

highlightjs / highlight.js

Public

JavaScript syntax highlighter with language auto-detection and zero dependencies.

highlightjs.org/

BSD-3-Clause license

22.7k stars 3.6k forks Branches Tags Activity

Star

Notifications

<> Code

Issues 101

Pull requests 22

Actions

Projects 1

Wiki

Security 2

Insights

main

14 Branches

113 Tags

Go to file

Go to file

<> Code

nixxquality

enh(erlang) highlight character literals (#3970)

last week

github	(chore) update Node.js test versions (#3954)	3 weeks ago
demo	(docs) make demo page more keyboard acces...	3 years ago
docs	(docs) articulation and grammar, checked by ...	3 months ago
extra	(chore) Fix repo URL (#3579)	2 years ago
highlight.js	enh(themes) Tweak background color for Gra...	3 years ago
src	enh(erlang) highlight character literals (#3970)	last week
test	enh(erlang) highlight character literals (#3970)	last week
tools	enh(1c) add punctuation and add 1c-light the...	last month
types	(fix) TS interface LanguageDetail > keywords (...)	2 weeks ago
.dockerignore	enh(build) slimmer Docker image and add .do...	4 years ago
.editorconfig	Support .editorconfig.	10 years ago
.eslintrc.js	Fix linting error with 'import' keyword (#3929)	2 months ago
.eslintrc.lang.js	(chore) Simplify linting configuration (#2900)	4 years ago
.git-blame-ignore-revs	(chore) blame ignore eslint --fix	2 years ago
.gitattributes	add (gitattributes) LF for txt files	9 years ago
.gitignore	(docs) improve api docs	3 years ago
.mocharc.json	chore(tests) allow running of individual tests a...	4 years ago
.readthedocs.yaml	(fix) use myst-parser	3 years ago
.tokeignore	chore(docs) - more lovely badges!	4 years ago
CHANGES.md	enh(java) Add goto keyword	2 weeks ago
CONTRIBUTING.md	(docs) update fiddle links	3 years ago
CONTRIBUTORS.md	(chore) update contributors	2 years ago
Dockerfile	(fix) Ensure git is installed in docker image (#2...	4 years ago
	chore: Cleanup LICENSE (#2145)	5 years ago

📄 LICENSE		
📄 MAINTAINERS_GUIDE.md	move maintainers guide	3 years ago
📄 README.CDN.md	(chore) rename master branch to main	3 years ago
📄 README.md	bump 11.9.0	3 months ago
📄 SECURITY.md	(chore) version 10 EOL	3 years ago
📄 SUPPORTED_LANGUAGES.md	(docs) add 3rd support zig grammar to SUPP...	last month
📄 VERSION_10_BREAKING_CHANGES.md	chore(docs) update my profile URL on GitHub	4 years ago
📄 VERSION_10_UPGRADE.md	(chore) rename master branch to main	3 years ago
📄 VERSION_11_UPGRADE.md	(chore) version 10 EOL	3 years ago
📄 package-lock.json	bump 11.9.0	3 months ago
📄 package.json	bump 11.9.0	3 months ago
📄 tsconfig.json	(chore) fix TS issue, bump 10.7.1	3 years ago

Highlight.js

latest v11.9.0
license BSD-3-Clause
npm install 5.04 MB
minified 109 kB
npm downloads 6.8M/week
jsDelivr CDN 170.1M/month

build success
CodeQL passing
Snyk security monitored

discord
issues 123
help welcome 81
good first issue 30

Highlight.js is a syntax highlighter written in JavaScript. It works in the browser as well as on the server. It can work with pretty much any markup, doesn't depend on any other frameworks, and has automatic language detection.

Contents

- [Basic Usage](#)
 - [In the Browser](#)
 - [Plaintext Code Blocks](#)
 - [Ignoring a Code Block](#)
 - [Node.js on the Server](#)
- [Supported Languages](#)
- [Custom Usage](#)
 - [Using custom HTML](#)
 - [Using with Vue.js](#)
 - [Using Web Workers](#)
- [Importing the Library](#)
 - [Node.js / `require`](#)
 - [ES6 Modules / `import`](#)
- [Getting the Library](#)
 - [Fetch via CDN](#)
 - [cdnjs \(link\)](#)
 - [jsdelivr \(link\)](#)
 - [unpkg \(link\)](#)
 - [Download prebuilt CDN assets](#)

- [Download from our website](#)
- [Install via NPM package](#)
- [Build from Source](#)
- [Requirements](#)
- [License](#)
- [Links](#)

Upgrading to Version 11

As always, major releases do contain breaking changes which may require action from users. Please read [VERSION_11_UPGRADE.md](#) for a detailed summary of breaking changes and any actions you may need to take.

Support for older versions

Please see [SECURITY.md](#) for long-term support information.

