

You can override the HTML comment by using the option `omitTag` as explained above. Both, the tag and the actual comment in the Markdown file are case-insensitive.

## Additional infos

- This plugin outputs a semantically correct table of contents. Sub-lists are rendered within the parent `<li>` tag and not as a separate (empty) `<li>`.
- Headlines can be in an arbitrary order. For example, h3, h2, h4. Please note that the jump from h2 to h4 causes a double-indentation, which is correct.

# markdown-it-table-of-contents

A table of contents plugin for Markdown-it. Simple, customizable and with a default slugifier that matches that of [markdown-it-anchor \(>5.0.0\)](#).

## Usage

CommonJS:

```
var MarkdownIt = require("markdown-it");
var md = new MarkdownIt();

md.use(require("markdown-it-anchor").default); // Optional,
// but makes sense as you really
// want to link to something, see
// info about recommended plugins
// below
md.use(require("markdown-it-table-of-
contents"));
```

ESM:

```

import MarkdownIt from 'markdown-it';
import MarkdownItTOC from 'markdown-it-toc';
import MarkdownItTableOfContents from 'markdown-it-table-of-contents';

const md = new MarkdownIt();
md.use(MarkdownItTOC);
const md = new MarkdownIt();
md.use(MarkdownItTableOfContents);

<h3 id="a-message-from-our-sponsors">A
message from our sponsors</h3>
<p></p>
<p></p>
<h2 id="hello">Hello world, I think you
should read this article</h2>
<p></p>
<h2 id="what-is-next?>What's next?</h2>
<p>Lorem ipsum</p>
<h2>Read this next...</h2>
<p></p>
<h4 id="related">See related articles</h4>
<h4>Read this next...</h4>
<h2>What's next?</h2>
<h2>Read this next...</h2>
<h4># Title
<!-- omit from toc -->
# Title
<!-- omit from toc -->
<h1>the TOC
<h2>Example for omitting headlines from
the TOC

```

If you want to exclude single headlines, you can use a special HTML comment to omit the next headline from the TOC:

```

need to allow HTML in MarkdownIt:
line before the headline you want to exclude. Furthermore, you
For this to work, the HTML comment must come right in the
HTML code:
<!-- omit from toc -->
# Title
<!-- omit from toc -->
# Title
<!-- omit from toc -->
# Sub heading 1
Some nice text
## Sub heading 2
Some even nicer text

```

For this to work, the HTML comment must come right in the line before the headline you want to exclude. Furthermore, you need to allow HTML in MarkdownIt:

```

const md = new MarkdownIt({ html:
  true });

```

specified in „usage“ above:

... would render this HTML using the default options

```

## Sub heading 2
Some nice text
# Sub heading 1
Some even nicer text

```

## Example markdown

```

import MarkdownIt from 'markdown-it';
import MarkdownItTOC from 'markdown-it-toc';
import MarkdownItTableOfContents from 'markdown-it-table-of-contents';

const md = new MarkdownIt();
const md = new MarkdownIt();
md.use(MarkdownItTableOfContents);
md.use(MarkdownItTOC);

table-of-contents:

Want to use a table of contents generator client-side in the
browser? Try my `` webcomponent `` table-of-contents-element-on
GitHub. Advantage: can use more sophisticated query selector
and better support for HTML customization.

```

Then add `` where you want the table of contents to be added

```

const md = new MarkdownIt();
const md = new MarkdownIt();

```

```

import MarkdownIt from 'markdown-it';
import MarkdownItTableOfContents from 'markdown-it-table-of-contents';

```

This markdown:

# Heading

... would render this HTML using the default options

```

## Sub heading 2
Some nice text
# Sub heading 1
Some even nicer text

```

Some even nicer text

Some nice text

## Sub heading 1

Some nice text

This markdown:

```

## Sub heading 2
Some nice text
# Sub heading 1
Some even nicer text

```

Some even nicer text

Some nice text

## Sub heading 1

Some nice text

This markdown:

```
#### See related articles {#related}
```

HTML output:

```
<h1 id="article">Article</h1>
<p>

    <ul>
        <li>
            <ul>
                <li><a href="#a-
message-from-our-sponsors">A
message from our sponsors</a></
li>
            </ul>
        </li>
        <li><a href="#hello">Hello
world, I think you should read
this article</a></li>
        <li><a href="#what's-
next%3F">What's next?</a>
            <ul>
                <li>
                    <ul>
                        <li><a
href="#related">See related
articles</a></li>
                    </ul>
                </li>
            </ul>
        </li>
    </ul>
</p>
```

```
<h1 id="heading">Heading</h1>

<ul>
    <li><a href="#heading">Heading</a>
        <ul>
            <li><a href="#sub-
heading-1">Sub heading 1</a></
li>
            <li><a href="#sub-
heading-2">Sub heading 2</a></
li>
        </ul>
    </li>
</ul>

<h2 id="sub-heading-1">Sub heading 1</
h2>
<p>Some nice text</p>

<h2 id="sub-heading-2">Sub heading 2</
h2>
<p>Some even nicer text</p>
```

