Repaso XHTML y CSS

Layout *(disposición)*

El diseño de las páginas web habituales se divide en bloques: **cabecera, menú, contenidos y pie de página**. Una solución muy básica, para evitar grandes espacios en blanco, consiste en crear páginas con una anchura fija adecuada y centrar la página horizontalmente respecto de la ventana del navegador. A continuación, se muestra el aspecto de una página centrada a medida que aumenta la anchura de la ventana del navegador.

Utilizando la propiedad `margin` de CSS, podemos centrar una página web horizontalmente. La solución consiste en agrupar todos los contenidos de la página en un elemento `<div>` y asignarle a ese `<div>` unos márgenes laterales automáticos. El `<div>` que encierra los contenidos se suele llamar `contenedor` (*en inglés* se denomina `wrapper` o `container`):

```
#contenedor {
   width: 300px;
   margin: 0 auto;
}

<body>
  <div id="contenedor">
    <h1> Lorem ipsum dolor sit amet ... </h1>
  </div>
</body>
```

El valor `0 auto` significa que los márgenes superior e inferior son iguales a `0` y los márgenes laterales (izquierdo y derecho) toman un valor de `auto`, es decir, se reparten el espacio equitativamente.

Sin embargo, modificando el código CSS anterior se puede conseguir un diseño dinámico o *fluido,* que se adapta a la anchura de la ventana del navegador y permanece siempre centrado. Estableciendo la anchura del elemento contenedor mediante un porcentaje, su anchura se adapta de forma continua a la anchura de la ventana del navegador.

```
#contenedor {
    width: 70%;
    margin: 0 auto;
}
```

De esta forma, si se reduce la anchura de la ventana del navegador, la página se verá más estrecha y si se maximiza la ventana del navegador, la página se verá más ancha, como muestran las imágenes a continuación:

Alturas/anchuras máximas y mínimas

Cuando se diseña la **estructura de una página** web, se debe tomar la decisión de optar por un diseño de anchura fija o un diseño cuya anchura se adapta a la anchura de la ventana del navegador. Sin embargo, la mayoría de las veces **sería conveniente que la anchura de la página sea variable y se adapte a la anchura de la ventana del navegador, pero respetando ciertos límites.** En otras palabras, que la anchura de la página no sea tan pequeña como para que no se puedan mostrar correctamente los contenidos y tampoco sea tan ancha como para que las líneas de texto no puedan leerse cómodamente.

CSS define cuatro propiedades que permiten limitar la anchura y altura mínima y máxima de cualquier elemento de la página. Las propiedades son: **max-width, min-width, max-height y min-height.**

De esta forma, para conseguir un diseño de anchura variable pero controlada, se podrían utilizar reglas CSS como la siguiente:

```
#contenedor {
    min-width: 500px;
    max-width: 900px;
}
```

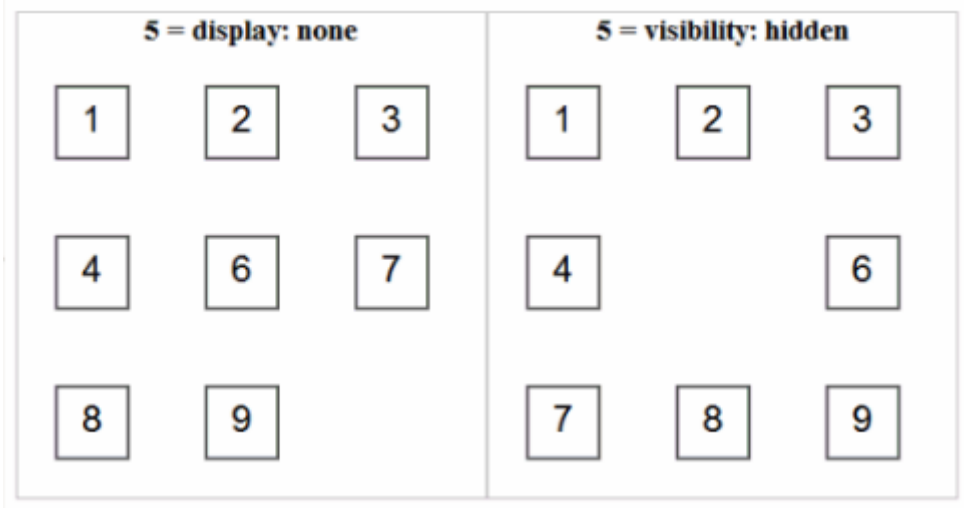
Visualización

CSS define cuatro propiedades para controlar su visualización: display, visibility, overflow y z-index.

