

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:

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

Node.js

```
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, [Grunt plugin](#), [Gulp plugin](#), [Koa middleware wrapper](#) and [Express middleware wrapper](#).

For lint-like capabilities take a look at [HTMLLint](#).

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		Description	
4	caseSensitive	Treat attributes in case sensitive manner (useful for custom HTML tags)	
	collapseBooleanAttributes	Omit attribute values from boolean attributes	
	collapseInlineTagWhitespace	Don't leave any spaces between display:inline; elements when collapsing. Must be used in conjunction with collapseWhitespace=tru	
	collapseWhitespace	Collapse white space that contributes to text nodes in a document tree	
	conservativeCollapse	Always collapse to 1 space (never remove it entirely). Must be used in conjunction with collapseWhitespace=tru	
	continueOnError	Handle parse errors instead of aborting.	
	customAttrAssign	Arrays of regex'es that allow to support custom attribute assign expressions (e.g. ' ')	
	customAttrCollapse	Regex that specifies custom attribute to strip newlines from (e.g. /ng-class/)	
	customAttrSurround	Arrays of regex'es that allow to support custom attribute surround expressions (e.g. <input {{#if value}} {{/checked}}>)	
	customEventAttributes	Arrays of regex'es that allow to support custom event attributes for minifyJS (e.g. ng-click	

Description

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

[Omit attribute values from boolean attributes](#)

Don't leave any spaces between display:inline; elements when collapsing. Must be used in conjunction with collapseWhitespace=tru

[Collapse white space that contributes to text nodes in a document tree](#)

Always collapse to 1 space (never remove it entirely). Must be used in conjunction with collapseWhitespace=tru

Arrays of regex'es that allow to support custom attribute assign expressions (e.g. ' ')

Regex that specifies custom attribute to strip newlines from (e.g. /ng-class/)

Arrays of regex'es that allow to support custom attribute surround expressions (e.g. <input {{#if value}} {{/checked}}>)

Arrays of regex'es that allow to support custom event attributes for minifyJS (e.g. ng-click

Running benchmarks		Running local server	
	Transformation of internal representation (e.g. removal of id attribute)	↑	cd benchmarks npm install npm run benchmark
	Benchmarks for minified HTML:		npm run serve

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

↑

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

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.

Running benchmarks

Benchmarks for minified HTML:

cd benchmarks
npm install
npm run benchmark

Running local server

npm run serve

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

`decodeEntities`

`html5`

`ignoreCustomComments`

`ignoreCustomFragments`

`includeAutoGeneratedTags`

`keepClosingSlash`

`maxLength`

`minifyCSS`

`minifyJS`

`minifyURLs`

`noNewlinesBeforeTagClose`

`preserveLineBreaks`

Description

Use direct Unicode characters whenever possible

Parse input according to HTML specifications

Array of regex'es that allow ignore certain comments, when matched

Array of regex'es that allow ignore certain fragments, when matched (e.g. `<?php ... { { ... } }`, etc.)

Insert tags generated by HTML parser

Keep the trailing slash on singleton elements

Specify a maximum line length. Compressed output will be broken by newlines at valid HTML points

Minify CSS in style elements and style attributes (uses [clean-css](#))

Minify JavaScript in script elements and event attributes (uses [Terser](#))

Minify URLs in various attributes (uses [relateurl](#))

Never add a newline before a tag that closes an element

Always collapse to 1 line when possible (never remove it entirely) when

<div> <div>Option</div> <div> <div>preventAttributesEscaping</div> <div>processConditionalComments</div> <div>processScripts</div> <div>quoteCharacter</div> <div>removeAttributeQuotes</div> <div>removeComments</div> <div>removeEmptyAttributes</div> <div>removeEmptyElements</div> <div>removeOptionalTags</div> <div>removeRedundantAttributes</div> <div>removeScriptTypeAttributes</div> <div>removeStyleLinkTypeAttributes</div> </div> </div> <div> <div>Description</div> <div> <div>whitespace between tags include line break. Must be used in conjunction with collapseWhitespace= true</div> <div>Prevents the escaping of the values of attributes</div> <div>Process contents of conditional comments through minifier</div> <div>Array of strings corresponding to types of script elements to process through minifier (e.g. text/ng-template, text/x-handlebars-template, etc</div> <div>Type of quote to use for attribute values (' or ")</div> <div>Remove quotes around attributes when possible</div> <div>Strip HTML comments</div> <div>Remove all attributes with whitespace-only values</div> <div>Remove all elements with empty contents</div> <div>Remove optional tags</div> <div>Remove attributes when value matches default.</div> <div>Remove type="text/javascript" from script tags. Other type attribute value are left intact</div> <div>Remove type="text/css" from style and link tags. Other type attribute values are left intact</div> </div> </div>	6
<div> <div>Option</div> <div> <div>removeTagWhitespace</div> <div>sortAttributes</div> <div>sortClassName</div> <div>trimCustomFragments</div> <div>useShortDoctype</div> </div> </div> <div> <div>Description</div> <div> <div>Remove space between attributes whenever possible. Note that will result in invalid HTML</div> <div>Sort attributes by frequency</div> <div>Sort style classes by frequency</div> <div>Trim white space around ignoredCustomFragments</div> <div>Replaces the doctype with short (HTML5) doctype</div> </div> </div> <div> <div>Sorting attributes / style classes</div> <div> <div>Minifier options like sortAttributes and sortClassName won't impact the plain-text size of the output. However, they form long repetitive chains of characters that should improve compression ratio of gzip used in HTTP compression.</div> </div> </div> <div> <div>Special cases</div> <div> <div>Ignoring chunks of markup</div> <div> <div>If you have chunks of markup you would like preserved, you can wrap them <!-- htmlmin:ignore -->.</div> </div> </div> </div>	7