

HTML

HTML stands for **HYPERTEXT-MARKUP-LANGUAGE**. It is the standard markup language for document/article designing that are displayed in web browsers. It can be associated by technologies (CSS) and scripting such as **JavaScript**. The definition of computer program, making **HTML** a programming language.

HTML Uses

We used this language for web development and designing. This language is used for describing the structure of web pages means when we see any website in web browsers and its contents are made by this HTML markup language. Its means author publishes the online documents with headings, images, text, line, tables and graphs etc.

HTML TOOLS

Best online HTML editors are given below:

- i. NOTEPAD++ (free open-source tool)
- ii. Sublime Text (that resembles NOTEPAD++)
- iii. Visual Studio Code
- iv. Atom (most popular tool)

HTML Errors

HTML Errors means a web browser request a service from web server, and the server might return an error code '**404 Not Found**'. These errors are also known as HTTP status messages. The server always shows the message for every request.

There are **two** types of Errors:

- i. Syntax Error
- ii. Logical Error

A **Syntax error** that are occurred in source code of a program. Means when we break the rules of writing statement of programming. It's caught by **compiler**.

A **logical error** is a 'bug' or (mistake) in a program's source code. The result shows unexpected behavior or incorrect message. Program shows **runtime problem**.

Main errors are:

- 404 page not found.
- 305 use proxy.
- Missing character encoding.
- Incorrect formatted HTML.
- Improper tables.

- Missing ALT text in HTML.
- Not supported tags or attributes.
- Missing </head>, </body>, </table> tags.
- Head tag must be start with <head>.
- Script type missing
- Missing <noscript>

1. **404 page not found**

The **HTTP** 404 page not found client error means that the server can't find the requested resource. A 404-status code does not indicate where the resource is temporarily or permanently missing.

2. **305 use proxy**

The HTTP 305 means deprecated status code, that this resource requested by the client is merely available through a proxy. The address of proxy is provided in the server's response by online.

3. **Missing character encoding**

All web pages should define the character set that they are currently using in browser. Character sets are rather technical, they simply tell the online browser what quite set of characters are utilized in this web-page/site.

If a page contains an English character founds on the typical keywords, will have a different character set than one that should display the other characters like Chinese, Japanese characters. The character encoding talks about the user's agent what kind of data to read and display in this source code of a program.

For most English web pages, the character encoding will be entered into the website like this: `<meta http-equiv="Content-Type" content="text/html; charset=iso-88598-1">` This meta tag should be within the <head> and </head> tags for websites or web pages and is not a sensitive case.

- `http-equiv="Content-Type"` tells the what type of meta tag. (Because meta tags have several types)
- `content="text/html"`; its means this tag tells the html document contains only text.
- `charset=iso-88598-1">` tells the web browser using iso-88598-1 character set.

4. **Incorrect formatted HTML**

The common mistakes are usually just a plain human mistakes like:

- **Missing closing tags**

HTML tags have both opening(<head>) and closing(</head>) tags. I always recommended closing <p> tag, even if it is not required now. This usually makes editing your hypertext-markup-language easier as well.

- **Missing Quotation marks for attributes value**

Older version of HTML do not require that your attributes surround values with the quotation marks. But in future version, it is compulsory to use quotation marks surrounding the attributes value.

Example:

```
<img src=myimage.jpg>
<font color=#FF00FF>
<p style=font-face: Time new roman, Geneva>
```

5. Improper tables

Tables are common culprit of improper hypertext-markup-language. It is very easy to incorrectly code tables and most websites or web browsers will let you get away with it. Most common table mistakes are:

- Create tables with different number of cells in each row.
- Not closing the <table>, <tr>, or <td> tags or closing them with proper tags </table>, </tr>, </td>.
- Inserting <td>'s outside of a <tr>
- Placing the tables with inline elements, such as or <h1>.

6. Missing ALT text in HTML

All images have ALT attributes like as:

[](#). As of HTML version 4.01 must require for this.

7. Not supported tags or attribute

Using code that is not part of the HTML standards is not suitable. There also are many attributes of HTML tags that a lot of browsers will identify, but these aren't a part of HTML standard. For accessibility and compatibility reasons, we should always all be using a minimum of version 4.01.

To find out if your page contains not supported HTML tags or attributes, so validate it at the [W3C's HTML Validator](#). If you don't have a <DOCTYPE>, then it won't know which version of HTML to validate your page with.

8. Missing </head>, </body>, </table> tags

The form tag is a block-level tag, meaning that it starts a new section of your page (like as <h1> and <p> do). It is a very common mistake to use the form tags to surrounding the smaller section of your page.

