



Elementos en bloque y en línea

En HTML y CSS, los elementos se clasifican en dos categorías principales: elementos en línea (inline) y elementos de bloque (block). Esta clasificación determina cómo se comportan estos elementos en el flujo de la página web.

Elementos en Línea (Inline Elements)

Comportamiento y Características:

- Flujo de Contenido: Los elementos en línea se sitúan dentro del flujo normal del texto. No comienzan en una nueva línea.
- Dimensiones: No aceptan valores de ancho (width) y alto (height). Su tamaño se determina por su contenido.
- Margen y Relleno (Padding y Margin): Aceptan márgenes y paddings, pero solo afectan horizontalmente (izquierda y derecha).

Usos y Ejemplos:

Utilizados para contenido que no necesita romper el flujo de texto, como `<a>`, ``, `` y ``. Ejemplo: `` es frecuentemente usado para aplicar estilos a una porción específica del texto.

Elementos de Bloque (Block Elements)

Comportamiento y Características:

- Flujo de Contenido: Comienzan en una nueva línea y ocupan el ancho completo disponible del contenedor padre.
- Dimensiones: Pueden tener un ancho (width) y alto (height) definidos. Por defecto, el ancho es de 100% del contenedor padre.
- Margen y Relleno: Aceptan valores de margen y relleno en todas las direcciones, y estos afectan a su disposición en la página.

Usos y Ejemplos:

Utilizados para estructurar y organizar grandes secciones de contenido, como párrafos (`<p>`), encabezados (`<h1>-<h6>`), y secciones (`<div>`). Ejemplo: `<div>` es un contenedor genérico utilizado para agrupar otros elementos y aplicarles estilos.



Debemos entender las diferencias entre ambos elementos para poder crear nuestras estructuras luego con CSS. No podemos tratar por igual a un elemento en bloque y en línea, es uno de los errores más comunes cuando se comienza a maquetar. Y muchas veces, esta es la parte más difícil, diferenciarlos.

Incorporación de Estilos con la Etiqueta `<link>`

Concepto

La etiqueta `<link>` en HTML se utiliza para enlazar hojas de estilo externas (CSS) al documento HTML. Esto permite separar el contenido (HTML) del diseño y formato (CSS), facilitando la gestión y mantenimiento del código.

Estructura de la Etiqueta Link

La etiqueta `<link>` tiene varios atributos importantes:

- **rel (relationship):** Indica el tipo de relación que el documento HTML tiene con el recurso vinculado. Para las hojas de estilo CSS, se utiliza `rel="stylesheet"`.
- **href:** Indica la ruta del archivo de hoja de estilo. Puede ser una ruta relativa o absoluta.

💡 En VSC tienes un snippet que es `link:css + TAB` que completa la etiqueta con lo necesario para linkear una hoja de estilos. Puedes ver todos los snippets de EMMET en este [link](#).

Ejemplo de Uso

Supongamos que tienes una hoja de estilo llamada `style.css` en la misma carpeta que tu archivo HTML. Para vincularla, tu etiqueta `<link>` en el `<head>` de tu documento HTML se vería así:

Unset

```
<head>
  <link rel="stylesheet" href="style.css" />
</head>
```

Siguiendo el último código que vimos, se vería de la siguiente manera:

HTML (index.html)

Unset

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <div class="container">
      <div class="item item-1">Item 1</div>
      <div class="item item-2">Item 2</div>
      <div class="item item-3">Item 3</div>
    </div>
  </body>
</html>
```

CSS (style.css)

Unset

