

2. CSS

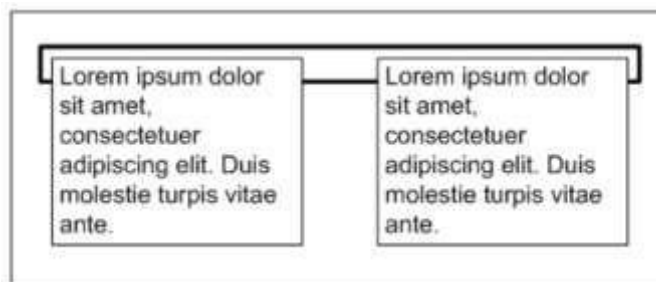
1. Galería: En una galería de imágenes, éstas se mostrarán siempre con el mismo tamaño, por lo que para que el número de imágenes que se muestran en cada fila se adapte al tamaño de la pantalla, tendremos que utilizar posicionamiento de tipo float. Por lo demás, lo único que habrá que modificar será el margen, el espaciado y el borde que queremos emplear para ubicar cada imagen:

```
#galeria {
  max-width: 650px;
}
#galeria img { float: left; margin: 1em; padding: .8em;
  border: 1px solid #CCC;
}
#galeria .padding{ padding: .3em;
}
#galeria .margin{ margin: .3em;
}
#galeria .border{
  border: 2px solid #000;
}
<div id="galeria">
  
  
  
  
  
  
  
    </div>
```

2. Visualización: Cuando usamos posicionamiento “float”, lo elementos “desaparecen” del flujo normal del documento y pasan a “flotar” sobre éste. De esta forma, es posible que algunos o todos los elementos flotantes se salgan de su elemento contenedor (si los elementos interiores están posicionados de forma flotante y el elemento contenedor no dispone de más contenidos):

```
#contenedor { border: thick solid #000;
}
#izquierda { float: left; width: 40%; border:solid;
}
#derecha { float: right; width: 40%; border:solid;
}
<div id="contenedor">
  <div id="izquierda">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis
    molestie turpis vitae ante.</div>
```

```
<div id="derecha">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla  
bibendum mi non lacus.</div> </div>
```



Una opción para solucionar esto sería incluir un nuevo elemento dentro del contenedor principal que sea invisible, usando la propiedad `visibility:hidden`, y no permitir que tenga elementos float a sus lados (`clear:both`), con lo cual éste se ubicará bajo todos los floats:

```
<div style="clear: both; visibility:hidden"></div>
```

3. Uso de iconos: Este tipo de imágenes, que son decorativas en lugar de imágenes de contenido, se deberían añadir con CSS y no con elementos de tipo ``, utilizando la propiedad `background` (y `background-image`), mostrando una imagen de fondo sin repetición en el enlace y añadir el padding necesario al texto del enlace para que no se solape con la imagen de fondo:

```
.boton{ background: lightgrey url("correo.png") no-repeat left center;  
padding-left:30px; }
```

CSS nos permite también definir los enlaces como si fuesen botones, dando aspecto a las cajas que los contienen y jugando con sus bordes, relleno y márgenes:

```
a { margin: 1em 0; float: left; clear: left; text-decoration: none;  
color:black;  
} .boton {  
padding: 4px 4px 4px 30px; text-align: right; background: lightgrey  
url("correo.png") no-repeat left center; }  
<a class="boton" href="#">Contacta</a>
```



4. Menú horizontal: Los elementos de un menú se crean por medio de listas desordenadas. Para mostrarlo en forma de menú horizontal, tendríamos que cambiar un par de cosas:

- Ocultar los iconos que se muestran por defecto en cada elemento de la lista (`list-style-type:none`).

- Para que el menú sea horizontal, tendríamos que mostrar los elementos de la lista () en línea, es decir, con display:inline.
- Para separar cada uno de los elementos, mostraríamos el borde lateral derecho como separación (border-right:solid). Podemos modificar el último elemento para que no muestre su borde derecho (border-rigth:none).

```
ul {
  list-style-type: none; margin: 0; padding: 0;
  min-width: 160px; position:fixed; left:10px; background-color: #f1f1f1;
} li{
  display:inline-block; border-right: 1px solid #bbb;
}
li:last-child { border-right: none;
}
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">Enlace 1</a></li>
<li><a href="#">Enlace 2</a></li>
<li><a href="#">Enlace 3</a></li> </ul>
```

