

Validación de documentos



Como hemos comentado anteriormente, la sintaxis de HTML 4 y XHTML está basada en un DTD, y por tanto, se trata de una sintaxis más estricta que HTML 5. A continuación se muestran algunos ejemplos que muestran estas diferencias:

- En **HTML 4** y **XHTML**, todos los elementos se tienen que cerrar para ser válidos, tanto con una etiqueta de cierre correspondiente (como `</body>`) como, en el caso de elementos vacíos, con una barra inclinada al final de la etiqueta (como `
`). Con **HTML 5** ya no es necesario añadir a los elementos vacíos una barra inclinada para su validación.
- En **HTML 4** y **XHTML**, todos los atributos tienen que tener valores, incluso si son redundantes, mientras que en **HTML 5** los atributos de tipo *verdadero* ó *falso* se pueden especificar sin ningún valor.
- En **HTML 5** se pueden emplear nombres de etiquetas o atributos en minúscula, mayúscula o una mezcla de ambas, así como valores de atributos entre comillas o sin comillas (siempre y cuando esos valores no contengan espacios u otros caracteres reservados), mientras que en XHTML no.

1. Algunas pautas básicas

En general, vemos que HTML 5 es mucho más flexible a la hora de validar sus documentos, al menos en lo que se refiere a la sintaxis. Sin embargo, conviene seguir un estilo de sintaxis y ajustarse a él, especialmente si se trabaja en equipo, donde el mantenimiento y la legibilidad del código son cruciales.

Algunas directrices que se podrían considerar son las siguientes:

- Emplear minúsculas para todos los elementos y atributos, como en XHTML.
- Aunque algunos elementos no requieren etiquetas de cierre, recomendamos que todos los elementos que incluyan contenido se cierren (como en `<p>texto</p>`)
- Aunque se pueden dejar valores de atributo sin comillas, es muy probable que usemos algunos que sí las necesiten. Por tanto, se recomienda que siempre las utilicemos para mantener la coherencia. En concreto, utilizaremos comillas *dobles* para definir el valor de los atributos.
- Omitir la barra inclinada final de los elementos que no tienen contenido (como `<meta>`)
- Evitar proporcionar valores redundantes para atributos de tipo *verdadero* ó *falso*, principalmente utilizados en algunas etiquetas HTML destinadas a campos de formulario, y que veremos más adelante.

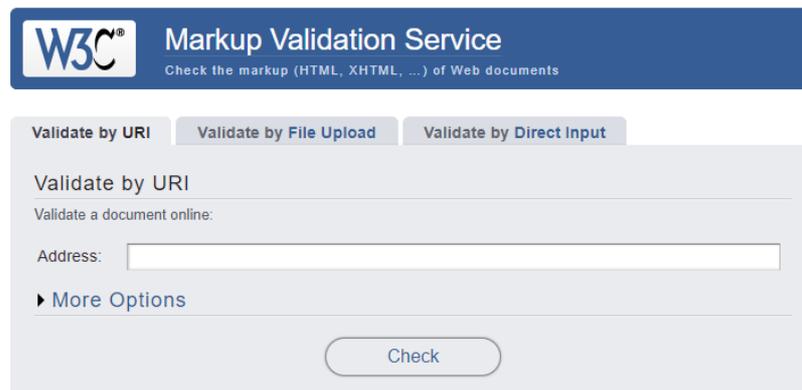
2. Herramientas de validación

Los programadores que trabajan con lenguajes de programación deben corregir su código antes de continuar, ya que de lo contrario no se podrá utilizar el programa. Sin embargo, en los lenguajes de marcas, como HTML, los navegadores son más permisivos, y cuando encuentran un error tratan de intuir lo que se quiso decir y continúan con la carga del documento. Se trata de lenguajes más flexibles a errores pero que puede provocar algunos problemas estéticos y/o funcionales.

Para evitar esto y asegurarnos de que nuestro código está correctamente escrito, se utiliza un **validador HTML**. Se trata de una herramienta que analiza nuestro código y nos dice los errores que tenemos, junto a una breve descripción de los mismos.

2.1. Validador online de W3C

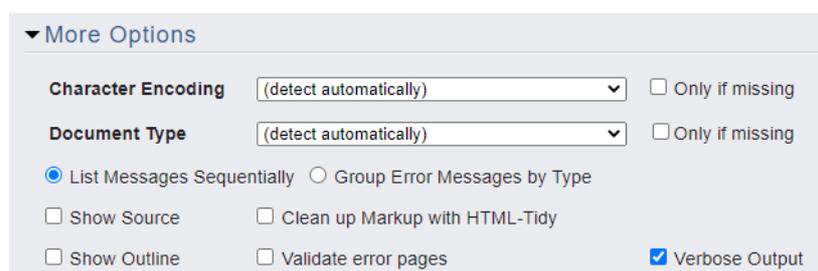
Podemos utilizar el **validador oficial del W3C**, que contiene **Nu HTML Checker**, un validador de HTML5 en fase experimental, pero que puede usarse sin problemas para la búsqueda de errores.