```
.container {
  width: 600px;
  margin: 60px auto;
```

```
/* Estilos de FlexBox */
display: flex;
justify-content: center;
align-items: center;
}
.item {
  flex-grow: 1;
}
.item-1 {
  background-color: salmon;
}
.item-2 {
  background-color: lightblue;
}
.item-3 {
  background-color: lightgreen;
}
```

RESULTADO



Ventajas de Usar la Etiqueta Link

1. **Separación de preocupaciones:** Al mantener el HTML y CSS en archivos separados, tu código es más organizado y fácil de mantener.
2. **Reutilización:** Puedes usar la misma hoja de estilo en múltiples páginas HTML, lo que garantiza un diseño coherente en todo el sitio web.
3. **Caché del navegador:** Los navegadores suelen almacenar en caché las hojas de estilo externas, lo que significa que una vez descargadas, no necesitan serlo de nuevo para otras páginas del mismo sitio, lo que reduce los tiempos de carga.

Buenas Prácticas

- Coloca la etiqueta `<link>` dentro del `<head>` para que los estilos se carguen antes que el contenido de la página.

- Utiliza nombres claros y descriptivos para tus archivos CSS (como `main.css`, `mobile.css`).
- Minimiza y optimiza tus archivos CSS para mejorar los tiempos de carga.

La incorporación de estilos a través de la etiqueta `<link>` es una técnica estándar en el desarrollo web moderno, proporcionando un método eficiente y organizado para controlar el aspecto visual de tus páginas web.

CSS Grid Layout

Concepto de CSS Grid

CSS Grid Layout es un sistema de diseño bidimensional en CSS. Permite manejar tanto filas como columnas, proporcionando un nivel de control más complejo y preciso sobre los diseños web en comparación con Flexbox, que es principalmente unidimensional.

Componentes Principales de CSS Grid

1. **Grid Container:** El elemento principal que define un espacio de grid. Se establece utilizando `display: grid` o `display: inline-grid`.
2. **Grid Items:** Los hijos directos del grid container. Cada elemento hijo se coloca en la cuadrícula.

Propiedades Principales del Grid Container

- **grid-template-columns y grid-template-rows:** Definen las columnas y filas del grid, especificando el tamaño de cada una. Puedes usar valores fijos (como `px`), flexibles (`fr` para fracciones) o una combinación de ambos.
- **grid-gap:** Establece el espacio entre filas y columnas.
- **grid-auto-rows y grid-auto-columns:** Especifican el tamaño de las filas y columnas que se crean implícitamente.
- **justify-items y align-items:** Alinean los elementos dentro de sus celdas en el eje horizontal y vertical, respectivamente.
- **justify-content y align-content:** Alinean el grid completo dentro del contenedor si el grid es más pequeño que el área del contenedor.

Propiedades de los Grid Items

- **grid-column-start, grid-column-end, grid-row-start, y grid-row-end:** Especifican dónde comienza y termina un elemento en la cuadrícula, permitiendo que los elementos ocupen más de una celda.

- **grid-column** y **grid-row**: Versiones abreviadas para definir el inicio y el fin de un elemento en las filas y columnas.
- **grid-area**: Permite especificar simultáneamente dónde comienza y termina un elemento en ambas, filas y columnas.

Ejemplo Básico

HTML

Unset

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <div class="container">
      <div class="item item-1">Item 1</div>
      <div class="item item-2">Item 2</div>
      <div class="item item-3">Item 3</div>
    </div>
  </body>
</html>
```

CSS (style.css)

Unset

```
.container {
  width: 600px;
  margin: 60px auto;

  /* Estilos de Grid */
  display: grid;
```

```
    justify-content: center;
    align-items: center;
}
.item {
    flex-grow: 1;
}
.item-1 {
    background-color: salmon;
}
.item-2 {
    background-color: lightblue;
}
.item-3 {
    background-color: lightgreen;
}
```

RESULTADO



Uso de CSS Grid

CSS Grid es particularmente útil para:

- Crear diseños complejos y multidimensionales.
- Diseñar interfaces de usuario que requieren una alineación precisa en filas y columnas.
- Construir layouts responsivos que se adaptan a diferentes tamaños de pantalla.

Buenas Prácticas

- Usa CSS Grid para diseños más complejos que requieren control bidimensional.
- Combina CSS Grid con medidas relativas (como %, vw, vh, fr) para lograr diseños responsivos.
- Utiliza la propiedad **grid-gap** para un espaciado consistente entre elementos del grid.

CSS Grid es una herramienta poderosa en el desarrollo front-end, proporcionando a los diseñadores y desarrolladores una manera flexible y eficiente de crear diseños web avanzados.

Uso de Imágenes de Fondo en CSS

Concepto de Background con Imagen

El uso de imágenes de fondo en CSS permite a los diseñadores web añadir imágenes detrás del contenido de un elemento. Esta técnica se utiliza para mejorar la estética visual, resaltar secciones específicas, o crear un contexto visual para el contenido.

