Estandar de programacion

Descripcion

Este documento define el formato y las reglas de estilo para HTML y CSS. Su objetivo es mejorar la colaboración, la calidad del código y habilitar la infraestructura de soporte. Se aplica a los archivos en bruto que funcionan que utilizan HTML y CSS, incluidos los archivos GSS. Las herramientas son libres de ofuscar, minimizar y compilar siempre que se mantenga la calidad general del código. 2 General

Reglas generales

2.1.1 Protocolos

Use HTTPS para recursos incrustados cuando sea posible.

Siempre use HTTPS (https :) para imágenes y otros archivos multimedia, hojas de estilo y scripts, a menos que los archivos respectivos no estén disponibles a través de HTTPS.

<! - No recomendado: omite el protocolo -> <script src = "// ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"> </script> <! - - No recomendado: utiliza HTTP -> <script src = "http://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"> </script> <! - Recommended -> <script src = "https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"> </script> / \* No recomendado: omite el protocolo \* / @ importe '//fonts.googleapis.com/css?family=Open+Sans'; / \* No recomendado: utiliza HTTP \* / @import 'http://fonts.googleapis.com/css?family=Open+Sans';/\* Recomendado \* / @import' https://fonts.googleapis.com/css ? family = Open + Sans ';

2.2 Reglas generales de formato

2.2.1 Indentacion

Sangre por 2 espacios a la vez.

No uses pestañas o combina tablas y espacios para la sangría.

<ul> <li> Fantástico <li> Genial </ul> .example {color: blue; }

2.2.2 Capitalizacion

Utilice solo minúsculas.

Todo el código debe estar en minúsculas: esto se aplica a los nombres de elementos HTML, atributos, valores de atributos (a menos que sea texto / CDATA), selectores de CSS, propiedades y valores de propiedad (con la excepción de cadenas).

<! - No recomendado -> <A HREF="/"> Inicio </A> <! - Recomendado -> <img src = "google.png" alt = "Google"> / \* No recomendado \* / color: # E5E5E5; / \* Recomendado \* / color: # e5e5e5;

Use only lowercase.

All code has to be lowercase: This applies to HTML element names, attributes, attribute values (unless text/CDATA), CSS selectors, properties, and property values (with the exception of strings).

<!-- Not recommended --><A HREF="/">Home</A><!-- Recommended --><img src="google.png" alt="Google">/\* Not recommended \*/color: #E5E5E5;/\* Recommended \*/color: #e5e5e5;

2.2.3 Espacios en blanco

Eliminar los espacios en blanco al final.

Los espacios en blanco traseros son innecesarios y pueden complicar diferencias.

<! - No recomendado -> <p> ¿Qué? \_ <! - Recomendado -> <p> Sí, por favor.

2.3 Reglas generales meta

2.3.1 Codificacion

Utilice UTF-8 (sin BOM).

Asegúrese de que su editor use UTF-8 como codificación de caracteres, sin una marca de orden de bytes.

Especifique la codificación en plantillas y documentos HTML a través de <meta charset = "utf-8">. No especifique la codificación de las hojas de estilo, ya que éstas suponen UTF-8.

(Puede encontrar más información sobre codificaciones y cuándo y cómo especificarlas en Manejo de codificaciones de caracteres en HTML y CSS).

2.3.2 Comentarios

Explique el código según sea necesario, cuando sea posible.

Use los comentarios para explicar el código: ¿Qué cubre, para qué sirve, por qué se usa o prefiere la solución respectiva?

(Este elemento es opcional, ya que no se considera una expectativa realista de exigir siempre un código totalmente documentado. El kilometraje puede variar mucho para el código HTML y CSS y depende de la complejidad del proyecto).

2.3.3 Elementos de accion

3 HTML

Marca todos los elementos de acción y con TODO.

Resalte todos usando solo la palabra clave TODO, no otros formatos comunes como @@.

Agregue un contacto (nombre de usuario o lista de correo) entre paréntesis como con el formato TODO (contacto).

Agregue elementos de acción después de dos puntos como en TODO: elemento de acción.

{# TODO (john.doe): revisit centering #} <center> Test </center> <! - TODO: eliminar las etiquetas opcionales -> <ul> <li> Apples </li> <li> Oranges </ li> </ul>

3.1 HTML Reglas de estilo

3.1.1 Tipos de documentos

Usa HTML5.

Se prefiere HTML5 (sintaxis HTML) para todos los documentos HTML: <! DOCTYPE html>.

(Se recomienda usar HTML, como texto / html. No use XHTML. XHTML, como application / xhtml + xml, carece de soporte de navegador e infraestructura y ofrece menos espacio para la optimización que HTML.)