Como vemos, existen tres formas de pasar el validador:

- **Validate by URI:** se indica la dirección URI de la página que se quiere validar.
- **Validate by file upload:** se selecciona el archivo HTML de nuestro equipo, se sube al validador y se realiza el análisis.
- **Validate by direct input:** se escribe directamente el código HTML en el área de texto habilitada, y se analiza dicho código.

Además, si nos fijamos en la parte inferior de la herramienta de validación, vemos la opción **More options** donde podemos personalizar el análisis eligiendo el tipo de codificación de caracteres, el tipo de documento (*HTML 5, XHTML 1.0, HTML 4.01, ...*) y personalizar la página de resultados.



Si pasamos el validador a la [página web del I.E.S. San Vicente](#), por ejemplo, obtenemos el siguiente resultado:

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://iessanvicente.com/>

Checker Input

Show source outline image report

Check by address

https://iessanvicente.com/

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

1. **Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
 From line 34, column 1; to line 34, column 72

```
r /> </p>=<p> &n
```
2. **Error** The `align` attribute on the `img` element is obsolete. [Use CSS instead](#).
 From line 108, column 5; to line 108, column 119

```
jpg">==</a
```

Document checking completed.

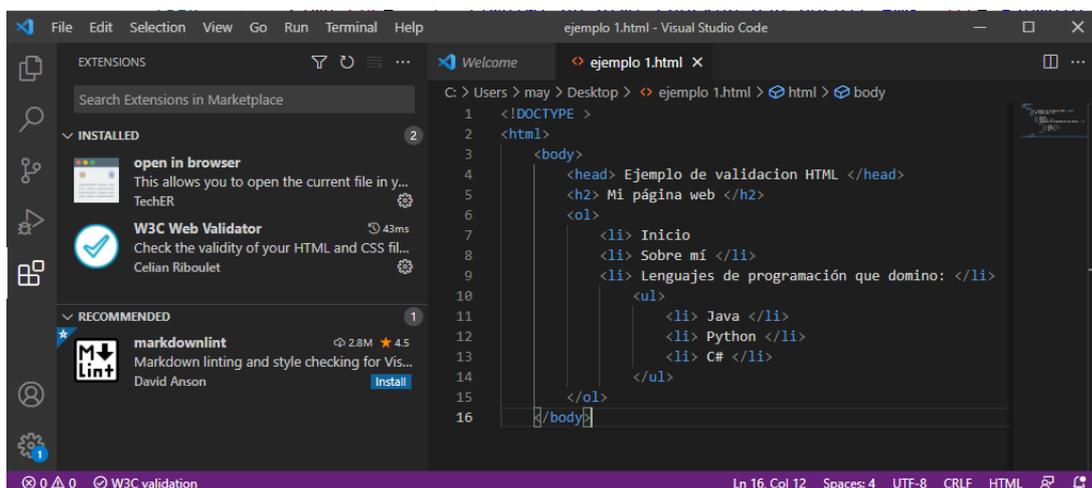
Used the HTML parser. Externally specified character encoding was UTF-8.
Total execution time 807 milliseconds.

Como vemos, la página tiene dos errores, ambos relacionados con atributos del elemento ``.

Es una buena práctica acostumbrarse a utilizar el validador para crear código HTML de calidad y sin errores, ya que si tenemos documentos sin errores de validación, es más probable que se vean correctamente en cualquier tipo de dispositivo.

2.2. Extensiones para VS Code

En el editor **VS Code** podemos encontrar el paquete [W3C Web Validator](#), que permite analizar los documentos HTML directamente desde el editor. Una vez instalado, verás que aparece en la parte izquierda de la barra de estado un icono nuevo, llamado *W3C validation*.



Para validar un documento que tengamos creado, basta con pulsar sobre dicho icono y comenzará la validación. Si se detectan errores, éstos aparecen directamente en el documento. Además, todos los errores y advertencias vienen con una descripción que ayuda a corregirlos.

Ejercicio 1:

Instala la extensión *W3C Web Validator* en VS Code y crea un documento HTML, llamado **validacion.html**, con el siguiente código. Pasa el validador y corrige los errores que encuentres:

```
<!DOCTYPE >
<html>
  <body>
    <head> Ejemplo de validacion HTML </head>
    <h2> Mi página web </h2>
    <ol>
      <li> Inicio
      <li> Sobre mí </li>
      <li> Lenguajes de programación que domino: </li>
        <ul>
          <li> Java </li>
          <li> Python </li>
          <li> C# </li>
        </ul>
      </ol>
    </body>
```