**UNIDADES DE**MEDIDA EN CSS3

En CSS3 es necesario utilizar medidas ya que son usadas por muchas de las propiedades más importantes. Con estas propiedades definimos altos, anchos, márgenes y tamaños de letra entre muchos otros formatos. Hasta el momento hemos mencionado los píxeles, pero en realidad en la web existen dos grandes grupos de unidades de medida disponibles para realizar nuestras maquetaciones y estructuras HTML de forma profesional.

**Unidades Absolutas**

Estas unidades son las que están completamente definidas y no dependen de otros valores de medida. Entre las unidades de medida absoluta se encuentran:

* **in:** (Pulgadas en inglés). Una pulgada equivale a 2.54cm.
* **cm:** Centímetros.
* **mm:** Milímetros.
* **pt:** Puntos, Un punto equivale a 1 pulgada/72, aproximadamente 0.35 milímetros.
* **pc:** Picas. Una pica equivale a 12 puntos, aproximadamente 4.23 milímetros.

La ventaja de las unidades absolutas es que siempre tendremos un valor fijo, sin preocuparnos por cálculos externos, siempre estaremos seguros de que el valor definido no cambiará en ninguna pantalla ni resolución en la que sea vista nuestra página web. La desventaja de las unidades de medida absolutas es que son las menos eficientes a la hora de trabajar bajo el modelo de Mobile-first o Responsive Design, ya que las resoluciones y medidas de pantalla cambian constantemente.

**Unidades Relativas**

Las unidades de medida son mucho más flexibles, ya que dependen de otros valores de referencia para adaptarse. Un ejemplo de esta unidad es el porcentaje (%), Si aplicamos al hijo de un elemento HTML una propiedad para que ocupe el 50% del ancho de su padre, éste siempre medirá exactamente el 50%, sin importar el tamaño de su padre. Si el elemento padre se hace más grande o más pequeño el hijo siempre ocupará la mitad del valor total (50%).

Las unidades de medida relativas se dividen en dos partes:

**Relativas a la tipografía**

Existen tres tipos de medidas relativas que se basan en el tamaño de la tipografía.

* **em:** Se basa en el elemento que contenga al elemento.

Ejemplo:   
  
<div style=”font-size: 30px”>   
    <p id=”miParrafo”>Lorem ipsum…. </p>   
</div>   
#miParrafo { width: 2em; // 1em = 30px, 2em = 60px } 

En este ejemplo, le estamos diciendo a la etiqueta < p > hija de un element < div > con un tamaño de letra de 30px, que tenga un ancho de 2em. En este caso 1em es equivalente a la medida de la Fuente del elemento padre o sea 30px, por tanto, en este caso 2em es igual a 60px.

* **rem:** Se basa en el tamaño de fuente que tenga la raíz del documento HTML es decir el < body >.

Ejemplo:   
  
<body>   
    <div style=”font-size: 30px”>   
        <p id=”miParrafo”>Lorem ipsum…. </p>   
   </div>   
</body>   
body {   
font-size: 16px;   
}   
#miParrafo {   
width: 2rem; // 1rem = 16px, 2rem = 32px   
} 

* **ex:**relativa con respecto a la altura de la letra “x” minúscula del tipo y tamaño de letra del elemento padre.

**Relativas al viewport (Área visible del navegador/ventana)**

* **vw:** (Viewport Width) Relativo al ancho del viewport. Si el ancho es 1280px entonces 1vw es equivalente a 1280px.
* **vh:** (Viewport Height) Relativo al alto del viewport, si el alto son 900px entonces 1vh es equivalente a 900px.
* **vmin:** Entre vw y vh toma el que tenga menor valor.
* **vmax:** Entre vw y vh toma el que tenga mayor valor.

Las medidas relativas al viewport son muy importantes cuando estamos haciendo diseños web que se adaptarán a múltiples dispositivos.