Informes mejores y más rápidos



Publicado el 28 de abril de 2023 por [Editor](https://www.xbrl.org/news/better-faster-reports/)

El grupo de trabajo de especificaciones XBRL ha actualizado su nota de grupo de trabajo (WGN) sobre el uso de HTML en los informes XBRL en línea, incluida información adicional sobre la preparación de etiquetas de bloque de texto.

Este WGN incorpora orientación adicional para desarrolladores de software y usuarios técnicos involucrados en la preparación de informes XBRL en línea.

Inline XBRL usa HTML para su presentación y WGN, publicado por primera vez en diciembre de 2022, proporciona una guía mejorada para informes digitales estructurados de alto rendimiento y fácil consumo.

El WGN ya incluye sugerencias que se pueden incluir para mejorar la experiencia del usuario al usar el software Inline XBRL Viewer y orientación sobre la selección de etiquetas HTML.

Ahora proporciona orientación adicional sobre la compatibilidad con HTML e información adicional sobre la preparación de etiquetas de bloque de texto.

La lectura esencial para los desarrolladores de software Inline XRBL se puede encontrar  [aquí](https://xbrl.us3.list-manage.com/track/click?u=fbbc983fbc59adb89ff279b5b&id=af54ac86d9&e=45983b0199) .

[HTML](https://www.xbrl.org/tag/html/) [SPEC](https://www.xbrl.org/tag/spec/) [XII NOTICIAS](https://www.xbrl.org/tag/xii/)





**1. Información general**

Este documento proporciona una serie de recomendaciones para la construcción de HTML para su uso en informes [XBRL en línea](https://www.xbrl.org/specification/inlinexbrl-part1/rec-2013-11-18/inlinexbrl-part1-rec-2013-11-18.html). Estas recomendaciones están destinadas a mejorar la compatibilidad y el funcionamiento con el software de visualización Inline XBRL, garantizar la calidad de los datos XBRL extraídos y mejorar el rendimiento.

**2. Nombres de clase CSS reservados**

El software de visualización Inline XBRL puede aplicar clases de CSS a un informe Inline XBRL renderizado para habilitar el resaltado de hechos y otras funciones. Para evitar colisiones entre las clases de CSS utilizadas por el software de visualización y las clases de CSS utilizadas para diseñar el informe Inline XBRL, los nombres de clase que comienzan con -ixv- deben considerarse reservados para su uso por parte del software de visualización Inline XBRL y no deben utilizarse en un informe Inline XBRL. o cualquier hoja de estilo adjunta.

Si el software del visor Inline XBRL necesita agregar clases CSS a un informe Inline XBRL, debe asegurarse de que todas esas clases tengan el prefijo -ixv-, por ejemplo, -ixv-selected-fact.

Este documento también propone una clase de CSS que se puede usar para proporcionar una pista sobre el software del visor XBRL en línea (consulte la [Sección 8](https://www.xbrl.org/WGN/html-for-ixbrl-wgn/WGN-2023-04-19/html-for-ixbrl-wgn-2023-04-19.html#sec-hint-container)). Esto usa un prefijo de -ixh-. El prefijo -ixh-también debe considerarse reservado y las clases de CSS que comienzan con este prefijo no deben usarse. en los informes Inline XBRL o cualquier hoja de estilo que lo acompañe, que no sea como se describe en este o en los documentos publicados por XBRL International.

**3. HTML eficiencia**

El HTML en los informes XBRL en línea puede ser muy grande, especialmente cuando se genera mediante un proceso de conversión de PDF a HTML, y esto puede provocar una carga y un rendimiento de procesamiento lentos. Esto puede ser particularmente problemático para el software de visualización Inline XBRL, ya que dicho software generalmente requerirá que el informe se represente por completo antes de que el software pueda ser completamente funcional.

