Getting Markdown plugins to work with Gulp

I finally solved two big misteries: Why does Gulp give you the ability to write your own code and how to integrate plugins into a Gulp-based Markdown project.

Why write your own code

There are times when there is no plugin available for a certain task or the plugin doesn't as expected.

In my gulp-starter I use Markdown-It to convert Markdown to HTML.

I would love to use plugins to provide equivalent functionality to what I get using Markdown in WordPress via the Jetpack Plugin in the classic editor.

Using a Gulp plugin (gulp-markdown-it) is a good way to get started but I wasn't successful in adding Markdown-It plugins to work with the Gulp plugin.

A solution I found was to write my own code and bypass the Gulp Markdown-It plugin altogether while still using Markdown-It as the parser.

The next sections explain how I did it.

Integrating plugins to Markdown workflows

The first step is to require the plugins we want to use.

I'm using CommonJS modules rather than ESM modules. Eventhough all currently supported versions of Node support ESM imports, not all modules I'm using do and I don't want to go through the hassle of converting them.

I'm also using the Markdown-It tool directly rather than using a plugin. I will explain why in the next section.

We require all the plugins we want to use before we use them.

```
// require gulp first
const gulp = require('gulp');

'gulp'// require markdown-it markdownIt = require('markdown-it');

// req'markdown-it'// require markdown-it plugins
const abbr = require("markdown-it-abbr");
const alerts = require("markdown-it-alerts");
const anc = require("markdown-it-anchor");
const attrs = require("markdown-it-attrs");
const embed = require("markdown-it-block-embed");
const fn = require("markdown-it-footnote");
const figs = require("markdown-it-implicit-figures");
const kbd = require("markdown-it-kbd");
const prism = require("markdown-it-prism");
const toc = require("markdown-it-table-of-contents");
const list = require("markdown-it-task-lists");
```

First, we configure Markdown-It parser optionsp.

I chose to use the Commonmark preset. Commonmark solves a lot of the problems that Markdown has experienced, pretty much from the beginning.

I also chose to enable the following options by setting them to true:

`html` : Enable HTML tags in source

`xhtmlOut` : Uses / to close single tags (
). :This is only for full CommonMark compatibility.

`linkify` : Autoconvert URL-like text to links

`typographer` : Enable some language-neutral replacement + quotes beautification : For the full list of replacements, see https://github.com/markdown-it/markdown-it/blob/master/lib/rules_core/replacements.js

```
// Markdown-It Options
const options = {
```

```
  preset: 'commonmark',
  html: true,
  xhtmlOut: true,
  linkify: true,
  typographer: true,
};
```

Next step is to initialize Markdown-It and then declare the plugins we want to use via the use method.

The `use()` method also allows is to pass options to the plugins as the second parameter of the `use()` declaration.

```
const md = new markdownIt();
md.use(abbr);
md.use(alerts);
md.use(anc);
md.use(attrs);
md.use(embed);
md.use(fn);
md.use(figs);
md.use(kbd);
md.use(prism);
md.use(toc);
md.use(list);
```

The first custom function will actually convert the file to HTML and replace the extension with `.html`.

```
function markdownToHtml(file) {
  const result = md.render(file.contents.toString());
  file.contents = new Buffer(result);
  file.path = replaceExt(file.path, '.html');
  return;
}
```

Next is our Markdown converter task. We use gulp-tap plugin to run the external

function inside the task. Otherwise, it looks like a regular task and it works like one too :).

```
function markdown() {
  return gulp
       .src('src/md-content/*.md')
       .pipe(tap(markdownToHtml))
       .pipe(gulp.dest('src/html-content/'));
}
```

Customizing Markdown-It

Right now we have all our functionality running, from definition lists to table of contents. But there are parts that can't be done with plugins and is not part of the Markdown-It API.

> We'll explore how to do this in a future post.