Basic Usage

In the Browser

The bare minimum for using highlight.js on a web page is linking to the library along with one of the themes and calling `highlightAll`:

```
<link rel="stylesheet" href="/path/to/styles/default.min.css">
<script src="/path/to/highlight.min.js"></script>
<script>hljs.highlightAll();</script>
```

This will find and highlight code inside of `<pre><code>` tags; it tries to detect the language automatically. If automatic detection doesn't work for you, or you simply prefer to be explicit, you can specify the language manually by using the `class` attribute:

```
<pre><code class="language-html">...</code></pre>
```

Plaintext Code Blocks

[README](#) [BSD-3-Clause license](#) [Security](#)

```
<pre><code class="language-plaintext">...</code></pre>
```

Ignoring a Code Block

To skip highlighting of a code block completely, use the `nohighlight` class:

```
<pre><code class="nohighlight">...</code></pre>
```

Node.js on the Server

The bare minimum to auto-detect the language and highlight some code.

```
// load the library and ALL languages
hljs = require('highlight.js');
html = hljs.highlightAuto('<h1>Hello World!</h1>').value
```

To load only a "common" subset of popular languages:

```
hljs = require('highlight.js/lib/common');
```



To highlight code with a specific language, use `highlight` :

```
html = hljs.highlight('<h1>Hello World!</h1>', {language: 'xml'}).value
```



See [Importing the Library](#) for more examples of `require` vs `import` usage, etc. For more information about the result object returned by `highlight` or `highlightAuto` refer to the [api docs](#).

Supported Languages

Highlight.js supports over 180 languages in the core library. There are also 3rd party language definitions available to support even more languages. You can find the full list of supported languages in [SUPPORTED_LANGUAGES.md](#).

Custom Usage

If you need a bit more control over the initialization of Highlight.js, you can use the [highlightElement](#) and [configure](#) functions. This allows you to better control *what* to highlight and *when*.

For example, here's the rough equivalent of calling [highlightAll](#) but doing the work manually instead:

```
document.addEventListener('DOMContentLoaded', (event) => {  
  document.querySelectorAll('pre code').forEach((el) => {  
    hljs.highlightElement(el);  
  });  
});
```



Please refer to the documentation for [configure](#) options.

Using custom HTML

We strongly recommend `<pre><code>` wrapping for code blocks. It's quite semantic and "just works" out of the box with zero fiddling. It is possible to use other HTML elements (or combos), but you may need to pay special attention to preserving linebreaks.

Let's say your markup for code blocks uses divs:

```
<div class='code'>...</div>
```



To highlight such blocks manually:

```
// first, find all the div.code blocks  
document.querySelectorAll('div.code').forEach(el => {  
  // then highlight each  
  hljs.highlightElement(el);  
});
```



Without using a tag that preserves linebreaks (like `pre`) you'll need some additional CSS to help preserve them. You could also [pre and post-process line breaks with a plug-in](#), but *we recommend using CSS*.

To preserve linebreaks inside a `div` using CSS:

```
div.code {  
  white-space: pre;  
}
```



Using with Vue.js

See [highlightjs/vue-plugin](https://highlightjs.com/vue-plugin) for a simple Vue plugin that works great with Highlight.js.

An example of `vue-plugin` in action:

```
<div id="app">
  <!-- bind to a data property named `code` -->
  <highlightjs autodetect :code="code" />
  <!-- or literal code works as well -->
  <highlightjs language='javascript' code="var x = 5;" />
</div>
```



Using Web Workers

You can run highlighting inside a web worker to avoid freezing the browser window while dealing with very big chunks of code.

In your main script:

```
addEventListener('load', () => {
  const code = document.querySelector('#code');
  const worker = new Worker('worker.js');
  worker.onmessage = (event) => { code.innerHTML = event.data; }
  worker.postMessage(code.textContent);
});
```



In worker.js:

```
onmessage = (event) => {
  importScripts('<path>/highlight.min.js');
  const result = self.hljs.highlightAuto(event.data);
  postMessage(result.value);
};
```



Importing the Library

First, you'll likely be installing the library via `npm` or `yarn` -- see [Getting the Library](#).

Node.js / `require`

Requiring the top-level library will load all languages:

```
// require the highlight.js library, including all languages
const hljs = require('./highlight.js');
const highlightedCode = hljs.highlightAuto('<span>Hello World!</span>').value
```



For a smaller footprint, load our common subset of languages (the same set used for our default web build).

```
const hljs = require('highlight.js/lib/common');
```



For the smallest footprint, load only the languages you need:

```
const hljs = require('highlight.js/lib/core');
hljs.registerLanguage('xml', require('highlight.js/lib/languages/xml'));

const highlightedCode = hljs.highlight('<span>Hello World!</span>', {language: 'xml'}).value
```



ES6 Modules / `import`

Note: You can also import directly from fully static URLs, such as our very own pre-built ES6 Module CDN resources. See [Fetch via CDN](#) for specific examples.

The default import will register all languages:

```
import hljs from 'highlight.js';
```



It is more efficient to import only the library and register the languages you need:

```
import hljs from 'highlight.js/lib/core';
import javascript from 'highlight.js/lib/languages/javascript';
hljs.registerLanguage('javascript', javascript);
```



If your build tool processes CSS imports, you can also import the theme directly as a module:

```
import hljs from 'highlight.js';
import 'highlight.js/styles/github.css';
```



Getting the Library

You can get highlight.js as a hosted, or custom-build, browser script or as a server module. Right out of the box the browser script supports both AMD and CommonJS, so if you wish you can use RequireJS or Browserify without having to build from source. The server module also works perfectly fine with Browserify, but there is the option to use a build specific to browsers rather than something meant for a server.

