

# Introduction to *Pressdown*

*Pressdown* is an extension of Markdown Extra that adds special processing directives to handle the specific needs of documentation tasks. Some of the directives help reduce markup, and others allow for easily escaping Twig, Markdown, and *Pressdown* directives.

CBD

## The Classed Block Directive

One main feature is the Classed Block Directive - `{@x-c .+ x@}`. This allows you to briefly create HTML blocks with a specific assigned class. The letter `x` can be replaced by one or two letters that represent the following tags: a, b(lockquote), c(ode), f(igure), f(ig)c(aption), d(iv), l(i), o(l), p, s(pan), and u(l). The letter `c` can be replaced by a single class name or a series of class names separated by a `..`. You can also leave it empty and no class name will be applied to the block at all. See Figure 1 below for an example.

With *Pressdown* this is how you make an element with a specific class:

```
this:
@{s-z I need to be a span with "z" class s@}

becomes this:
<span class="z">I need to be a span with "z" class</span>
```

Then you can use `.z` in your CSS files to do something cool.

Figure 1: The Classed Block Directive `{@x-c .+ x@}`

You can nest classed blocks and use Markdown Extra within them! Below is how I created the above sample. You'll see a Twig expression for the press variable `pressDown`. It itself evaluates to the classed block directive: `{@s-a Pressdown s@}`.

```
{@f-
  With {{ pressDown }} this is how you make an element with a specific class:

  ""none
  this:
  @{s-z I need to be a span with "z" class s@}

  becomes this:
  <span class="z">I need to be a span with "z" class</span>
  ""

  Then you can use '.z' in your CSS files to do something cool.
  f@}
  {@fc- Figure 1: The Classed Block Directive '{@x-c .+ x@}' fc@}
```

After Twig, *Pressdown*, and Markdown Extra parsing, the final html markup looks like this:

```
<figure>With <span class="a"-Pressdown</span> this is how you make an element with a specific class:
<pre><code class="language-none">
  this:
  @{s-z I need to be a span with "z" class s@}

  becomes this:
  &lt;span class="z"&gt;I need to be a span with "z" class&lt;/span&gt;</code></pre>
  Then you can use <code>.z</code> in your CSS files to do something cool.</figure>
<figcaption>Figure 1: The Classed Block Directive <code>{@x-c .+ x@}</code></figcaption>
```

## Figures and Captions

If you were paying attention above, you noticed that Figure 1 uses classed blocks of the `f(figure)` and `f(ig)c(aption)` variety. **CLI Press** automatically styles figures and captions to look like insets as the one above. But as simple as the classed block directive is, **Pressdown** has an even better way to create figures and captions. The Figure Directive - `{=fig-name .+ fig=}` `(caption)?` .

Let's break down what that does, and how **Pressdown** helps you with it. The `.+` is the content that will go in the `figure` tag. The `name` should be a unique name for the figure. This name allows you to create a link to the figure from any file using the Figure Link Directive. Finally the `caption` is an optional caption that will be added to the bottom of the figure, like the caption on the example above. And here's the best part: **Pressdown** will automatically number your figures for you, and prefix your caption with "Figure X:". Don't worry about renumbering figures. It's handled.

To see the directive in action, here's an example of how to create *Figure 1* using the Figure Directive.

```
{=fig-ClassedBlockDirective
  With {{ pressDown }} this is how you make an element with a specific class:

  ``none
  this:
  @{{s-z I need to be a span with "z" class s@}}

  becomes this:
  <span class="z">I need to be a span with "z" class</span>
  ``

  Then you can use '.z' in your CSS files to do something cool. fig=}{The Classed Block Directive '{@x-c .+ x@}'}
```

And here's that **Pressdown** in action:

With **Pressdown** this is how you make an element with a specific class:

```
this:
@{{s-z I need to be a span with "z" class s@}}

becomes this:
<span class="z">I need to be a span with "z" class</span>
```

Then you can use `.z` in your CSS files to do something cool.

Figure 1: The Classed Block Directive `{@x-c .+ x@}`