

Reglas de estilo de Google para HTML

1. Reglas generales

1.1 Reglas de estilo generales

1.1.1 Protocolo

Utiliza direcciones HTTPS siempre que sea posible para agregar recursos, y no omitas el protocolo al escribir una dirección.

1.2 Reglas de formato generales

1.2.1 Indentación

Cada nivel de indentación supone dos espacios.

No usar tabuladores o mezclar tabuladores y espacios para indentar.

Por defecto, VSCode corrige este tipo de errores automáticamente.

1.2.2 Capitalización (*Mayúsculas*)

Utiliza sólo minúsculas para escribir el código HTML y CSS (etiquetas, atributos, valores, etc.).

1.2.3 Espacio en blanco finales

Elimina todos los espacios en blanco innecesarios en el documento y al final de la línea.

1.3 Reglas generales de metadatos

1.3.1 Codificación (encoding)

Utiliza UTF-8. Asegurate de que tu editor usa la codificación de caracteres UTF-8.

Especifica la codificación en los documentos HTML mediante la etiqueta `<meta charset="utf-8">`.

1.3.2 Comentarios

Es interesante introducir explicaciones del código, siempre que sea posible y facilite su comprensión.

No es realista querer un documento perfectamente documentado. El volumen de trabajo o su variabilidad pueden hacerlos contraproducentes. Es importante valorar cuando merece la pena documentar.

1.3.3 Elementos pendientes

Utiliza el término `TODO` para especificar una tarea por hacer en el documento. En documentos colaborativos se puede incluir entre paréntesis el nombre del usuario que pretende emprender esa acción.

2 HTML

2.1 Reglas de estilo HTML

2.1.1 Tipo de documento

Usa HTML5. Esto se indica al principio de un documento html mediante el elemento `<!DOCTYPE html>`.

Aunque está bien cerrar todas las etiquetas HTML, no cierre las de elementos vacíos (Ej: `
`, en lugar de `
`)

2.1.2 Validación de HTML

Procura que tu código HTML sea válido siempre que sea posible. Hacerlo es una base de calidad del código que contribuye al uso apropiado de HTML y su aprendizaje.

Para ello podemos usar utilidades web, como el [W3C HTML validator](#).

2.1.3 Semántica

Usa HTML de acuerdo a su propósito. Usa las etiquetas respectivamente para lo que han sido diseñadas (Ej: `p` para párrafos, `a` para anclar un enlace, etc.).

Esto facilita la accesibilidad, eficiencia y reusabilidad del código.

2.1.4 Accesibilidad multimedia

Provee contenido alternativo para los elementos multimedia. Tiene importantes implicaciones en materia de accesibilidad. Incluir texto alternativo para imágenes y videos determinantes en el contexto de nuestro documento es una práctica excelente. Sin embargo, se desaconseja incluir contexto en imágenes que solo tiene propósitos decorativos y no aportan contexto.

2.1.5 Separacion de componentes

Es vital para un documento web separar la estructura de su presentación, así como de su comportamiento. Mantén estrictamente separadas la estructura (HTML), el estilo (CSS) y el comportamiento (scripting) y procura que sus combinaciones sean absolutamente mínimas.

Mantener el contacto de entre ellos al mínimo, enlazando los elementos mínimos para conseguir el estilo y comportamiento que buscamos.

2.1.6 Caracteres referencia

No usar caracteres referencia.

Gracias al uso de la codificación UTF-8, no hay necesidad de usar este tipo de caracteres, mejorando la legibilidad del código, salvo excepciones como `<` ó `&`.

2.1.7 Etiquetas opcionales

Por motivos de optimización, comienza a aceptarse la omisión de ciertas etiquetas. Esta es una tendencia de estilo reciente, y puramente opcional, pero es interesante tener conocimiento de ella. La [especificación de HTML5](#) define qué etiquetas pueden ser omitidas

2.1.8 Atributos

Omitir el atributo `type` al hacer referencias a hojas de estilo y scripts (Por defecto, las etiquetas `link` y `script` implican por defecto los tipos `text/css` y `text/javascript`, respectivamente).

2.2 Reglas de formato HTML

2.2.1 Formato general

Usa una nueva línea para cada bloque, lista o elemento de una tabla, e indenta a cada hijo para que cuelguen de éste. En caso de que los elementos sean pequeños y su indentación provoque mucho espacio en blanco, está contemplado poner dichos elementos en una sola línea.

2.2.2 "Enrollar" las líneas de HTML

A pesar de no existir un límite de longitud de línea en HTML, se recomienda romper las líneas largas *enrollándolas* para facilitar su lectura.

Cuando se envuelve una línea, su continuación en la línea siguiente debe estar indentada al menos 4 espacios adicionales (2 indentaciones de HTML) con respecto a su línea original.

2.3.2 Comillas en HTML

Usa comillas dobles (`"`) en lugar de comillas simples (`'`) para los atributos y valores.