Incorrect:

```
< table><form><tr><td>
.....
</td></tr></form></table>
```

Correct:

```
<form><table><tr>
.....
</tr></table></form>
```

9. Head tag must be start with <head>

<title>, <meta>, and <style> tags must be within the <head> and </head> tags.

10. Script type missing

Scripting languages such as JavaScript and VBScript are very popular nowadays. HTML standards require that you identify the type of scripting language that is being used in web page. Most scripts include attribute language. In future, the attribute language are going to be replaced with the sort of attribute.

```
<script type="text/JavaScript">
<script type="text/VBScript">
```

11. Missing <noscript>

Any JavaScript that performs a function or outputs information must have a tag that gives evidence or opposite for what the JavaScript does.

```

<script type="text/JavaScript">
..... JavaScript stuff here .....

</script>

<noscript>

<p>Access the <A
href="http://anyplace.com/image">image.

</A>

</noscript>

```

The best thanks to determine what mistakes you're making is to validate your page.

How to find & fix errors in HTML?


By following these steps, we can find and fix errors in HTML:

1. Open online simple tool <http://validator.w3.org/> that automatically check your code and point out the errors/problems on your code might have, such as the tags are unsupported, missing characters or quotes around attributes. So, they display syntax error when you compile the code.

This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or

2. You can do testing of your website by three ways:
 - i. Validate by **URL**
 - ii. Validate by **File Upload**
 - iii. Validate by **Direct Input**
3. We will use the 1st option: Validate by URL
4. Next, enter the website name in address box. E.G; <http://www.transpire.com>

→ ↻ validator.w3.org/#validate_by_uri+with_options



Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI
Validate by File Upload
Validate by Direct Input

Validate by URI

Validate a document online:

Address:

▼ More Options

Character Encoding
☐ Only if missing

Document Type
☐ Only if missing

☒ List Messages Sequentially
☐ Group Error Messages by Type

☐ Show Source
☐ Clean up Markup with HTML-Tidy

☐ Show Outline
☐ Validate error pages
☐ Verbose Output

Check

- Click the Check Button. You will see a list of errors and warnings of the web page/site.

Nu Html Checker

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

Showing results for <https://www.transpire.com/>

Checker Input

Show
☐ source
☐ outline
☐ image report
Options...

Check by

Check

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

Message Filtering

- Warning** The `type` attribute is unnecessary for JavaScript resources.
From line 18, column 1; to line 18, column 31
`ie=edge"><script type="text/javascript">+if(!g`
- Warning** The `type` attribute for the `style` element is not needed and should be omitted.
From line 82, column 1; to line 82, column 56
`= 'all' /><style id='wp-block-library-inline-css' type='text/css'>+.has-`

- The Error are displayed as below. You will see the list of warnings and errors.
- Let us understand the errors first. Warnings can be ignored.

Press the Message Filtering button to collapse the filtering options and error/warning/info counts.

Message Filtering

Errors (9) · [Hide all errors](#) · [Show all errors](#)

- 1 ☒ Bad start tag in `img` in `noscript` in `head`.
- 2 Stray end tag `</>`. (2) · [Hide all](#) · [Show all](#)
 - 2.1 ☒ Stray end tag `noscript`.
 - 2.2 ☒ Stray end tag `head`.
- 3 ☒ Start tag `body` seen but an element of the same type was already open.
- 4 ☒ An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#). (4)
- 5 ☒ Element `style` not allowed as child of element `body` in this context. (Suppressing further errors from this subtree.)

Warnings (50) · [Hide all warnings](#) · [Show all warnings](#)

- 1 ☒ The `type` attribute is unnecessary for JavaScript resources. (44)
- 2 ☒ The `type` attribute for the `style` element is not needed and should be omitted. (4)
- 3 ☒ The `main` role is unnecessary for element `main`.
- 4 ☒ Section lacks heading. Consider using `h2` - `h6` elements to add identifying headings to all sections.

8. To ignore the warnings, click Hide all warnings from the message filtering section below.
9. Only errors will be shown as below.

Message Filtering 50 messages hidden by filtering

Errors (9) · [Hide all errors](#) · [Show all errors](#)

- 1 ☒ Bad start tag in `img` in `noscript` in `head`.
- 2 Stray end tag `</>`. (2) · [Hide all](#) · [Show all](#)
 - 2.1 ☒ Stray end tag `noscript`.
 - 2.2 ☒ Stray end tag `head`.
- 3 ☒ Start tag `body` seen but an element of the same type was already open.
- 4 ☒ An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#). (4)
- 5 ☒ Element `style` not allowed as child of element `body` in this context. (Suppressing further errors from this subtree.)

10. Let us understand the error one by one and how to fix it.