La mejora del rendimiento de la representación en los informes XBRL en línea se trata con más detalle en la [Nota del grupo de trabajo sobre el rendimiento de la representación en XBRL en línea](https://www.xbrl.org/WGN/ixbrl-rendering-performance-wgn/WGN-2021-10-27/ixbrl-rendering-performance-wgn-2021-10-27.html). Esto incluye una recomendación para usar la content-visibility: autopropiedad CSS que puede mejorar sustancialmente el rendimiento de representación de dichos documentos en algunos navegadores.

**4. Selección de etiquetas HTML**

La especificación Inline XBRL no impone ninguna regla sobre la elección de las etiquetas HTML utilizadas para lograr un resultado de diseño particular. Por ejemplo, no es necesario marcar los datos tabulares con etiquetas y, o que los encabezados usen etiquetas de encabezado HTML (, , <table>, etc.). Estas funciones se pueden etiquetar mediante etiquetas genéricas como o y aplicando el estilo apropiado, o mediante cualquier otro enfoque compatible con HTML. <tr><td><h1><h2><h3><div><span>

El uso de etiquetas HTML más específicas puede mejorar la facilidad de uso y la accesibilidad de los documentos HTML y, por lo tanto, de los informes Inline XBRL, pero la especificación Inline XBRL no lo requiere.

**5. Etiquetas de bloque de texto**

Inline XBRL proporciona un mecanismo para incluir marcado HTML del documento de origen en un valor de hecho XBRL. Este mecanismo se habilita estableciendo el valor del escape atributo en un ix:nonNumericelemento en true. Cuando está habilitado, cualquier etiqueta HTML que aparezca dentro de la etiqueta XBRL en línea se incluirá en el valor de hecho resultante. El valor de hecho resultante será un fragmento XHTML válido.

Si se establece en false, las etiquetas HTML no se incluyen en la salida; solo el contenido de texto se incluye en la salida. El valor de hecho resultante será texto sin formato.

Por ejemplo, considere una ix:nonNumericetiqueta alrededor del siguiente contenido:

**<b>**Bold text**</b>**

Con escape="false", el valor de hecho resultante será " Bold text".

Con escape="true", el valor de hecho incluirá las etiquetas HTML, es decir, " <b>Bold text</b>".

La forma en que se interpreta un valor de hecho está determinada por su tipo de datos. Los hechos que están destinados a contener XHTML normalmente usarán dtr-types:textBlockItemType. El valor de un hecho que usa dtr-types:textBlockItemTypeDEBE ser XHTML válido, por lo que las etiquetas XBRL en línea para tales hechos deben usarescape="true"

Si escape="false"se utiliza, los caracteres especiales que aparecen en el texto del hecho, como <, >y &no se escaparán de XML y, como resultado, es posible que el valor del hecho resultante no sea XHTML válido.

El uso de escape="true"hechos que no se espera que contengan XHTML, como xbrli:stringItemType, dará lugar a un marcado inesperado en el valor del hecho resultante, lo que probablemente provocará que las etiquetas XHTML se muestren directamente al usuario cuando vea dichos hechos.

Por lo tanto, es importante que el valor del escapeatributo coincida con el tipo de datos del hecho:

* Los hechos con un tipo de datos de dtr-types:textBlockItemTypedeben usar escape="true".
* Hechos con otros tipos de datos, como xbrli:stringItemTypedebería usar escape="false".

**5.1 Uso de escape="false" en etiquetas de bloque de texto**

Si el contenido de texto de una etiqueta Inline XBRL no contiene caracteres que deban escaparse en XML (<y &), entonces se puede usar escape="false"en etiquetas de bloque de texto, porque la cadena de texto sin formato resultante también es XHTML válido. Por lo general, esto no se recomienda, ya que se perderá cualquier formato HTML dentro de la etiqueta.

**5.2 Escape al volver a serializar**

En esta sección, las referencias a "valor de hecho" se refieren al valor semántico de un hecho en un informe XBRL. Si el informe se vuelve a serializar en otro formato, los valores de hecho deben escaparse adecuadamente para ese formato. Por ejemplo, cuando un informe se serializa en la sintaxis XML de XBRL v2.1 (xBRL-XML), los valores de hecho deben tener escape XML. Esto significa que un valor de hecho de:

<b>Profit &amp; Loss</b>

Se escapa a:

&lt;b&gt;Profit &amp;amp; Loss&lt;/b&gt;

Este escape no está relacionado con el comportamiento del escapeatributo y es un artefacto del formato que se utiliza. Por ejemplo, si un hecho se serializara en xBRL-JSON, en su lugar se escaparía de JSON (reemplazando \con \\, y "con \").