## Options

You may specify options when using the plugin, like so:

```
md.use(require("markdown-it-table-of-
contents"), options);
```

These options are available:

|                                   |   |  |  |  |  |         |   |          |
|-----------------------------------|---|--|--|--|--|---------|---|----------|
| Name                              | Description Default   | Headings<br>etc)   | The class for<br>the container<br>“table-of-contents”  | A custom<br>slugification<br>function<br>EncodedURIComponent ( | /\([ \[toc\] \]/im<br>to be<br>the marker<br>Regex<br>pattern of<br>with TOC<br>HTML<br>comment<br>tag to<br>exclude next <!-- omit from toc | omitTag | Type of list<br>(ul for<br>unordered, ol for<br>ordered)<br>A function<br>for<br>headings<br>formatting<br>md.renderInline (con | format   |
| includeLevel                      | Levels to use [1,2]   | (2 for h2:s  | Detailed<br>Description Default  | A container<br>class for<br>the<br>“table-of-contents”         | DIY  | slugify | markerPattern   | listType |
| containerClass                    | etc)  | etc)   | the<br>class for<br>the container<br>“table-of-contents”   | Custom<br>slugification<br>function<br>EncodedURIComponent (   | /\([ \[toc\] \]/im<br>to be<br>the marker<br>Regex<br>pattern of<br>with TOC<br>HTML<br>comment<br>tag to<br>exclude next <!-- omit from toc | omitTag | listType  | format   |
| order                             | Of course, both plugins can be combined. markdown-it-anchor<br>ignores headings that already have an id attribute.<br>Furthermore, markdown-it-table-of-contents can handle<br>unusual heading orders. Consider the full example below: | Var md = new MarkdownIt();<br>md.use(markdownItTOC, {<br>"includeLevel": [2,3,4]<br>});<br>md.use(markdownItTOC, {<br>});<br>md.use(require("markdown-it-toc"), {<br>});<br>md.use(require("markdown-it-anchor"), {<br>});<br>md.use(require("markdown-it-headers"), {<br>});<br># Article | ## A message from our sponsors<br>Ad<br>this article {#hello}<br>## Hello world, I think you should read<br>that's article next?<br>Lorem ipsum<br>Read this next... | format   | format   | format  | format  | format   |
| full example with unusual heading | These options are available:  |  |  |  |  |         |   |          |

## markdown-it-anchor

This plugin transforms all headlines in a markdown document so that the HTML code includes an id. It *slugifies* the headline:

```
## Hello world, I think you should read  
this article
```

Becomes

```
<h2 id="hello-world-i-think-you-should-  
read-this-article">Hello world,  
I think you should read this  
article</h2>
```

## markdown-it-attrs

This plugin lets you attach custom attributes to your headlines. This is especially useful, if you have long headlines but want short anchors:

```
## Hello world, I think you should read  
this article {#hello}
```

Becomes

```
<h2 id="hello">Hello world, I think you  
should read this article</h2>
```

| Name   | Description   | Default           |
|--|---|-------------------|
| containerHeaderHtml  | Optional HTML string for container header             | undefined         |
| containerFooterHtml  | Optional HTML string for container footer             | undefined         |
| transformLink  | A function for transforming the TOC links             | undefined         |
| transformContainerOpen   | A function for transforming the container opening tag | (see source code) |
| transformContainerClose  | A function for transforming the container closing tag | (see source code) |
| getTokensText  | A function for extracting text from tokens for titles | (see source code) |
| format is an optional function for changing how the headings are displayed in the TOC. |   |                   |

## Recommended Plugins

```
return <button>Toggle</button>
button><h3>Table of Contents</h3>
return <h3>'</h3>;
} ,
} ,
transformContainerClose: () => {
  return '';
}
getTokensText is a function that can be used to check how text is extracted from tokens to support more ways headlines are build. See source code for more information on equivalent function in markdown-it-amcho.
```

```
    By default, TOC headings will be formatted using markdown-  
    its internal MD formatting rules (i.e. it will be formatted using the  
    same rules / extensions as other markdown in your document).  
    You can override this behavior by specifying a custom format  
    function. The function should accept two arguments:  
    1. content - The heading text, as a markdown string.  
    2. md - markdown-it's internal markdown parser object. This  
    should only be need for advanced use cases.  
function format(content, md) {  
    // manipulate the headings as you like  
    return manipulateHeadingsString;  
}
```