Aunque está bien con HTML, no cierre los elementos vacíos, es decir, escriba <br>, no <br />.

3.1.2 HTML Validacion

Utilice HTML válido siempre que sea posible.

Utilice un código HTML válido a menos que no sea posible debido a los objetivos de rendimiento inalcanzables del tamaño del archivo.

Utilice herramientas como el validador de HTML W3C para probar.

El uso de HTML válido es un atributo de calidad de línea de base medible que contribuye a conocer los requisitos técnicos y las restricciones, y que garantiza el uso correcto de HTML.

<! - No recomendado -> <title> Prueba </title> <article> Esto es solo una prueba. <! - Recomendado -> <! DOCTYPE html> <meta charset = "utf-8"> < title> Test </title> <article> Esto es solo una prueba. </article>

3.1.3 Semantica

Utilizar HTML de acuerdo a su finalidad

Use elementos (a veces llamados incorrectamente "etiquetas") para lo que se han creado. Por ejemplo, use elementos de encabezado para encabezados, p elementos para párrafos, elementos para anclajes, etc.

El uso de HTML de acuerdo con su propósito es importante por razones de accesibilidad, reutilización y eficiencia de código.

<! - No recomendado -> <div onclick = "goToRecommendations ();"> Todas las recomendaciones </div> <! - Recomendado -> <a href="recommendations/"> Todas las recomendaciones </a>

3.1.4 Multimedia

Proporcionar contenidos alternativos para multimedia.

Para multimedia, como imágenes, videos, objetos animados a través de lienzos, asegúrese de ofrecer un acceso alternativo. Para imágenes que significa el uso de texto alternativo significativo (alt) y para transcripciones y subtítulos de video y audio, si están disponibles.

Proporcionar contenidos alternativos es importante por razones de accesibilidad: un usuario ciego tiene pocas pistas para decir de qué se trata una imagen sin @alt, y es posible que otros usuarios no tengan forma de entender de qué se trata el contenido de video o audio.

(Para imágenes cuyos atributos alt introducirían redundancia, y para imágenes cuyo propósito es puramente decorativo para el que no puede usar CSS de inmediato, no utilice un texto alternativo, como en alt = "".)

<! - No recomendado -> <img src = "spreadsheet.png"> <! - Recomendado -> <img src = "spreadsheet.png" alt = "Captura de pantalla de la hoja de cálculo.">

3.1.5 Separación de intereses

Estructura separada de la presentación del comportamiento.

Mantenga estrictamente la estructura (marcado), la presentación (estilo) y el comportamiento (secuencias de comandos), y trate de mantener la interacción entre los tres en un mínimo absoluto.

Es decir, asegúrese de que los documentos y las plantillas contengan solo HTML y HTML que tenga únicamente fines estructurales. Mueva todo lo que se presenta a las hojas de estilo, y todo el comportamiento a los guiones.

Además, mantenga el área de contacto lo más pequeña posible al vincular la menor cantidad posible de hojas de estilo y scripts de documentos y plantillas.

Separar la estructura de la presentación del comportamiento es importante por razones de mantenimiento. Siempre es más caro cambiar las plantillas y documentos HTML que actualizar las hojas de estilo y los scripts.

<! - No recomendado -> <! DOCTYPE html> <título> HTML apesta </title> <link rel = "hoja de estilo" href = "base.css" media = "screen"> <link rel = "hoja de estilo" href = "grid.css" media = "screen"> <link rel = "hoja de estilo" href = "print.css" media = "print"> <h1 style = "fuente-tamaño: 1em;"> HTML apesta </ h1> <p> He leído sobre esto en algunos sitios, pero ahora estoy seguro: <u> ¡¡HTML es estúpido !! 1 </u> <center> No puedo creer que no haya forma de controlar el estilo ¡de mi sitio web sin hacer todo de nuevo! </center> <! - Recommended -> <! DOCTYPE html> <title> Mi primer rediseño solo para CSS </title> <link rel = "hoja de estilo" href = " default.css "> <h1> Mi primer rediseño solo para CSS </h1> <p> He leído sobre esto en algunos sitios, pero hoy lo estoy haciendo: separando preocupaciones y evitando cualquier cosa en el HTML de mi sitio web que es de presentación. <p> ¡Es increíble!

3.1.6 Referencia identidad

No utilice referencias de entidad.

No es necesario utilizar referencias de entidad como & mdash ;, & rdquo ;, o & # x263a ;, suponiendo que la misma codificación (UTF-8) se usa para archivos y editores, así como entre equipos.

