



Docs_

Tips que te salvarán la vida



5 tips para html que te salvarán la vida

- Ver código adjunto en formato ZIP

Hay veces que te toca revisar aquellos archivos HTML. Tú, como desarrollador, dispuesto a hacer tu trabajo rápidamente, esperas que el código esté bien estructurado, semánticamente claro, y sin estilos en línea. Bueno, hay veces que no vas a encontrar el código así, pero puedes comenzar a cambiar eso, aplicando una serie de buenas prácticas para que otros devs encuentren tu código HTML de lo más limpio posible. Y he aquí 5 tips que harán que tu código sea más prolijo y claro para otros devs.

- El HTML importa
- Indentación
- ¿Tabs o Espacios?
- HTML semántico
- Separar el CSS en archivos externos

En este post, voy a ir explicando cómo aplicar estas técnicas y por qué partiendo de un código ilegible hasta volverlo código prolijo y claro. Entonces, si estás interesado en mejorar en buenas prácticas o si tan solo quieres repasar tus fundamentos, te recomiendo seguir leyendo este post.

Docs_ Generalidades Curiosidades

El HTML importa

HTML en sí no es un lenguaje de programación. Es un lenguaje de marcado, significa que HTML define la estructura del contenido de un sitio web. Esto, algunas veces, hace que no sea tan relevante en comparación con lenguajes de programación, como Javascript, y por tanto, es descuidado en el desarrollo. Pero, **realmente importa**. Si nosotros como desarrolladores descuidamos el HTML, no podremos reconocer las secciones de nuestra web, por tanto, esto hará que nuestro código no sea mantenible y no pueda escalar en el tiempo.

Vamos a comenzar con un código HTML con muchas malas prácticas y poco a poco, convertiremos el código a uno de calidad aplicando buenas prácticas.

```
<!DOCTYPE html>
<html>

<head>
  <title>Tu Blog</title>
</head>

<body style="margin: 0">
  <div
    style="display: grid; height: 100vh; grid-gap: 5px; grid-template-columns:
repeat(2, 1fr); grid-template-rows: 100px 90px 1fr 90px; grid-template-areas: 'header
header' 'nav nav' 'aside main' 'footer footer';">
    <div
      style="display: flex; justify-content: center; align-items: center; font-
size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: header;
background-color: lightblue;">
      HEADER</div>
    <div
      style="display: flex; justify-content: center; align-items: center; font-
size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: nav;
background-color: #cccc;">
      NAV</div>
    <div
      style="display: flex; justify-content: center; align-items: center; font-
size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: aside;
background-color: yellow;">
      ASIDE</div>
    <div
      style="display: flex; justify-content: center; align-items: center; font-
size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: main;
background-color: orange;">
      CONTENT</div>
    <div
```

Docs_ Generalidades Curiosidades

```
        style="display: flex; justify-content: center; align-items: center; font-size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: footer; background-color: peru;">
        FOOTER</div>
    </div>
</body>

</html>
```

Indentación

El bloque de HTML previo puede funcionar sin problemas, el navegador es capaz de leerlo e interpretarlo. Pero, y nosotros? Probablemente sí, porque el código es pequeño y legible a simple vista. Sin embargo, imagínense ese código en mil o dos mil líneas. No podríamos saber qué elementos son padres o hijos de otros elementos, y aunque funcione, se vuelve tedioso para otros desarrolladores buscar la relación de cada elemento con los demás, por lo que es importante que se especifique a los elementos por niveles en el editor de código, solo así podremos saber quién es el ancestro, padre o hijo de cada elemento. Así que sé bueno e indenta tu código!

