

# pandoc-emphasize-code

## Contents

Usage . . . . .	1
Syntax . . . . .	2
Rendering to HTML . . . . .	2
Rendering with LaTeX . . . . .	3
Regular Highlighting . . . . .	3
Install . . . . .	4
From Hackage . . . . .	4
Build . . . . .	4
Run . . . . .	4
Changelog . . . . .	5
License . . . . .	5

## Usage

Often when working with code examples in documentation, printed or web hosted, or in presentation slideshows, you might want to emphasize parts of a code snippet.

You can get away with manually writing the target markup, in LaTeX or raw HTML, but if you want to render the same document in multiple output formats, this gets really tedious. Also, having to write the markup by hand can be error prone.

This filter lets you specify *ranges* of a code block to emphasize, and have the filter generate the appropriate markup for you. It recognizes code blocks with the `emphasize` attribute present:

```
{ . haskell emphasize=2:3-2:14,3:3-3:12 }
myFunc = do
  newStuffHere
  andThisToo notThis
  notSoRelevant
```

In the example above, the identifier `newStuffHere` and `andThisToo` will be emphasized.

Currently, the following output formats are supported:

- HTML (html and html5)
- LaTeX (latex and beamer)
- GitHub-Flavored Markdown (markdown\_github)
- RevealJS (revealjs)

## Syntax

The value of the `emphasize` attribute is a comma-separated list of *ranges*. A *range* consists of two positions, separated by a dash. A *position* consists of a *line number* and a *column number*, separated by a colon.

The syntax can be described in EBNF, like so:

```
line number    = natural number;
column number  = natural number;
position       = line number, ":", column number;
range          = position, "-", position;
ranges         = range, { ("", range) };
```

(\* definition of natural number excluded for brevity \*)

There must be at least one range in the comma-separated list. A range can span multiple lines.

## Rendering to HTML

The code block above would render HTML output like the following (lines broken for readability):

```
<pre class="haskell"><code>myFunc = do
  <mark>newStuffHere</mark>
  <mark>andThisToo</mark> notThis
  notSoRelevant</code></pre>
```

When rendering to html5 or revealjs, the emphasized ranges are wrapped in `<mark>` tags. The default browser styling is black text on yellow background, but can be customized with CSS:

```
code mark {
  background-color: black;
  color: white;
}
```

The html and markdown\_github output formats use `<em>` tags instead of `<mark>` tags. By default, `<em>` tags are rendered in italic type, but can be customized with CSS:

```
code em {
  font-weight: bold;
  font-style: normal;
}
```

**NOTE:** There is no additional syntax highlighting when emphasizing code and rendering to HTML, as there is no way to use Pandoc’s highlighter and embed custom HTML tags. You might be able to add that using a Javascript highlighter running on the client.

## Rendering with LaTeX

When rendering using LaTeX, two things are required:

- The `listings` package needs to be included.
- You need to define a `CodeEmphasis` command, styling the emphasized code in `lstlistings`.

If you’re not using a custom LaTeX template, you can use the YAML front matter in a Markdown source file to add the requirements:

```
header-includes:
- \usepackage{listings}
- \lstset{basicstyle=\ttfamily}
- \newcommand{\CodeEmphasis}[1]{\textcolor{red}{\textit{#1}}}
```

**NOTE:** When rendering as Beamer slides, any frame including an emphasized block must be marked as `fragile`:

```
## My Slide {.fragile}

{.haskell emphasize=2:3-2:14,3:3-3:12}
myFunc = do
  newStuffHere
  andThisToo notThis
  notSoRelevant
```

## Regular Highlighting

You can still use regular Pandoc highlighting (the *skylighting* library):

```
{.haskell}
myFunc :: The Type -> Signature
myFunc = do
  newStuffHere
  andThisToo notThis
```

```
notSoRelevant
```

It gives you all the nice colors:

```
myFunc :: The Type -> Signature
myFunc = do
  newStuffHere
  andThisToo notThis
  notSoRelevant
```

The drawback is that you have two different highlighting systems now, one for emphasized code, one for regular code blocks.

## Install

Executables for Linux and macOS are available in the Releases page.

### From Hackage

If you'd rather install using cabal or stack, you can use the following command:

```
cabal install pandoc-emphasize-code
```

The package is available at Hackage.

## Build

Requirements:

- Cabal or Stack, either works.

To install from sources, run:

```
git clone git@github.com:owickstrom/pandoc-emphasize-code.git
cd pandoc-emphasize-code
stack setup
stack install
```

## Run

If you have installed from sources, and you have `~/ .local/bin` on your PATH, you can use the filter with Pandoc like so:

```
pandoc --filter pandoc-emphasize-code input.md output.html
```

## Changelog

- (next)
  - Revert to use newlines in HTML pre tags
  - Use default Setup.hs script
- **0.2.1**
  - Support revealjs output
  - Use <mark> for HTML5 and RevealJS, <em> for HTML and GFM
- **0.2.0**
  - Use Lucid to render HTML, fixes issue #1
- **0.1.1**
  - Restructured modules
  - Separated pretty printer
  - Better error messages
  - Improved validation
  - Documentation styling improvements
- **0.1.0**
  - First release
  - Support for multiple ranges
  - Rendering support for HTML, Markdown, and LaTeX

## License

Mozilla Public License Version 2.0