

Search the Docs

### What's on this Page

[What a Shortcode is](#)

[Use Shortcodes](#)

[Use Hugo's Built-in Shortcodes](#)

[Privacy Config](#)

[Create Custom Shortcodes](#)



## CONTENT MANAGEMENT

# Shortcodes

---

Shortcodes are simple snippets inside your content files calling built-in or custom templates.

## What a Shortcode is

Hugo loves Markdown because of its simple content format, but there are times when Markdown falls short. Often, content authors are forced to add raw HTML (e.g., video `<iframe>`'s) to Markdown content. We think this contradicts the beautiful simplicity of Markdown's syntax.

Hugo created **shortcodes** to circumvent these limitations.

A shortcode is a simple snippet inside a content file that Hugo will render using a predefined template. Note that shortcodes will not work in template files. If you need the

In addition to cleaner Markdown, shortcodes can be updated any time to reflect new classes, techniques, or standards. At the point of site generation, Hugo shortcodes will easily merge in your changes. You avoid a possibly complicated search and replace operation.

## Use Shortcodes [↗](#)

Shortcodes | Hugo - Static Site Generator | Tutorial 9



In your content files, a shortcode can be called by calling `{{% shortcodeName parameters %}}`. Shortcode parameters are space delimited, and parameters with internal spaces can be quoted.

The first word in the shortcode declaration is always the name of the shortcode. Parameters follow the name. Depending upon how the shortcode is defined, the parameters may be named, positional, or both, although you can't mix parameter types in a single call. The format for named parameters models that of HTML with the format `name="value"`.

Some shortcodes use or require closing shortcodes. Again like HTML, the opening and closing shortcodes match (name only) with the closing declaration, which is prepended with a slash.

Here are two examples of paired shortcodes:

```
{{% mdshortcode %}}Stuff to `process` in the *center*.{{% /mdshortcode %}}
```

```
{{< highlight go >}} A bunch of code here {{< /highlight >}}
```

The examples above use two different delimiters, the difference being the % character in the first and the <> characters in the second.

## Shortcodes with raw string parameters [↗](#)

You can pass multiple lines as parameters to a shortcode by using raw string literals:

```
{{< myshortcode `This is some <b>HTML</b>,
and a new line with a "quoted string".` >}}
```

## Shortcodes with Markdown [↗](#)

In Hugo 0.55 we changed how the % delimiter works. Shortcodes using the % as the outer-most delimiter will now be fully rendered when sent to the content renderer. They can be part of the generated table of contents, footnotes, etc.

If you want the old behavior, you can put the following line in the start of your shortcode template:

```
{{ $_hugo_config := `{ "version": 1 }` }}
```

## Shortcodes Without Markdown [↗](#)

The < character indicates that the shortcode's inner content does not need further rendering. Often shortcodes without Markdown include internal HTML:

```
{{< myshortcode >}}<p>Hello <strong>World!</strong></p>{{< /myshortcode >}}
```

## Nested Shortcodes [↗](#)

You can call shortcodes within other shortcodes by creating your own templates that leverage the `.Parent` variable. `.Parent` allows you to check the context in which the shortcode is being called. See [Shortcode templates](#).

## Use Hugo's Built-in Shortcodes [↗](#)

Hugo ships with a set of predefined shortcodes that represent very common usage. These shortcodes are provided for author convenience and to keep your Markdown content clean.

### figure [↗](#)

`figure` is an extension of the image syntax in Markdown, which does not provide a shorthand for the more semantic [HTML5 `<figure>` element](#).

The `figure` shortcode can use the following named parameters:

#### src

URL of the image to be displayed.

#### link

If the image needs to be hyperlinked, URL of the destination.

#### target

Optional `target` attribute for the URL if `link` parameter is set.

#### rel

Optional `rel` attribute for the URL if `link` parameter is set.

## alt

Alternate text for the image if the image cannot be displayed.

## title

Image title.

## caption

Image caption. Markdown within the value of `caption` will be rendered.

## class

`class` attribute of the HTML `figure` tag.

## height

`height` attribute of the image.

## width

`width` attribute of the image.

## attr

Image attribution text. Markdown within the value of `attr` will be rendered.

## attrlink

If the attribution text needs to be hyperlinked, URL of the destination.

