

HTMLMinifier

[HTMLMinifier](#) is a highly configurable, well-tested, JavaScript-based HTML minifier.

Installation

From NPM for use as a command line app:

```
npm install html-minifier-terser -g
```

From NPM for programmatic use:

```
npm install html-minifier-terser
```

Usage

Note that almost all options are disabled by default. Experiment and find what works best for you and your project.

For command line usage please see `html-minifier-terser --help` for a list of available options.

Sample command line:

Node.js

```
html-minifier-terser --collapse-  
whitespace --remove-comments --  
minify-js true
```

```
const { minify } = require('html-  
minifier-terser');
```

```
const result = await minify(`<p  
title="blah" id="moo">foo</p>`,  
{  
  removeAttributeQuotes: true,  
});
```

```
result: // <p title=blah id=moo>foo</p>
```

See corresponding [blog post](#) for all the gory details of [how it works](#), description of each option, testing results and conclusions. Also see corresponding [Ruby wrapper](#), and for Node.js, [Gruut middleware wrapper](#). For limit-like capabilities take a look at [HTML limit](#).

Minification comparison

How does HTMLMinifier compare to other solutions — [HTML Minifier from Will Peavy](#) (1st result in [Google search for “html minifier”](#)) as well as [htmlcompressor.com](#) and [minimize?](#)

Site	Original size (KB)	HTMLMinifier minimize	Will Peavy	htmlcomp
Google	52	48	52	54
Stack Overflow	177	143	154	154
HTMLMinifier	252	171	230	250
Bootstrap CSS	271	260	269	229
BBC	355	324	353	344
Amazon	466	430	456	474
Twitter	469	394	462	513
Wikipedia	703	569	682	708
Eloquent Javascript	870	815	840	864
NBC	1701	1566	1689	1705
New York Times	1731	1583	1726	1680
ES draft	6296	5538	5733	n/a

Options Quick Reference

Most of the options are disabled by default.

Option

caseSensitive

collapseBooleanAttributes

collapseInlineTagWhitespace

collapseWhitespace

conservativeCollapse

continueOnParseError

customAttributeAssign

customAttributeCollapses

customEventAttributes

customAttributeSurround

Description

Treat attributes in case sensitive manner (useful for custom HTML tags)

omitAttributeValuesFromTags

booleanAttributes

collapseWhitespace

collapseInlineTagWhitespace

collapseWhitespace

alwaysCollapseTo1Space (never remove it entirely). Must be used in conjunction with `HandleParseErrors` instead of `aborting`.

arrays of regex'es that allow to support custom attribute assign expressions (e.g. `#{if value}`)

regular expression that specifies custom attributes to strip newlines from

arrays of regex'es that allow to support custom attributes surround expressions (e.g. `/ng-class/`)

arrays of regex'es that allow to support custom event attributes for minifyjs (e.g. `ng-click`)

Running local server

npm run serve

Running benchmarks

cd benchmarks
npm install
npm run benchmark

Benchmarks for minified HTML:

HTMLMinifier can't know that original markup was only half of the tree; it does its best to try to parse it as a full tree and it loses information about tree being malformed or partial in the beginning. As a result, it can't create a partial/malformed tree at the time of the output.

Output of resulting markup (e.g. `<p>foo</p>`)

↑
attribute

Transformation of internal representation (e.g. removal of id

Minifying JSON-LD

You can minify script tags with JSON-LD by setting the option `{ processScripts: ['application/ld+json'] }`. Note that this minification is very rudimentary, it is mainly useful for removing newlines and excessive whitespace.

Preserving SVG tags

SVG tags are automatically recognized, and when they are minified, both case-sensitivity and closing-slashes are preserved, regardless of the minification settings used for the rest of the file.

Working with invalid markup

HTMLMinifier **can't work with invalid or partial chunks of markup**. This is because it parses markup into a tree structure, then modifies it (removing anything that was specified for removal, ignoring anything that was specified to be ignored, etc.), then it creates a markup out of that tree and returns it.