Las únicas excepciones se aplican a los caracteres con un significado especial en HTML (como <y &), así como a los caracteres de control o "invisibles" (como los espacios sin interrupción).

<! - No recomendado -> El símbolo de moneda para el euro es & ldquo; & eur; & rdquo;. <! - Recomendado -> El símbolo de moneda para el euro es "€".

3.1.7 Etiquetas opcionales

Omitir etiquetas opcionales (opcional).

Para fines de optimización y capacidad de escaneo de archivos, considere omitir etiquetas opcionales. La especificación HTML5 define qué etiquetas pueden omitirse.

(Este enfoque puede requerir que se establezca un período de gracia como una guía más amplia, ya que es significativamente diferente de lo que se enseña normalmente a los desarrolladores web. Por razones de coherencia y simplicidad, es mejor omitir todas las etiquetas opcionales, no solo una selección).

<! - No recomendado -> <! DOCTYPE html> <html> <head> <title> Gastar dinero, gastar bytes </title> </head> <body> <p> Sic. </p> </ body> </html> <! - Recommended -> <! DOCTYPE html> <title> Ahorrando dinero, salvando bytes </title> <p> .

3.1.8 Tipos de atributo

Omitir atributos de tipo para hojas de estilo y scripts.

No use atributos de tipo para hojas de estilo (a menos que no use CSS) y scripts (a menos que no use JavaScript).

No es necesario especificar atributos de tipo en estos contextos ya que HTML5 implica texto / css y texto / javascript como valores predeterminados. Esto se puede hacer de forma segura incluso para navegadores más antiguos.

<! - No recomendado -> <link rel = "hoja de estilo" href = "https://www.google.com/css/maia.css" type = "text / css"> <! - Recommended - > <link rel = "hoja de estilo" href = "https://www.google.com/css/maia.css"> <! - No recomendado -> <script src = "https://www.google. com / js / gweb / analytics / autotrack.js "type =" text / javascript "> </script> <! - Recommended -> <script src =" https://www.google.com/js/gweb /analytics/autotrack.js "> </script>

3.2 HTML Reglas de formato

3.2.1 Formato general

Use una nueva línea para cada bloque, lista o elemento de tabla, y guarde cada elemento secundario.

Independientemente del estilo de un elemento (como CSS permite a los elementos asumir una función diferente por propiedad de visualización), coloque cada bloque, lista o elemento de tabla en una nueva línea.

Además, hágalos sangrar si son elementos secundarios de un elemento de bloque, lista o tabla.

(Si tiene problemas con los espacios en blanco entre los elementos de la lista, es aceptable colocar todos los elementos de li en una línea. Se recomienda que el indicador coloque una advertencia en lugar de un error).

<blockquote> <p> <em> Space </em>, la última frontera. </p> </blockquote> <ul> <li> Moe <li> Larry <li> Curly </ul> <table> < thead> <tr> <th scope = "col"> Ingresos <th scope = "col"> Impuestos <tbody> <tr> <td> $ 5.00 <td> $ 4.50 </table>

3.2.2 HTML Envoltura de lineas

Romper líneas largas (opcional).

Si bien no hay una recomendación de límite de columna para HTML, puede considerar el ajuste de líneas largas si mejora significativamente la legibilidad.

Cuando se ajusta la línea, cada línea de continuación debe estar sangrada al menos 4 espacios adicionales de la línea original.

<md-progress-circular md-mode = "indeterminate" class = "md-accent" ng-show = "ctrl.loading" md-diámetro = "35"> </md-progress-circular> <md-progress- circular md-mode = "indeterminado" class = "md-accent" ng-show = "ctrl.loading" md-diámetro = "35"> </md-progress-circular> <md-progress-circular md-mode = "indeterminado" clase = "md-accent" ng-show = "ctrl.loading" md-diámetro = "35"> </md-progress-circular>

3.2.3 HTML Citas

Al citar valores de atributos, utilice comillas dobles.

Utilice comillas dobles ("") en lugar de comillas simples ('') alrededor de los valores de los atributos

<! - No recomendado -> <a class='maia-button maia-button-secondary'> Iniciar sesión </a> <! - Recomendado -> <a class = "maia-button maia-button- secundaria "> Iniciar sesión </a>

4 CSS

4.1 CSS Estilo CSS

4.1.1 CSS Validacion

Use CSS válido cuando sea posible

A menos que se trate de errores de validación de CSS o que requiera una sintaxis de propiedad, use un código CSS válido.

Utilice herramientas como el validador de CSS del W3C para probar.