Propiedades Relacionadas con Background

1. **background**: Define una o más imágenes como fondo de un elemento.
 - **Sintaxis**: `background: url('ruta-de-la-imagen.jpg');`
2. **background-size**: Controla el tamaño de la imagen de fondo.
 - **Valores Comunes**:
 - **cover**: Escala la imagen para cubrir completamente el elemento.
 - **contain**: Escala la imagen para ajustarse dentro del elemento.
 - Valores específicos (como `100px 200px`).
3. **background-repeat**: Define si la imagen se repetirá y cómo.
 - **Valores Comunes**:
 - **no-repeat**: No repite la imagen.
 - **repeat**: Repite la imagen en ambas direcciones.
 - **repeat-x** o **repeat-y**: Repite la imagen horizontal o verticalmente.
4. **background-position**: Establece la posición inicial de la imagen de fondo.
 - **Valores**: Pueden ser palabras clave (`top`, `bottom`, `left`, `right`, `center`) o valores específicos (`50% 20%`, `10px 30px`).
5. **background-attachment**: Determina si la imagen de fondo es fija o se desplaza con el resto de la página.
 - **Valores**: `scroll`, `fixed`, `local`.

Ejemplo Completo

Unset

```
.elemento {  
  background: url("imagen.jpg");  
  background-size: cover;  
  background-repeat: no-repeat;  
  background-position: center;  
  background-attachment: fixed;  
}
```

Este ejemplo establece una imagen de fondo que cubre todo el elemento, centrada, sin repetirse y fija al hacer scroll.

Uso Práctico

Las imágenes de fondo son útiles para:

- Crear banners o secciones con un fuerte impacto visual.
- Establecer un ambiente o tema visual en una página.
- Mejorar la experiencia del usuario con elementos visuales interesantes.

Buenas Prácticas

- Opta por imágenes de alta calidad pero optimizadas para no afectar el tiempo de carga de la página.
- Asegúrate de que el texto sobre la imagen de fondo sea legible.
- Utiliza imágenes de fondo de manera estratégica para no distraer del contenido principal.

El uso adecuado de imágenes de fondo puede mejorar significativamente el diseño de un sitio web, haciendo que sea más atractivo y enriqueciendo la experiencia del usuario.

Pseudo Clases en CSS

Concepto de Pseudo Clases

Las pseudo clases son palabras clave que se añaden a los selectores en CSS para especificar un estado especial de un elemento seleccionado. Permiten aplicar estilos a un elemento no solo en función de su información en el árbol del documento, sino también en respuesta a la interacción del usuario, su estado o posición.

Uso Común de Pseudo Clases

1. **:hover**: Aplica estilos cuando el usuario pasa el cursor sobre un elemento. Muy utilizado para botones y enlaces.

Unset

```
a:hover {  
  color: blue;  
}
```

2. **:focus**: Usado para estilizar un elemento (como un campo de formulario) cuando está enfocado.

Unset

```
input:focus {  
  border-color: green;  
}
```

3. **:active**: Aplica estilos en el momento que un elemento (como un botón o enlace) es activado o clickeado por el usuario.

Unset

```
button:active {  
  background-color: darkblue;  
}
```

4. **:first-child** y **:last-child**: Selecciona el primer o último elemento hijo de un contenedor.

Unset

```
li:first-child {
```

```
font-weight: bold;

}
```

5. **:nth-child()**: Selecciona uno o más elementos según su orden en los hijos de su contenedor, aceptando fórmulas y palabras clave.

Unset

```
p:nth-child(2n) {
  color: red;
}
```

Importancia de las Pseudo Clases

Las pseudo clases son esenciales para:

- Mejorar la interactividad: Responder a acciones del usuario como pasar el cursor, hacer clic, etc.
- Aplicar estilos específicos a elementos en ciertas posiciones: Por ejemplo, el primer elemento, el último, o cada tercer elemento.
- Controlar la apariencia de elementos en distintos estados: Como cuando un enlace ha sido visitado (**:visited**) o un checkbox está marcado (**:checked**).

Buenas Prácticas

- Utiliza las pseudo clases para crear interfaces de usuario más dinámicas e interactivas.
- Combina las pseudo clases con transiciones y animaciones para efectos visuales más suaves.
- Asegúrate de que los estilos aplicados mejoren la experiencia del usuario y no la obstaculicen, especialmente en términos de accesibilidad.

Las pseudo clases en CSS son herramientas poderosas que permiten a los desarrolladores web crear experiencias más ricas e interactivas para los usuarios. Su correcta utilización puede aumentar significativamente la usabilidad y atractivo visual de un sitio web.