Tu reto es indentar el bloque HTML previo. Sabrás que lo habrás conseguido cuando puedas distinguir quién es el elemento padre y el elemento hijo. Dejaré el reto resuelto más abajo. Sin embargo, no bajes hasta que lo hayas intentado, en la práctica está la maestría.

```
<!DOCTYPE html>
<html>

<head>
    <title>Tu Blog</title>
</head>

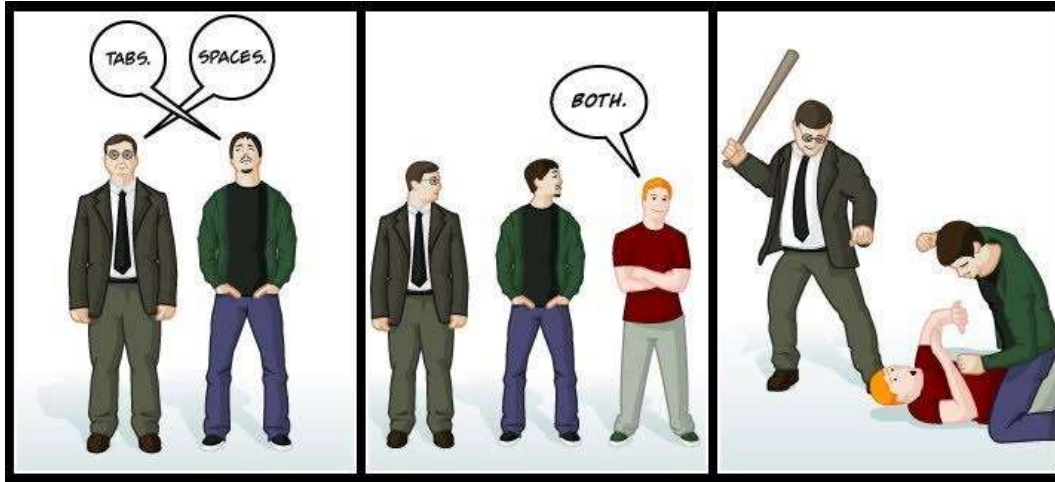
<body style="margin: 0">
    <div
        style="display: grid; height: 100vh; grid-gap: 5px; grid-template-columns:
repeat(2, 1fr); grid-template-rows: 100px 90px 1fr 90px; grid-template-areas: 'header
header' 'nav nav' 'aside main' 'footer footer';">
        <div
            style="display: flex; justify-content: center; align-items: center; font-size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: header; background-color: lightblue;">
            HEADER
        </div>
    </div>
```


Docs_ Generalidades Curiosidades

```
        style="display: flex; justify-content: center; align-items: center; font-size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: nav; background-color: #cccc;">
        NAV
    </div>
    <div
        style="display: flex; justify-content: center; align-items: center; font-size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: aside; background-color: yellow;">
        ASIDE
    </div>
    <div
        style="display: flex; justify-content: center; align-items: center; font-size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: main; background-color: orange;">
        CONTENT
    </div>
    <div
        style="display: flex; justify-content: center; align-items: center; font-size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: footer; background-color: peru;">
        FOOTER
    </div>
</div>
</body>
</html>
```

Docs_ Generalidades Curiosidades

¿Tabs o Espacios?



Tabs o espacios es uno de las discusiones eternas que se van a dar en cada foro de programación. Entre tantas opiniones, lo mejor es experimentar con ambos y ver con cuál te sientes más cómodo, ~~en silencio, sino se quiere terminar como en la imagen~~, o simplemente escoger lo que viene por defecto en tu editor de código . A partir de ahí, sugiero que te quedes con esa configuración en solo tus **proyectos personales**. Esto es porque cuando se comienza a trabajar en equipo se suele llegar a un acuerdo entre todos, y la configuración se vuelve única para todo el equipo.

No a la divitis, sí a la semántica

Divitis es un término que se ha creado para indicar que un desarrollador está abusando del uso de las etiquetas `div` y `span` como contenedores genéricos en lugar de usar elementos estructurales y semánticos de HTML.

Antes de profundizar, debemos saber que las etiquetas `div` son elementos de agrupación, se supone que están hechos para agrupar las secciones de una página en divisiones. Por ejemplo, una división en la parte superior como un header con un logo y unos links, un sidebar o hasta un footer. Los `divs` son elementos de agrupación con ningún estilo por defecto. No tienen margin, padding o border, el único estilo por defecto es que es un elemento de tipo bloque. `display: block;`

Si bien se puede maquetar una página usando `divs` y `spans`, esto es considerado una mala práctica, no es escalable y se vuelve una tortura para los desarrolladores al tener tantos `divs` y `spans` en un archivo. Afortunadamente, desde HTML5 han aparecido etiquetas con semántica, estos tienen un significado, una razón del porqué están ahí. Así, se vuelve sencillo para nosotros el poder diferenciar cada bloque de código.