Las propiedades display y visibility controlan la visualización de los elementos. Habitualmente se utilizan junto con JavaScript para crear efectos dinámicos como mostrar y ocultar determinados textos o imágenes.

La propiedad display permite ocultar completamente un elemento haciendo que desaparezca de la página. Como el elemento oculto no se muestra, el resto de elementos de la página se mueven para ocupar su lugar.

Por otra parte, la propiedad visibility permite hacer invisible un elemento, lo que significa que el navegador crea la caja del elemento pero no la muestra. En este caso, el resto de elementos de la página no modifican su posición, ya que aunque la caja no se ve, sigue ocupando sitio.



Las posibilidades de la propiedad `display` son más avanzadas que simplemente ocultar elementos. En realidad, la propiedad `display` modifica la forma en la que se visualiza un elemento. Los valores más utilizados son **inline**, **block**, **inline-block** y **none**. El valor **block** muestra un elemento como si fuera un elemento de bloque, independientemente del tipo de elemento que se trate. El valor **inline** visualiza un elemento en forma de elemento en línea, independientemente del tipo de elemento que se trate. El valor **none** oculta un elemento y hace que desaparezca de la página. El resto de elementos de la página se visualizan como si no existiera.

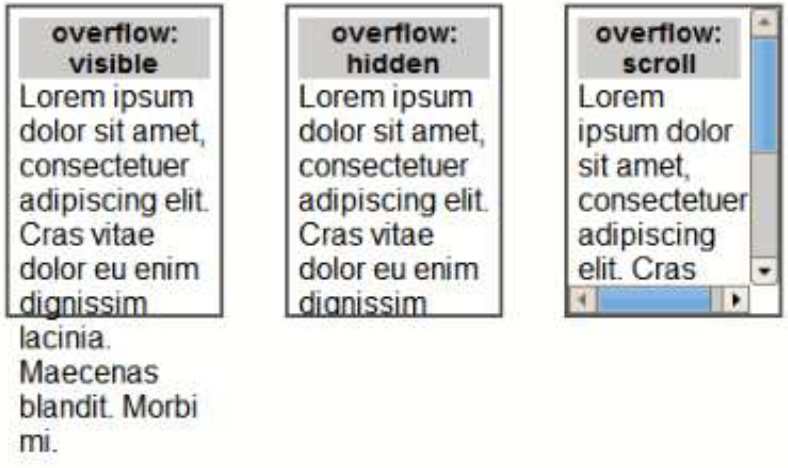
https://www.w3schools.com/CSSref/pr_class_display.asp

Propiedad `overflow`

Normalmente, los contenidos de un elemento se muestran en el espacio reservado para ese elemento. Sin embargo, en algunas ocasiones el contenido de un elemento no cabe en el espacio reservado y se desborda.

Los valores de la propiedad `overflow` tienen el siguiente significado:

- **visible**: el contenido no se corta y se muestra sobresaliendo la zona reservada para visualizar el elemento. Este es el comportamiento por defecto.
- **hidden**: el contenido sobrante se oculta y sólo se visualiza la parte del contenido que cabe dentro de la zona reservada para el elemento.
- **scroll**: solamente se visualiza el contenido que cabe dentro de la zona reservada para el elemento, pero también se muestran barras de *scroll* que permiten visualizar el resto del contenido.
- **auto**: el comportamiento depende del navegador, aunque normalmente es el mismo que la propiedad **scroll**.



Propiedad `z-index`

CSS permite también controlar la posición tridimensional de las cajas posicionadas. De esta forma, es posible indicar las cajas que se muestran delante o detrás de otras cajas **cuando se producen solapamientos**. La posición tridimensional de un elemento se establece sobre un tercer eje llamado Z y se controla mediante la propiedad `z-index`.

El valor más común de la propiedad `z-index` es un número entero. Cuanto más alto sea el valor numérico, más cerca del usuario se muestra la caja. **Esta propiedad sólo tiene efecto en los elementos posicionados, por lo que es necesario que `z-index` vaya acompañada de la propiedad `position`.** Si debes posicionar un elemento pero no quieres moverlo de su posición original ni afectar al resto de elementos de la página, utiliza el posicionamiento relativo (`position: relative`).