igual, al jugar con un porcentaje (tamaños) de elementos, serán porcentajes calculados del papá, es decir, son heredados.

No olvidar el uso de OBJECT FIT, para los elementos (sobre todo si son imágenes).

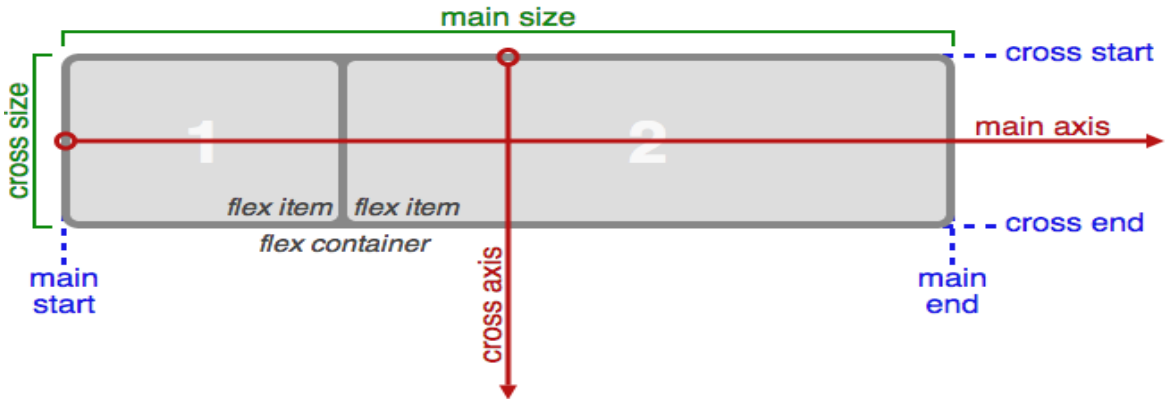
GRID COLUMN – aplicado al elemento en específico (es decir, el hijo del contenedor GRID) me permitirá marcar un inicio y final a cubrir por tal elemento, basándose en la teoría de líneas que forman las casillas; haciendo uso de los números puros, si hago uso de SPAN, irá contando no las líneas, sino las columnas.

¿Propiedades AUTOFIT Y MINMAX? ME AYUDAN AHORRARME LA MEDIAQUERY, pues con el uso de REPEAT, le diríamos que calcule cuantas pueda caber y que se auto ajusta, y lo hará a partir del MIN MAX, que no puede tener menos de un número en específico y luego el ancho que sea 1FR (lo que se pueda).

GRID AUTOFLOW sería como la dirección en la que se irán formando los elementos de la malla, puede ser en fila (ROW) o en columna (COLUMN).

A su vez, puedo llenar cualquier espacio sobrante con DENSE.

CSS FLEXBOX:



<introducción teórica> Un nuevo modelo de LAYOUT, un modelo de cómo aparecerán elementos en pantalla. Podemos alinear elementos en el eje vertical y el horizontal, redefinir el orden en el que dibuje el navegador los elementos que no es lo mismo que la posición.

Vuelve a requerir la lógica de padre e hijo, necesariamente.

Al utilizar la propiedad FLEX en el padre, inicio automáticamente con MAIN AXIS (horizontal) y el CROSS AXIS (vertical), sabiendo puedo manipular dichos ejes.

FLEX DIRECTION permite cambiar el eje principal (que por defecto es ROW)

Para alinear elementos en el eje principal utilizamos la propiedad JUSTIFY-CONTENT

Propiedades del JUSTIFY-CONTENT, el FLEX-START (default), el FLEX-END [estos dos van respecto a los ejes en los que trabajemos, SPACE-BETWEEN distribuye los elementos uniformemente a lo largo del eje en el que se encuentre, SPACE-AROUND (es como que le diera una especie de márgenes entre los elementos) que hace que el espacio alrededor de los elementos se distribuya, y CENTER, que se centran sobre el eje.

OTRA PROPIEDAD DE FLEX, ALING – ITEMS, alinea los elementos respecto al eje secundario; posee las siguientes propiedades FLEX-START; FLEX-END; STRETCH (default) RELLENA (si no le hemos puesto tamaño con WITDH Y HEIGTH a los elementos) todo el eje secundario y tengo BASELINE- alinea los elementos respecto a la línea base del texto. /****ojo que todas estas propiedades se aplican siempre a los contenedores****/

/***ahora si colocaré una propiedad aplicada al ítem***/

ALING-SELF sobrescribe para un único elemento los valores de ALING-ITEMS;

→Como datos curiosos el FLEX siempre intenta meter todo en la misma línea y sobrescribe los tamaños que se la hayan dado, si ponemos tamaños específicos que no deseemos perder necesitamos colocar una propiedad en el contenedor (padre) FLEX-WRAP: WRAP, que hará que salte a la siguiente línea si no encuentra espacio.

Trabajo con ejes verticales y horizontales, su funcionalidad principal sirve para alinear objetos. [temporal]---

Algunas de sus ventajas:

- Flexbox es la mejor opción para alinear el contenido dentro de los elementos.
- Utiliza Flexbox para posicionar los detalles más pequeños de un diseño.
- Flexbox funciona mejor sólo en una dimensión (filas o columnas).

Bibliografía:

https://www.arsys.es/blog/programacion/diseno-web/grid-flexbox-mejor-diseno-maquetacion-web/

https://en.wikipedia.org/wiki/CSS_grid_layout

https://www.w3schools.com/css/css_grid.asp

https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout

https://css-tricks.com/snippets/css/complete-guide-grid/

--- referencias más dinámicas, para practicar ---

https://learncssgrid.com/

http://cssgridgarden.com/#es

- FLEXBOX

https://flexboxfroggy.com/#es

https://www.youtube.com/user/escueladigitalperu/videos