Do not link to GitHub directly. The library is not supposed to work straight from the source, it requires building. If none of the pre-packaged options work for you refer to the [building documentation](#).

On Almond. You need to use the optimizer to give the module a name. For example:

```
r.js -o name=hljs paths.hljs=/path/to/highlight out=highlight.js
```



Fetch via CDN

A prebuilt version of Highlight.js bundled with many common languages is hosted by several popular CDNs. When using Highlight.js via CDN you can use Subresource Integrity for additional security. For details see [DIGESTS.md](#).

cdnjs ([link](#))

Common JS

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.9.0/styles/default.min.css">
<script src="https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.9.0/highlight.min.js"></script>
<!-- and it's easy to individually load additional languages -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.9.0/languages/go.min.js"></script>
```



ES6 Modules

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.9.0/styles/dark.min.css">
<script type="module">
import hljs from 'https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.9.0/es/highlight.min.js';
// and it's easy to individually load additional languages
import go from 'https://cdnjs.cloudflare.com/ajax/libs/highlight.js/11.9.0/es/languages/go.min.js';
hljs.registerLanguage('go', go);
</script>
```



jsdelivr ([link](#))

Common JS

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/highlightjs/cdn-release@11.9.0/build/styles/default.min.css">
<script src="https://cdn.jsdelivr.net/gh/highlightjs/cdn-release@11.9.0/build/highlight.min.js"></script>
<!-- and it's easy to individually load additional languages -->
<script src="https://cdn.jsdelivr.net/gh/highlightjs/cdn-release@11.9.0/build/languages/go.min.js"></script>
```

ES6 Modules

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/highlightjs/cdn-release@11.9.0/build/styles/default.min.css">
<script type="module">
import hljs from 'https://cdn.jsdelivr.net/gh/highlightjs/cdn-release@11.9.0/build/es/highlight.min.js';
// and it's easy to individually load additional languages
import go from 'https://cdn.jsdelivr.net/gh/highlightjs/cdn-release@11.9.0/build/es/languages/go.min.js';
hljs.registerLanguage('go', go);
</script>
```

unpkg (link)

Common JS

```
<link rel="stylesheet" href="https://unpkg.com/@highlightjs/cdn-assets@11.9.0/styles/default.min.css">
<script src="https://unpkg.com/@highlightjs/cdn-assets@11.9.0/highlight.min.js"></script>
<!-- and it's easy to individually load additional languages -->
<script src="https://unpkg.com/@highlightjs/cdn-assets@11.9.0/languages/go.min.js"></script>
```

ES6 Modules

```
<link rel="stylesheet" href="https://unpkg.com/@highlightjs/cdn-assets@11.9.0/styles/default.min.css">
<script type="module">
import hljs from 'https://unpkg.com/@highlightjs/cdn-assets@11.9.0/es/highlight.min.js';
// and it's easy to individually load & register additional languages
import go from 'https://unpkg.com/@highlightjs/cdn-assets@11.9.0/es/languages/go.min.js';
hljs.registerLanguage('go', go);
</script>
```

Note: The CDN-hosted `highlight.min.js` package doesn't bundle every language. It would be very large. You can find our list of "common" languages that we bundle by default on our [download page](#).

Download prebuilt CDN assets

You can also download and self-host the same assets we serve up via our own CDNs. We publish those builds to the [cdn-release](#) GitHub repository. You can easily pull individual files off the CDN endpoints with `curl`, etc; if say you only needed `highlight.min.js` and a single CSS file.

There is also an npm package [@highlightjs/cdn-assets](#) if pulling the assets in via `npm` or `yarn` would be easier for your build process.

Download from our website

The [download page](#) can quickly generate a custom single-file minified bundle including only the languages you desire.

Note: [Building from source](#) can produce slightly smaller builds than the website download.

Install via NPM package

Our NPM package including all supported languages can be installed with NPM or Yarn:

```
npm install highlight.js
# or
yarn add highlight.js
```

Alternatively, you can build the NPM package from source.

Build from Source

The [current source code](#) is always available on GitHub.

```
node tools/build.js -t node
node tools/build.js -t browser :common
node tools/build.js -t cdn :common
```



See our [building documentation](#) for more information.

Requirements

Highlight.js works on all modern browsers and currently supported Node.js versions. You'll need the following software to contribute to the core library:

- Node.js >= 12.x
- npm >= 6.x

License

Highlight.js is released under the BSD License. See our [LICENSE](#) file for details.


Links

The official website for the library is <https://highlightjs.org/>.

Further in-depth documentation for the API and other topics is at <http://highlightjs.readthedocs.io/>.

A list of the Core Team and contributors can be found in the [CONTRIBUTORS.md](#) file.

Releases 37

 **11.9.0 - Fall is in the air!** Latest
on Oct 8, 2023

[+ 36 releases](#)

Used by 2m



+ 1,956,935

Contributors 680



[+ 666 contributors](#)

Languages