## Example `figure` Input [↗](#)

figure-input-example.md

Copy

```
{{< figure src="/media/spfl13.jpg" title="Steve Francia" >}}
```

## Example `figure` Output [↗](#)

figure-output-example.html

```
<figure>
  
  <figcaption>
    <h4>Steve Francia</h4>
  </figcaption>
</figure>
```

## gist

Bloggers often want to include GitHub gists when writing posts. Let's suppose we want to use the [gist at the following url](https://gist.github.com/spf13/7896402):

```
https://gist.github.com/spf13/7896402
```

We can embed the gist in our content via username and gist ID pulled from the URL:

```
{{< gist spf13 7896402 >}}
```

## Example gist Input

If the gist contains several files and you want to quote just one of them, you can pass the filename (quoted) as an optional third argument:

gist-input.md

Copy

```
{{< gist spf13 7896402 "img.html" >}}
```

## Example gist Output

gist-output.html

```
<script type="application/javascript" src="https://gist.github.com/spf13/78
```

## Example `gist` Display

To demonstrate the remarkable efficiency of Hugo's shortcode feature, we have embedded the `spf13 gist` example in this page. The following simulates the experience for visitors to your website. Naturally, the final display will depend on your stylesheets and surrounding markup.

```
1 <!-- image -->
2 <figure {{ if isset .Params "class" }}class="{{ index .Params "class" }}"{{ end }}>
3   {{ if isset .Params "link" }}<a href="{{ index .Params "link" }}">{{ end }}
4     
5   {{ if isset .Params "link" }}</a>{{ end }}
6   {{ if or (or (isset .Params "title") (isset .Params "caption")) (isset .Params
"attr") }}
7     <figcaption>{{ if isset .Params "title" }}
8       <h4>{{ index .Params "title" }}</h4>{{ end }}
9       {{ if or (isset .Params "caption") (isset .Params "attr") }}<p>
10        {{ index .Params "caption" }}
11        {{ if isset .Params "attrlink" }}<a href="{{ index .Params "attrlink" }}"> {{ end }}
12         {{ index .Params "attr" }}
13         {{ if isset .Params "attrlink" }}</a> {{ end }}
14       </p> {{ end }}
15     </figcaption>
16   {{ end }}
17 </figure>
18 <!-- image -->
```

img.html hosted with  by GitHub

[view raw](#)

## highlight

This shortcode will convert the source code provided into syntax-highlighted HTML. Read more on [highlighting](#). `highlight` takes exactly one required `language` parameter and requires a closing shortcode.

```
{{< highlight html >}}  
<section id="main">  
  <div>  
    <h1 id="title">{{ .Title }}</h1>  
    {{ range .Pages }}  
      {{ .Render "summary" }}  
    {{ end }}  
  </div>  
</section>  
{{< /highlight >}}
```

## Example highlight Output [↗](#)

The `highlight` shortcode example above would produce the following HTML when the site is rendered:

tutorials/learn-html/index.html

```
<span style="color: #f92672">&lt;section</span> <span style="color: #a6e22e"  
  <span style="color: #f92672">&lt;div&gt;</span>  
    <span style="color: #f92672">&lt;h1</span> <span style="color: #a6e22e">  
      {{ range .Pages }}  
        {{ .Render "summary" }}  
      {{ end }}  
    <span style="color: #f92672">&lt;/div&gt;</span>  
<span style="color: #f92672">&lt;/section&gt;</span>
```

To see even more options for adding syntax-highlighted code blocks to your website, see [Syntax Highlighting in Developer Tools](#).



## instagram

If you'd like to embed a photo from [Instagram](#), you only need the photo's ID. You can discern an Instagram photo ID from the URL:

```
https://www.instagram.com/p/BWNjjyYFxVx/
```

## Example `instagram` Input

instagram-input.md

Copy

```
{{< instagram BWNjjyYFxVx >}}
```

You also have the option to hide the caption:

instagram-input-hide-caption.md

Copy

```
{{< instagram BWNjjyYFxVx hidecaption >}}
```

## Example `instagram` Output

By adding the preceding `hidecaption` example, the following HTML will be added to your rendered website's markup:

instagram-hide-caption-output.html

## Example `instagram` Display

Using the preceding `instagram` with `hidecaption` example above, the following simulates the displayed experience for visitors to your website. Naturally, the final display will be contingent on your style sheets and surrounding markup.

The `instagram-shortcode` refers an endpoint of Instagram's API, that's deprecated since October 24th, 2020. Thus, no images can be fetched from this API endpoint, resulting in an error when the `instagram-shortcode` is used. For more information please have a look at GitHub issue [#7879](#).