**6. Espacio en blanco en hechos de texto**

Cuando el contenido de un documento iXBRL se etiqueta con una ix:nonNumericetiqueta, se debe tener cuidado para garantizar que los valores de hecho extraídos del documento iXBRL conserven los espacios en blanco, de modo que se conserven los espacios entre palabras, párrafos y números.

Esta sección describe algunos de los problemas comunes.

**6.1 Etiquetas HTML a nivel de bloque**

Cuando una ix:nonNumericetiqueta utiliza el escape="false"atributo predeterminado, el valor de hecho resultante es la concatenación de todos los nodos de texto que son descendientes de la etiqueta y de los ix:continuationelementos a los que se hace referencia. Si no se tiene cuidado en la construcción del HTML, es posible que el valor extraído no contenga espacios en blanco en todos los lugares donde el espacio es visible en el informe representado.

Un lugar donde esto puede ocurrir es si el texto se divide en etiquetas de nivel de bloque HTML, como <p>o <div>:

**<ix:nonNumeric** name="eg:DescriptionOfPolicy" context="c1" id="f1" escape="false"**>**

 **<p>**This is the first part of the description.**</p><p>**This is the second part of the description.**</p>**

**</ix:nonNumeric>**

Esto se traducirá en dos párrafos separados:

Esta es la primera parte de la descripción.

Esta es la segunda parte de la descripción.

pero el valor de hecho extraído no incluirá ningún espacio entre las dos oraciones:

Esta es la primera parte de la descripción. Esta es la segunda parte de la descripción.

Incluir espacios en blanco entre el cierre </p>y la siguiente apertura <p>asegurará que se conserve un corte entre las oraciones y no afectará la representación del documento original:

**<ix:nonNumeric** name="eg:DescriptionOfPolicy" context="c1" id="f1" escape="false"**>**

 **<p>**This is the first part of the description.**</p>** **<p>**This is the second part of the description.**</p>**

**</ix:nonNumeric>**

Valor de hecho extraído:

Esta es la primera parte de la descripción. Esta es la segunda parte de la descripción.

Tenga en cuenta que este problema también se aplica a otros modos de visualización de CSS, como list-itemy table-row, y también a la <br>etiqueta.

**6.2 ix: continuación**

Una situación similar puede ocurrir cuando se usa ix:continuation. Esto puede afectar las etiquetas que usan escape="true"o escape="false".

**<div** class="page"**>**

 **<ix:nonNumeric**

 name="eg:DescriptionOfPolicy" context="c1" id="f1"

 escape="false" continuedAt="cont1"

 **>**This is a description**</ix:nonNumeric>**

**</div>**

**<div** class="page"**>**

 **<ix:continuation** id="cont1"**>**of my policy.**</ix:continuation>**

**</div>**

Este arreglo puede ocurrir cuando una oración se divide en un salto de página o de columna. La vista renderizada mostrará la oración dividida en dos páginas o columnas:

esta es una descripción de mi póliza.

pero el valor del hecho extraído será:

Esta es una descripción de mi póliza

Esta situación se puede evitar introduciendo espacio adicional dentro de los ix:continuationelementos.

**6.3 Uso de estilo CSS para crear espacios**

También es posible introducir espacios en la salida renderizada utilizando estilos CSS. Por ejemplo:

**<ix:nonNumeric**

 name="eg:DescriptionOfPolicy" context="c1" id="f1"

 escape="false"

**><span** style="padding-right: 10px"**>**My**</span><span>**policy**</span></ix:nonNumeric>**

Esto se traducirá como:

mi póliza

pero el valor de hecho extraído (usando escape="false") será:

mipolitica

El comportamiento al usarlo escape="true"dependerá del software del visor y también puede depender de si el estilo se aplica usando un estilo en línea o una clase CSS.