Docs_ Generalidades Curiosidades

Mencionando algunos elementos semánticos de HTML5 son:

- `<main></main>` indica el contenido principal y el más importante de una página web.
- `<header></header>` es el encabezado de una página, es donde se ubicará el logo y links principales
- `<section></section>` sirve para indicar qué contenido corresponde a qué parte de un esquema
- `<article></article>` funciona para contener información de entradas de blog, mensajes de foro o artículos de revistas.
- `<footer></footer>` es el pie de página, es la última parte de toda web. Se ubican los links restantes, sitios de mapas y redes sociales.

Continuando con nuestro bloque HTML, el reto es reemplazar todos los divs del código por etiquetas semánticas. Colocaré la solución en la parte inferior. Aún así, no dudes en intentarlo!

```
<!DOCTYPE html>
<html>

<head>
  <title>Tu Blog</title>
</head>

<body style="margin: 0">
  <section
    style="display: grid; height: 100vh; grid-gap: 5px; grid-template-columns:
repeat(2, 1fr); grid-template-rows: 100px 90px 1fr 90px; grid-template-areas: 'header
header' 'nav nav' 'aside main' 'footer footer';">

    <header
      style="display: flex; justify-content: center; align-items: center; font-
size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: header;
background-color: lightblue;">
      HEADER
    </header>

    <nav
      style="display: flex; justify-content: center; align-items: center; font-
size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: nav;
background-color: #cccc;">
      NAV
    </nav>
```

Docs_ Generalidades Curiosidades

```
<aside
  style="display: flex; justify-content: center; align-items: center; font-
size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: aside;
background-color: yellow;">
  ASIDE
</aside>

<main
  style="display: flex; justify-content: center; align-items: center; font-
size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: main;
background-color: orange;">
  CONTENT
</main>

<footer
  style="display: flex; justify-content: center; align-items: center; font-
size: 2em; font-weight: 700; font-family: 'arial'; color: #c0c0c0; grid-area: footer;
background-color: peru;">
  FOOTER
</footer>

</section>
</body>
</html>
```

Separa el CSS y Javascript del HTML

Existen tres formas de aplicar estilos al HTML, estos son en línea, dentro de una etiqueta `<style></style>` y en archivos CSS externos. La mejor práctica es mover nuestros estilos a archivos externos, así dejamos que cada archivo que tengamos tome **una única responsabilidad**. Nuestros archivos .html harán la estructura y el marcado, nuestros archivos .css aplicarán los estilos y adornarán la página, y los archivos .js harán que exista interacciones en nuestra página.

Docs_ Generalidades Curiosidades

Para terminar nuestro bloque de HTML, el reto es mover nuestros estilos a archivos externos, y aplicar clases a nuestros elementos. Atrévete!

HTML

```
<!DOCTYPE html>
<html>

<head>
  <title>Tu Blog</title>
  <link rel="stylesheet" href="estilos.css">
</head>

<body>
  <section class="layout">

    <header class="container header">
      HEADER
    </header>

    <nav class="container nav">
      NAV
    </nav>

    <aside class="container aside">
      ASIDE
    </aside>

    <main class="container main">
      CONTENT
    </main>

    <footer class="container footer">
      FOOTER
    </footer>

  </section>
</body>
</html>
```


Docs_ Generalidades Curiosidades

CCS

```
body {  
  margin: 0;  
}  
  
.layout {  
  display: grid;  
  height: 100vh;  
  grid-gap: 5px;  
  grid-template-columns: repeat(2, 1fr);  
  grid-template-rows: 100px 90px 1fr 90px;  
  grid-template-areas: 'header header' 'nav nav' 'aside main' 'footer footer';  
}  
  
.container {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  font-size: 2em;  
  font-weight: 700;  
  font-family: 'arial';  
  color: #c0c0c0;  
}  
  
.header {  
  grid-area: header;  
  background-color: lightblue;  
}  
  
.nav {  
  grid-area: nav;  
  background-color: #cccc;  
}  
  
.aside {  
  grid-area: aside;  
  background-color: yellow;  
}  
  
.main {  
  grid-area: main;  
  background-color: orange;  
}
```

Docs_ Generalidades Curiosidades

```
.footer {  
  grid-area: footer;  
  background-color: peru;  
}
```

Sé consistente

Lo dicho, **en la práctica está la maestría**. Ser consistente con cada tip aprendido aquí te ahorrará un montón de problemas, cada buena práctica aplicada te ayudará o al equipo en el que estés, porque escribirás código escalable y fácil de leer. Por ello, sigue practicando, práctica, nunca te rindas y nunca pares de aprender