El uso de CSS válido es un atributo de calidad de línea de base medible que permite detectar el código CSS que puede no tener ningún efecto y puede eliminarse, y que garantiza el uso adecuado de CSS.

4.1.2 Nombre de ID y Class

Utilice nombres de clase y de identificación significativos o genéricos.

En lugar de nombres de presentación o crípticos, siempre use nombres de identificación y de clase que reflejen el propósito del elemento en cuestión, o que sean genéricos.

Los nombres que son específicos y que reflejan el propósito del elemento deben ser preferidos ya que estos son más comprensibles y es menos probable que cambien.

Los nombres genéricos son simplemente una alternativa para los elementos que no tienen un significado particular o diferente al de sus hermanos. Normalmente se necesitan como "ayudantes".

El uso de nombres funcionales o genéricos reduce la probabilidad de cambios innecesarios de documentos o plantillas.

/ \* No recomendado: sin significado \* / # yee-1901 {} / \* No recomendado: presentación \* / .button-green {} .clear {} / \* Recomendado: específico \* / #gallery {} #login {} .video { } / \* Recomendado: genérico \* / .aux {} .alt {}

4.1.3 Estilo de nombres y clases

Use nombres de identificación y clase que sean lo más cortos posible pero tan largos como sea necesario.

Trate de transmitir de qué se trata una identificación o clase al tiempo que es lo más breve posible.

El uso de ID y nombres de clase de esta manera contribuye a niveles aceptables de comprensión y eficiencia de código.

/ \* No recomendado \* / #navigation {} .atr {} / \* Recomendado \* / #nav {} .author {}

4.1.4 Tipo de selectors

Evite los ID de clase y los nombres de clase con selectores de tipo

A menos que sea necesario (por ejemplo, con clases auxiliares), no utilice nombres de elementos junto con ID o clases.

Evitar los selectores de antepasados innecesarios es útil por razones de rendimiento.

/ \* No recomendado \* / ul # example {} div.error {} / \* Recomendado \* / #example {} .error {}

4.1.5 Propiedades ShortHand

Use las propiedades de taquigrafía cuando sea posible.

CSS ofrece una variedad de propiedades abreviadas (como fuente) que deben usarse siempre que sea posible, incluso en los casos en que solo se establece explícitamente un valor.

El uso de propiedades abreviadas es útil para la eficiencia y la comprensibilidad del código.

/ \* No recomendado \* / border-top-style: none; Familia tipográfica: palatino, georgia, serif; tamaño de letra: 100%; línea de altura: 1.6; parte inferior de relleno: 2em; relleno-izquierda: 1em; derecho de relleno: 1em; padding-top: 0; / \* Recomendado \* / border-top: 0; fuente: 100% / 1.6 palatino, georgia, serif; relleno: 0 1em 2em;

4.1.6 0 y unidades

Omita la especificación de la unidad después de los valores de “0”, a menos que sea necesario.

No use unidades después de 0 valores a menos que sean requeridos.

flex: 0px; / \* Este componente de base flexible requiere una unidad. \* / flex: 1 1 0px; / \* No es ambiguo sin la unidad, pero es necesario en IE11. \* / margen: 0; relleno: 0;

4.1.7 0s

Omitir "0" en los valores.

No ponga 0s delante de valores o longitudes entre -1 y 1.

tamaño de letra: .8em;

4.1.8 Notacion hexadecimal

Utilice la notación hexadecimal de 3 caracteres cuando sea posible.

Para los valores de color que lo permiten, la notación hexadecimal de 3 caracteres es más breve y concisa.

/ \* No recomendado \* / color: #eebbcc; / \* Recomendado \* / color: #ebc;

4.1.9 Prefijos

Selectores de prefijo con un prefijo específico de la aplicación (opcional).

En los proyectos grandes, así como para el código que se incrusta en otros proyectos o en sitios externos, use los prefijos (como espacios de nombres) para los nombres de identificación y clase. Utilice identificadores cortos y únicos seguidos de un guión.

El uso de espacios de nombres ayuda a evitar conflictos de nombres y puede facilitar el mantenimiento, por ejemplo, en las operaciones de búsqueda y reemplazo.

.adw-help {} / \* AdWords \* / # maia-note {} / \* Maia \* /

4.1.10 Delimitacion de IDs y clases

Separe las palabras en ID y nombres de clase por un guión.

No concatene palabras y abreviaturas en los selectores por ningún carácter (incluyendo ninguno) excepto guiones, para mejorar la comprensión y la capacidad de exploración.

