

Changelog

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](#), and this project adheres to [Semantic Versioning](#).

[1.0.0] - 2025-08-28

- Runs stable for years now, hence push version to 1.0. No breaking changes to v0.9.0.
 - **Added:** Now also works as ESM module
 - **Removed (dev):** (Internal) mocha replaced with node test runner for running tests
-

[0.9.0] - 2025-01-21

- **Added:** Headlines can now be omitted from the table of contents by placing a special HTML comment tag before the headline (fixes #65)
 - **Added:** Option `omitTag` can override the default tag `<!-- omit from toc -->`
-

[0.8.0] - 2024-09-10

- **Added:** Option `getTokensText` to override how text is extracted from tokens to build headlines and slugs (fixes #61), similar to the function in [markdown-it-anchor](#).

[0.7.0] - 2024-09-09

- **Added:** Override the container element
- **⚠ BREAKING CHANGE:** The plugin moved from *inline* *mode* to *block mode* (fixes #62)
- **Changed:** Updated tests, readme etc.
- **Removed:** Old `forceFullToc` attribute

Override the container element

Two new options that accept functions that return HTML to render custom containers (and more elements if necessary):

```
md.use(markdownItToc, {
  transformContainerOpen: () => {
    return '<button>Toggle</
    button><h3>Table of Contents</
    h3>';
  }
});
```

```
    },  
    transformContainerClose: () => {  
      return '';  
    }  
  }  
});
```

Inline mode is now block mode

Input:

Output before:

```
<p></p>
```

Output now:

The TOC now is generated in block mode, which removes the wrapping p tag. Wrapping a div in a p is considered invalid HTML.

If you really need a wrapping p-element, you can emulate the old behavior with the new container override functions:

```
const md = new MarkdownIt();  
md.use(markdownItTOC, {  
  transformContainerOpen: () => {  
    return '<p>';  
  },  
  transformContainerClose: () => {  
    return '</p>';  
  }  
});
```

```
}  
});
```

Be aware that the old tests/examples now behave differently
when using soft breaks before the markup:

Input:

```
# Article  
Text with soft line break (two spaces)
```

Headline

Output before:

```
<h1>Article</h1>  
<p>Text with soft line break (two  
spaces)<br>  
...</p>
```

Output now:

```
<h1>Article</h1>  
<p>Text with soft line break (two  
spaces)</p>
```

...

[0.6.0] - 2021-11-12

The TOC generator was rewritten, because the old *on-the-fly* generator couldn't deal with unexpected order of headings and double-indentations. It is now a three-step process:

1. Gather all headings in a list.
2. Turn that list into a nested tree.
3. Generate HTML code based on the nested tree.

Although all tests pass, this release could introduce some **breaking changes** for you, if you relied on the old way of doing things. Check the test cases to get a better understanding how this plugin handles various cases.

- **Added:** Support for markdown - it - attrs (fixes #54)
- **Changed:** Respects unexpected nesting order (fixes #55)
- **Changed:** Uses anchor targets from existing id attributes (for example, set by markdown - it - attrs or markdown - it - anchor)
- **Changed:** Now nests list correctly if there is a jump (for example: h2, h2, h4 -> h4 is now double-indented)
- **Removed:** unused tests