Cabe señalar que este uso de estilos para simular saltos de palabras también interferirá con las funciones estándar del navegador web, como copiar y pegar, y la búsqueda de texto, que tratarán el texto como si no hubiera espacio entre las palabras. Del mismo modo, cuando los documentos se publican en la web, es probable que los motores de búsqueda rompan la indexación correcta.

No se recomienda usar estilos para simular saltos de palabras. En su lugar, el informe debe incluir espacios en blanco en el HTML y, si es necesario, ajustar el ancho del espacio representado utilizando otros medios.

**6.4 Normalización de espacios en blanco (escape="false")**

Al renderizar HTML, los navegadores aplican la normalización de espacios en blanco a la mayoría de los espacios en blanco. Esto significa que los espacios en blanco y otros caracteres de espacios en blanco, como tabulaciones y saltos de línea, se representan como un solo espacio. Por ejemplo, los siguientes dos ejemplos se renderizarán de la misma manera:

<**p**>My policy</**p**>

<**p**>My

 policy</**p**>

Al extraer valores de hechos XBRL de una ix:nonNumericetiqueta con escape="false", se conservan todos los espacios en blanco del documento original. Esto significa que el valor de hecho resultante para los dos ejemplos anteriores será diferente.

La especificación Inline XBRL no prescribe cómo se deben presentar dichos valores de hechos a un usuario final y no define si se debe aplicar la normalización de espacios en blanco.

**7. Compatibilidad HTML**

Inline XBRL requiere que los documentos sean XHTML válido, la sintaxis basada en XML para HTML. Aunque similares, los navegadores tratarán XHTML y HTML de manera diferente. Que un documento se trate como XHTML o como HTML depende de una serie de factores, no todos los cuales están bajo el control del autor del documento. Por ejemplo, cuando se coloca un informe en un sitio web, el modo puede estar controlado por el tipo de contenido HTTP con el que se sirve, lo que dependerá de la configuración del servidor.

De manera similar, cuando los hechos de bloques de texto se extraen de un informe iXBRL, es posible que las herramientas que consumen el hecho resultante no siempre lo traten como XHTML en lugar de HTML.

Por esta razón, se recomienda que los informes iXBRL se construyan de manera que resulten en que se representen correctamente en los modos XHTML y HTML. Hay tres cuestiones que deben tenerse en cuenta al hacer esto:

1. Uso de etiquetas de cierre automático;
2. Reglas de escape; y
3. Modo de representación del navegador.

Estos se discuten en más detalle a continuación.

**7.1 Uso de etiquetas de cierre automático**

En XML, una etiqueta vacía puede "autocerrarse". <br />y <br></br>son completamente equivalentes en XML y, por lo tanto, en XHTML.

En HTML, las etiquetas que solo pueden estar vacías, como br, no pueden tener una etiqueta de cierre, y un analizador de HTML tratará una </br>etiqueta de cierre como si fuera otra <br>etiqueta. Esto significa que <br></br>se tratará como dos <br>etiquetas, lo que producirá una diferencia en la representación entre HTML y XHTML. Afortunadamente, las etiquetas de cierre automático (<br />) también se tratan como una sola etiqueta, por lo que los elementos de cierre automático que solo pueden estar vacíos producen los mismos resultados en HTML y XHTML.

Por el contrario, las etiquetas de cierre automático que se espera que tengan contenido también generarán diferencias entre HTML y XHTML. Por ejemplo, en XHTML, una span etiqueta vacía se puede representar como <span />, pero en HTML esto se interpretará como una etiqueta de apertura (una etiqueta de cierre se inferirá en algún momento más adelante en el documento). Esto produce una estructura DOM diferente, con el resultado de que el estilo CSS se puede aplicar de manera diferente.

Para garantizar la coherencia entre HTML y XHTML, las etiquetas que deben estar vacías deben usar la sintaxis de etiquetas de cierre automático, mientras que las etiquetas vacías que pueden contener contenido deben usar la notación expandida.