Input markup (e.g. `<p id="">foo`)

↓

Internal representation of markup in a form of tree
(e.g. `{ tag: "p", attr: "id", children: ["foo"] }`)

↓

Option	Description
<code>decodeEntities</code>	Use direct Unicode characters whenever possible
<code>html5</code>	Parse input according to HTML5 specifications
<code>ignoreCustomComments</code>	Array of regex'es that allow you to ignore certain comments, which will be matched
<code>ignoreCustomFragments</code>	Array of regex'es that allow you to ignore certain fragments, which will be matched (e.g. <code><?php ... { ... } ...</code> , etc.)
<code>includeAutoGeneratedTags</code>	Insert tags generated by HTML parser
<code>keepClosingSlash</code>	Keep the trailing slash on singleton elements
<code>maxLength</code>	Specify a maximum line length. Compressed output will be separated by newlines at valid HTML line points
<code>minifyCSS</code>	Minify CSS in style elements and style attributes (uses clean-css)
<code>minifyJS</code>	Minify JavaScript in script elements and event attributes (uses Terser)
<code>minifyURLs</code>	Minify URLs in various attributes (uses relateurl)
<code>noNewlinesBeforeTagClose</code>	Never add a newline before the tag that closes an element
<code>preserveLineBreaks</code>	Always collapse to 1 line break (never remove it entirely) without changing the original line numbers

Option	Description	Option	Description	Option	Description	Option	Description	Option	Description	Option	Description
Description	Remove space between attributes. Note that this will result in invalid HTML whenever possible.	removeTagWhitespace	Remove space between tags include line break. Must be used in conjunction with collapseWhitespace=true.	Description	Prevents the escaping of the values of attributes.	Processes contents of conditional comments through minifier.	Array of strings corresponding to types of script elements to process through minifier (e.g. text/ng-template, template, text/x-templatize, template-templatize, etc handlesbars-templatize, etc values (‘ or ’))	Type of quote to use for attribute values (‘ or ’)	Remove quotes around attributes when possible	Strip HTML comments	Remove empty attributes
Special cases	Sort attributes by frequency. Trim white space around sortClassName. Trim white space around sortAttributes and sortOptions like minifier options.	sortAttributes	sortClassName	trimComments	removeAllAttributesWithWhitespaces	removeEmptyElements	removeRedundantAttributes	removeOptimalTags	removeEmptyElements	removeScriptTypeAttributes	removeStyleLinkTypeAttributes
Ignoring chunks of markup	If you have chunks of markup you would like preserved, you can wrap them <!-- htmlmin: ignore -->.	removeTextCSS	removeTextScript	removeOptionalTags	removeAttributesWhenValueIsEmpty	removeAllElementsWithEmptyContents	removeScriptTypeAttributes	removeRedundantAttributes	removeOptimalTags	removeEmptyElements	removeScriptTypeAttributes
Sorting attributes / style classes	sortAttributes / sortClassName won't impact the plain-text size of the output. sortClassName won't improve compression ratio of gzip used in HTTP however, they form long repetitive chains of characters that should improve compression.	sortAttributes	sortClassName	stripHTMLComments	removeAllAttributesWithWhitespacesOnlyValues	removeEmptyAttributes	removeRedundantAttributes	removeOptimalTags	removeScriptTypeAttributes	removeEmptyElements	removeStyleLinkTypeAttributes
Description	Minifier options like sortAttributes and sortOptions.	minifierOptions	sortAttributes and sortOptions	whenPossible	values (‘ or ’)	optionalAttributes	removeAttributesWhenValueIsEmpty	removeEmptyContents	removeAllElementsWithEmptyContents	removeRedundantAttributes	removeOptimalTags
Special cases	sortAttributes by frequency. Trim white space around sortClassName. Trim white space around sortOptions like minifier options.	sortAttributes	sortClassName	whenPossible	values (‘ or ’)	optionalAttributes	removeEmptyContents	removeRedundantAttributes	removeOptimalTags	removeEmptyElements	removeStyleLinkTypeAttributes
Description	removeTextCSS removes the doctype short (HTML5) doctype. It places the doctype in the document conditional comments to support IE9 and earlier.	removeTextCSS	useShortDOCTYPE	whenEscaping	values (‘ or ’)	optionalAttributes	removeEmptyContents	removeRedundantAttributes	removeOptimalTags	removeEmptyElements	removeStyleLinkTypeAttributes
Special cases	sortAttributes by frequency. Trim white space around sortClassName. Trim white space around sortOptions like minifier options.	sortAttributes	sortClassName	whenEscaping	values (‘ or ’)	optionalAttributes	removeEmptyContents	removeRedundantAttributes	removeOptimalTags	removeEmptyElements	removeStyleLinkTypeAttributes