Better Markdown from Node

Unless I write an email in Gmail I do all my writing using Markdown a quick text-based writing system. I love Markdwon but have always been upset at the limitations placed in the original Markdown, the fact that there is no actual specification for Markdown and that Markdown's creator has been a douche when it comes to supporting the community effort of rectifying some of the shortcomings of the original Markdown tool (leading to the CommonMark effort).

I currently use Ulysses for MacOS/iOS as my primary writing tool and then copy and paste to a Markdown enabled instance of Wordpress using Jetpack. The one thing I've only been partially successful in doing is to convert the Markdown directly into HTML and uset it as is.

The best I've been able to do is convert the Markdown file to (badly formatted) HTML and insert it into a template... for some reason the converter keeps thinking that bulledted lists should be formated with paragraphs (as in `<li><p>`) and that creates issues with dropcaps in list items. I have to manually clean the documents after conversion and I'm pretty sure I can do better than that.

The idea

If I'm publishing the content to Wordpress I don't need to do anything further. The Markdown code is good enough to publish with only minor alterations (Wordpress has some issues with the way it renders Markdown and you need to preview your content before you publish).

There are times when I want to publish content to the web without having to worry about manually generating the HTML. In order to get the page ready we'll do the following:

1. generate the HTML from Markdown
2. Insert the generated HTML into an existing HTML template
3. Visually inspect the page and the underlying HTML
4. Clean up the resulting page based on inspection

The tools

I've plugged the Markdown process to my CSS Starter Kit. SO the template is linked to the CSS generated for the starter kit and also to Prism syntax highlighter.

The Markdown section uses the following tools

- Remarkable Markdown Gulp plugin
    - gulp-remarkable
- Wrapper to insert HTML into a template
    - gulp-wrap

Gulp build file

I've added two tasks to work with Markdow. The first task will generate HTML from all the files under `src/md-content` that ends with an `md` extension and place them in the `src/html-content` directory.

Note that this task will only convert to HTML the content of the Markdown file. It will not add head or body elements. That will come in the next task.

```
gulp.task('markdown', () => {
  return gulp.src('src/md-content/*.md'src/md-content/*.md'({
      preset: 'commonmark',
      typographer: true,
      remarkableOptions: {
        typographer: true,
        linkify: true,
        breaks: false
      }
    }))
    .pipe(gulp.dest('src/html-content/'));
});
```

The second time will build the full HTML files. It will take each of the HTML fragments we created with the Markdown task, insert them into a template and save them to the root of our `src` directory with the name of the fragment as the name of the HTML file.

```
gulp.task('build-template', ['markdown'], function() {
'markdown'('./src/html-content/*.html')
    .pipe($.wrap({src: './templa'./src/html-content/*.html'pipe(gulp.
});
```

We make the task dependent on Markdown to make sure we have the freshest HTML fragments before inserting them into the templates.

What's next

Working with Wordpress and its Markdown parser I've gotten used to typing in HTML code manually. Videos in the page are added as HTML with the following snippet

```
<div class="video">
<iframe width="560" height="315"
src="https://www.youtube.com/embed/K1SFnrf4jZo"
frameborder="0" allowfullscreen></iframe>
</div>
```

It would be nice if we can just tailor the Markdown we write to produce the same HTML without having to write it by hand.

Remarkable, the Markdown library I'm using for this project, has a rich plugin ecosystem. I'm researching how to run plugins inside the Gulp task. Once this is done I will be able to incorporate the results of the plugins into the HTML content. This may mean I'll have to use the HTML in the blog (instead of the Markdown file directly) but I still save myself from having to code the HTML manually :-)