**7.2 Reglas de escape**

XHTML debe ser XML bien formado, lo que significa que los caracteres especiales XML (<y &) siempre deben escaparse. Por lo general, no se requiere que el >carácter se escape en XML, pero con frecuencia es necesario para mantener la coherencia con <.

HTML sigue reglas de escape similares, pero aplica reglas diferentes dentro de diferentes elementos, más notablemente <style>y <script>elementos.

No se espera que el contenido con una <style> etiqueta tenga escape XML en HTML y una regla de estilo CSS como:

**div** > **p** {**color**: **red**}

no se entenderá si se representa como:

**div** &**gt**; **p** {**color**: **red**}

Por lo tanto, >no debe escaparse dentro de <style>las etiquetas.

<y &no forman parte de la sintaxis de CSS, pero pueden aparecer dentro de los comentarios de CSS y las cadenas de CSS. En este último caso, se pueden representar utilizando las secuencias de escape Unicode, \00003Cy \000026respectivamente.

Un problema similar ocurre con el contenido de <script>la etiqueta, donde los caracteres especiales XML se deben escapar en XHTML, pero no deben estar en HTML. Dado que el contenido de secuencias de comandos generalmente no está permitido en los entornos de generación de informes iXBRL, los enfoques para manejar esto no se analizan más.

En ambos casos, estos problemas de escape se pueden evitar por completo colocando el estilo o el contenido del script en un archivo separado.

**7.3 Modo de renderizado**

Además de los problemas sintácticos mencionados anteriormente, los navegadores, de forma predeterminada, utilizarán diferentes modos de representación para documentos XHTML y HTML. Los documentos XHTML usarán el "modo estándar", mientras que los documentos HTML usarán el "modo peculiar".

En su mayor parte, estos dos modos producen los mismos resultados, pero ciertas construcciones de HTML y CSS se tratarán de manera diferente. Una discusión completa de las diferencias está fuera del alcance de este documento, pero en muchos casos, estas diferencias se pueden resolver mediante el uso de construcciones CSS más explícitas.

**8. Contenedor de sugerencias de resaltado**

El software que muestra los informes iXBRL generalmente resaltará los hechos dentro del informe para diferentes propósitos. Determinar exactamente qué región del informe resaltar no es trivial, ya que cualquier enfoque que se adopte debe hacer frente a la amplia variedad de estructuras HTML diferentes que se ven en los informes iXBRL del mundo real.

El uso generalizado de elementos de posición absoluta, impulsado en gran medida por el uso de software de conversión de PDF a HTML, agrega una complejidad adicional a esto, e incluso cuando dicho HTML se maneja correctamente, el resultado final puede ser un poco confuso.

Por ejemplo, cuando se construyen párrafos completos de texto usando <p></p> etiquetas y luego se etiquetan, se puede aplicar un contorno o un color de fondo a los párrafos como un todo.





Si hay un solo elemento envolvente para el contenido, tendrá, de manera predeterminada, ancho y alto cero, por lo que aplicar resaltado a eso no tendrá el efecto deseado.

Este documento propone un mecanismo que puede ser utilizado por el software de creación para proporcionar una pista que permitirá que el software de visualización brinde un resaltado más claro de dicho contenido.

Cuando el ancestro HTML más cercano de un elemento ix:footnote, ix:nonNumeric, ix:nonFraction, ix:fractiono ix:nonNumerices un elemento HTML con una clase de -ixh-highlight-region, el elemento HTML debe tener una posición y dimensiones que correspondan al contenido dentro de la etiqueta. El software del visor puede optar por aplicar resaltado a ese elemento, en lugar de intentar determinar la extensión del contenido representado de las etiquetas iXBRL.

Esto producirá una apariencia similar a la Figura 1.

Cuando varios elementos iXBRL etiquetan el mismo contenido (etiquetas anidadas), pueden compartir la misma sugerencia de resaltado colocándola alrededor del elemento iXBRL más externo.