5. Menú horizontal desplegable: si queremos que el menú sea desplegable, tendríamos que jugar con la visibilidad de los elementos desplegados para mostrarlos únicamente cuando el ratón esté situado sobre su elemento padre:

- Los elementos del submenú en principio estarán ocultos (display:none), y los posicionaremos bajo el elemento del que dependen (position:absolute)
- Cuando se pase el ratón sobre el elemento del menú del que dependen, se mostrará el submenú

```
.desplegable{ display:none; position:absolute; background-color: #f9f9f9; min-
width:160px; box-shadow:0px 8px 16px 0px rgba(0,0,0,0.2); padding: 12px 16px;
}
li.video:hover .desplegable{
  display:block;
}
```

6. Menú vertical: Si queremos hacer el menú vertical, solamente tendremos que hacer un par de cambios:

- Cambiar el display de los elementos de la lista, para que se muestren en bloque.

- Cambiar el borde derecho por el borde inferior

```
ul { list-style-type: none; margin: 0; padding: 0; min-width: 160px;
width:200px; background-color: #f1f1f1;
} li{ border-bottom: 1px solid #bbb; left:30px;
}
li:last-child{
border-bottom: none;
}
```

7. Formularios: En un formulario podemos hacer varias cosas para optimizar su visualización:

- Añadir padding a los campos de texto: por defecto, los campos de texto de los formularios no incluyen ningún espacio de relleno, por lo que el texto introducido por el usuario aparece pegado a los bordes del cuadro de texto. Añadiendo un pequeño padding a cada elemento `<input>`, se mejora notablemente el aspecto del formulario:

```
input {padding: .2em;}
```

Sin padding	con padding
<input type="text" value="lorem ipsum"/>	<input type="text" value="lorem ipsum"/>
Contraseña	Contraseña
<input type="password" value="*****"/>	<input type="password" value="*****"/>

- Etiquetas alineadas y formateadas: Los elementos `<input>` y `<label>` de los formularios son elementos en línea, por lo que por defecto se muestran pegados uno a otro. Podemos hacer un par de cambios para separar unos de otros:

- Meter cada par de (etiqueta,input) dentro de un div
- Especificar un ancho fijo para el elemento label dentro del div

```
div {margin: .4em 0;}
div label {width: 25%; float: left;}
<form>
<fieldset>
<legend>Alta en el servicio</legend>
<div>
<label for="nombre">Nombre</label>
<input type="text" id="nombre" />
</div>
<div>
<label for="apellidos">Apellidos</label>
<input type="text" id="apellidos" size="35" />
</div>
</div>
```

```

<label for="dni">DNI</label>
<input type="text" id="dni" size="9" />
</div>
<div>
<label for="pass">Contraseña</label>
<input type="password" id="pass"/>
</div>
<input class="btn" type="submit" value="Dar de alta" />
</fieldset>
</form>

```

Alta en el servicio

Nombre

Apellidos

DNI

Contraseña

- c. Resaltar el campo seleccionado: podemos emplear la pseudo-clase `:focus` de CSS para que se resalte el elemento en el que estamos introduciendo datos

```
input:focus { border: 2px solid #000; background: #F3F3F3; }
```

Alta en el servicio

Nombre

Apellidos

DNI

Contraseña

8. Tablas: En las tablas también hay varias modificaciones habituales que podemos llevar a cabo:

- a. Incrementar el espaciado vertical de las celdas, para lo que usaremos el `padding`

```

th,td{
    padding: 0.3em;
}

```

- b. Mover de sitio el título de la tabla (con `caption-side`):

```
table{ caption-side: top;}
```

- c. Aplicar “líneas de cebra” (pintar el fondo de las filas pares de color diferente a las impares).

```
.par th, .par td {    background: #eee;
}
.impar th, .impar td {    background: #FFF;
}

<table>
<caption>Tabla de ejemplo</table>
<tr class="par">
  <td>1.1</td><td>1.2</td>
</tr>
<tr class="impar">
  <td>2.1</td><td>2.2</td>
</tr>
<tr class="par">
  <td>3.1</td><td>3.2</td>
</tr>
<tr class="impar">
  <td>4.1</td><td>4.2</td>
</tr>
</table>
```

- d. Eliminar bordes. Dejar, por ejemplo, únicamente el borde inferior (podríamos resaltar más el borde inferior de la última fila, por ejemplo):

```
th,td{
  border-bottom: 1px solid #999;
}
th:lastchild, td:lastchild { border-bottom: 2px solid #999; }
```