/ \* No recomendado: no separa las palabras "demo" e "imagen" \* / .demoimage {} / \* No recomendado: usa guión bajo en lugar de guión \* / .error\_estado {} / \* Recomendado \* / # video-id {} .ads-sample {}

4.1.11 Hacks

Evite la detección de agente de usuario así como los "hacks" de CSS: primero pruebe un enfoque diferente.

Es tentador abordar las diferencias de estilo en la detección de agente de usuario o los filtros especiales de CSS, las soluciones y los hacks. Ambos enfoques deben considerarse como último recurso para lograr y mantener una base de código eficiente y manejable. Dicho de otra manera, otorgar a la detección y los cortes un pase libre dañará los proyectos a largo plazo, ya que los proyectos tienden a tomar el camino de menor resistencia. Es decir, permitir y facilitar el uso de la detección y piratear significa utilizar la detección y piratear con mayor frecuencia, y con más frecuencia es demasiado frecuente.

4.2 CSS Reglas de formato

4.2.1 Orden de declaracion

Alfabetizar declaraciones.

Ponga las declaraciones en orden alfabético para lograr un código consistente de manera que sea fácil de recordar y mantener.

Ignore los prefijos específicos del proveedor para propósitos de clasificación. Sin embargo, deben mantenerse ordenados los múltiples prefijos específicos del proveedor para una determinada propiedad CSS (por ejemplo, el prefijo -moz viene antes de -webkit).

fondo: fucsia; borde: 1px sólido; -moz-border-radius: 4px; -webkit-border-radius: 4px; radio del borde: 4px; de color negro; text-align: center; texto-sangría: 2em;

4.2.2 Identacion en bloques

Indenta todo el contenido del bloque.

Aplique sangría a todo el contenido del bloque, es decir, reglas dentro de las reglas y declaraciones, para reflejar la jerarquía y mejorar la comprensión.

Pantalla de @media, proyección {html {background: #fff; color: # 444; }}

4.2.3 Declaracion de altos

Use un punto y coma después de cada declaración.

Termine cada declaración con un punto y coma por razones de consistencia y extensibilidad.

/ \* No recomendado \* / .test {display: block; altura: 100px} / \* Recomendado \* / .test {display: block; altura: 100px; }

4.2.4 Propiedad de nombre de altos

Use un espacio después de los dos puntos del nombre de una propiedad.

Siempre use un solo espacio entre la propiedad y el valor (pero no el espacio entre la propiedad y los dos puntos) por razones de coherencia.

/ \* No recomendado \* / h3 {font-weight: negrita; } / \* Recomendado \* / h3 {font-weight: negrita; }

4.2.5 Declaracion de separacion de bloques

Use un espacio entre el último selector y el bloque de declaración.

Siempre use un solo espacio entre el último selector y la llave de apertura que comienza el bloque de declaración.

El refuerzo de apertura debe estar en la misma línea que el último selector en una regla dada.

/ \* No recomendado: falta espacio \* / #video {margin-top: 1em; } / \* No recomendado: salto de línea innecesario \* / #video {margin-top: 1em; } / \* Recomendado \* / #video {margin-top: 1em; }

4.2.6 Separacion de declaracion y seleccion

Separadores y declaraciones separados por nuevas líneas.

Siempre comience una nueva línea para cada selector y declaración.

/ \* No recomendado \* / a: foco, a: activo {posición: relativo; superior: 1px; } / \* Recomendado \* / h1, h2, h3 {font-weight: normal; altura de la línea: 1.2; }

4.2.7 Reglas de separacion

Reglas separadas por nuevas líneas.

Siempre ponga una línea en blanco (dos saltos de línea) entre las reglas.

html {fondo: #fff; } cuerpo {margen: auto; ancho: 50%; }

4.2.8 CSS Citas

Utilice comillas simples ('') en lugar de dobles ("") para los selectores de atributos y valores de propiedad.

No utilice comillas en los valores de URI (url ()).

Excepción: si necesita usar la regla @charset, use comillas dobles, no se permiten comillas simples.

/ \* No recomendado \* / @import url ("https://www.google.com/css/maia.css"); html {font-family: "open sans", arial, sans-serif; } / \* Recomendado \* / @import url (https://www.google.com/css/maia.css); html {font-family: 'open sans', arial, sans-serif; }

4.3 CSS Reglas meta

4.3.1 Seccion de comentarios

Agrupar secciones por un comentario de sección (opcional).

Si es posible, agrupe las secciones de la hoja de estilo usando comentarios. Secciones separadas con nuevas líneas.

/ \* Encabezado \* /

# adw-header {}

/ \* Pie de página \* /

# adw-footer {}

/ \* Galería \* /

.adw-gallery {}