# HTML Library

An easy-to-use .NET function library for parsing and directly handling HTML and similar markup languages that use angle-bracketed tags. It supports structures where opening tags may contain name="value" property assignments, such as XML.

This library isn't the same as other HTML processing libraries you will come across. Here, everything takes an object-based approach. You might start by parsing your HTML text into the object model, but afterward, until you render it back out as text, it is purely a tree of generic nodes that can be used in any way they need to be, whether for data processing and conversion, page reformatting and refactoring, or just programmatically building a new page from scratch using **for** loops.

{Line}

## Table of Contents

You can jump to any section of this page from the following list.

* [25 Years of .NET](#_25_Years_of).
* [Yet Another HTML Library](#_Yet_Another_HTML).
* [Installation](#_Installation).
* [Usage Notes](#_Usage_Notes). Including a link to the full API documentation here on GitHub.

{Line}

## 25 Years of .NET

Although .NET doesn't officially turn 25 until February 13, 2027, I'm starting the celebration a little early.

To commemorate 25 years since the public release of the .NET framework, I'm open sourcing this and several other of my long-lived libraries and applications. Most of these have only previously been used privately in our own internal company productivity during the early 21st century but I hope they might find a number of new uses to complete in the next 25 years.

I have every intention of keeping these libraries and applications maintained, so if you happen to run into anything you would like to see added, changed, or repaired, just let me know in the Issues section and I'll get it done for you as time permits.

{Line}

Sincerely,

**Daniel Patterson, MCSD (danielanywhere)**

{Line}

## Yet Another HTML Library

HTML and XML are other of those areas where it seems like once people start to create a function library to handle the data, they insist on making it operate only in opinionated modes that enforce their own viewpoints of how they want to see that information working or fitting together.

After being frustrated too many times by the various limitations, quirks, and special coding requirements of the HTML handlers available circa 2000, I wrote my own generic handler to make sure I would be able to guarantee the productivity of any of my own systems that happened to require working easily and intuitively with angle braced data forms like HTML and XML.

There are no special cases other than whether or not various HTML4 opening tags have matching closing tags.

{Line}

### General Classes

In this library, the main container for HTML operations is **HtmlDocument**, which is inherited from **HtmlNodeItem**, which itself has a **Nodes** property that is a basic collection of **HtmlNodeItems**. There are predicate-based **FindMatch** and **FindMatches** functions to allow you to search throughout the tree using LINQ-like searches.

An **HtmlNodeItem**, also has an **Attributes** property, a collection of **HtmlAttributeItem** objects, and a **Text** property that is used to capture the text occurring after the opening tag and before the content of the **Nodes** collection.

The **HtmlNodeItem.NodeType** property is a string value, so any kind of node type that can be expressed with text is a valid node type. For example, to create an HTML Paragraph element, you could write something like the following.

{Include,Usage\HowTo\ProgrammaticallyCreatingAnHTMLDocumentFromScratch.md}

That might be all you need to know to get started. Please stay tuned, however, there will be a lot more documentation coming soon.

{Line}

## Installation

You can include this library in any .NET project using any supported programming language or target system. This library compiles as **.NET Standard 2.0** and is available in **NuGet** as

{Center}{Bold}{Big}Dan's Html Library{/Big}{/Bold}{/Center}

{Line}

**Instructions For Installation**

In **Visual Studio Community** or **Visual Studio Professional** editions:

* Right-click your project name in **Solution Explorer**.
* From the context menu, select **Manage NuGet Packages**.
* Click **Browse**.
* In the **Search** textbox, type **Dan's Html Library**.
* Accept the license agreement.
* In your code add the header line **using Html;**

{Line}

In **Visual Studio Code**:

* Run the command **NuGet: Add NuGet Package** (typically [Ctrl][Shift][P]).
* If there are multiple projects in the solution, select the open project to which the package will be assigned.
* In the **Search** textbox, type **Dan's Html Library**.
* Select the package.
* Select the version you wish to apply.

{Line}

## Usage Notes

This library is intended to be used on any target system, avoiding any kind of Windows dependencies whatsoever.

You can read how-to articles in the growing library at [**Docs/Usage/HowTo**](https://github.com/danielanywhere/Html/tree/main/Docs/Usage/HowTo). If you don't find what you need right away, either check back soon or create an Issue about the subject and I will make sure it gets explained.

To see working examples of various uses of this library, see the [**Source/HtmlTests**](https://github.com/danielanywhere/Html/tree/main/Source/HtmlTests) folder, where I add various tests and use-cases to a stand-alone application before publishing each version.

If you would like to see a bigger-picture view of the library in daily use, review some of the source of my other GitHub project [**danielanywhere/MarkdownEditor**](https://github.com/danielanywhere/MarkdownEditor). That project uses Dan's Html Library to perform almost all of the tasks that involve working directly with HTML.