## param [↗](#)

Gets a value from the current `Page`'s `params` set in front matter, with a fallback to the site `param` value. It will log an `ERROR` if the `param` with the given key could not be found in either.

```
{{< param testparam >}}
```

Since `testparam` is a `param` defined in front matter of this page with the value `Hugo Rocks!`, the above will print:

Hugo Rocks!

To access deeply nested params, use “dot syntax”, e.g:

```
{{< param "my.nested.param" >}}
```

## ref and relref [↗](#)

These shortcodes will look up the pages by their relative path (e.g., `blog/post.md`) or their logical name (`post.md`) and return the permalink (`ref`) or relative permalink (`relref`) for the found page.

`ref` and `relref` also make it possible to make fragmentary links that work for the header links generated by Hugo.

Read a more extensive description of `ref` and `relref` in the [cross references documentation](#)

`ref` and `relref` take exactly one required parameter of reference, quoted and in position 0.

## Example `ref` and `relref` Input [↗](#)

```
[Neat] ({{< ref "blog/neat.md" >}})
[Who] ({{< relref "about.md#who" >}})
```

## Example `ref` and `relref` Output [↗](#)

Assuming that standard Hugo pretty URLs are turned on.

```
<a href="https://example.com/blog/neat">Neat</a>
<a href="/about/#who">Who</a>
```

## tweet [↗](#)

You want to include a single tweet into your blog post? Everything you need is the URL of the tweet:

```
https://twitter.com/SanDiegoZoo/status/1453110110599868418
```

## Example `tweet` Input [↗](#)

Pass the tweet's user (case-insensitive) and ID from the URL as parameters to the `tweet` shortcode.

example-tweet-input.md

Copy

```
{{< tweet user="SanDiegoZoo" id="1453110110599868418" >}}
```

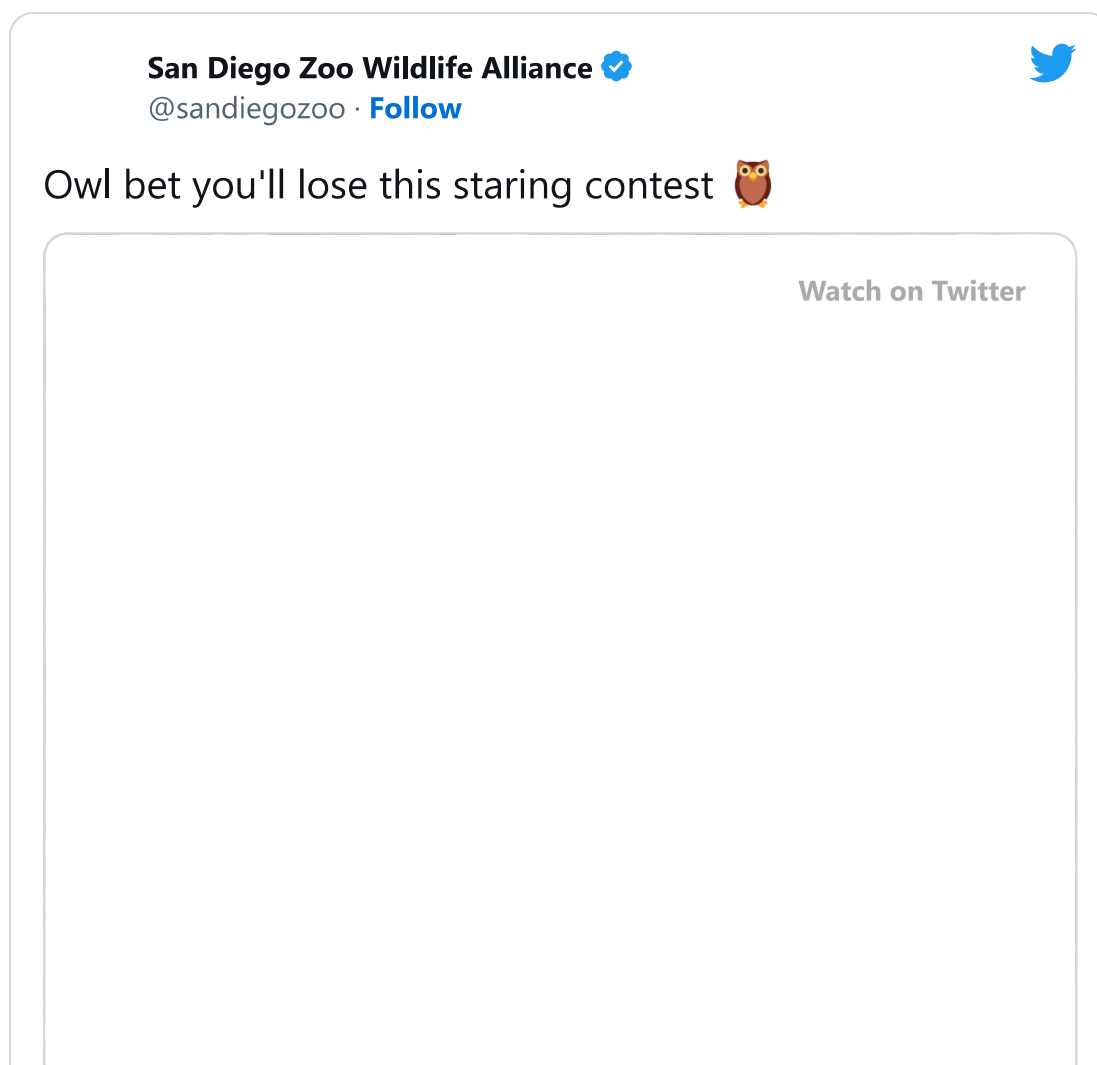
Using the preceding `tweet` example, the following HTML will be added to your rendered website's markup:

`example-tweet-output.html`

```
<blockquote class="twitter-tweet"><p lang="en" dir="ltr">Owl bet you&#39;ll  
<script async src="https://platform.twitter.com/widgets.js" charset="utf-8"
```

## Example `tweet` Display [↗](#)

Using the preceding `tweet` example, the following simulates the displayed experience for visitors to your website. Naturally, the final display will be contingent on your stylesheets and surrounding markup.



11:23 PM · Oct 26, 2021



441



Reply



Share

[Read 10 replies](#)

## vimeo

Adding a video from [Vimeo](#) is equivalent to the [YouTube Input shortcode](#).

```
https://vimeo.com/channels/staffpicks/146022717
```

## Example vimeo Input

Extract the ID from the video's URL and pass it to the `vimeo` shortcode:

example-vimeo-input.md

Copy

```
{{< vimeo 146022717 >}}
```

## Example vimeo Output

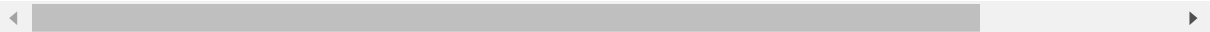
Using the preceding `vimeo` example, the following HTML will be added to your rendered website's markup:

example-vimeo-output.html

```
<div style="position: relative; padding-bottom: 56.25%; height: 0; overflow: hidden">
  <iframe src="https://player.vimeo.com/video/146022717" style="position: absolute; top: 0; left: 0; width: 100%; height: 100%;" />
</div>
```

If you want to further customize the visual styling of the YouTube or Vimeo output, add a `class` named parameter when calling the shortcode. The new `class` will be added to the `<div>` that wraps the `<iframe>` and will remove the inline styles. Note that you will need to call the `id` as a named parameter as well. You can also give the vimeo video a descriptive title with `title`.

```
{{< vimeo id="146022717" class="my-vimeo-wrapper-class" title="My vime
```



## Example vimeo Display [↗](#)

Using the preceding `vimeo` example, the following simulates the displayed experience for visitors to your website. Naturally, the final display will be contingent on your stylesheets and surrounding markup.

03:02

## youtube [↗](#)

The `youtube` shortcode embeds a responsive video player for [YouTube videos](#). Only the ID of the video is required, e.g.:

## Example `youtube` Input [↗](#)

Copy the YouTube video ID that follows `v=` in the video's URL and pass it to the `youtube` shortcode:

example-youtube-input.md

Copy

```
{{< youtube w7Ft2ymGmfc >}}
```

Furthermore, you can automatically start playback of the embedded video by setting the `autoplay` parameter to `true`. Remember that you can't mix named and unnamed parameters, so you'll need to assign the yet unnamed video ID to the parameter `id`:

example-youtube-input-with-autoplay.md

Copy

```
{{< youtube id="w7Ft2ymGmfc" autoplay="true" >}}
```

For [accessibility reasons](#), it's best to provide a title for your YouTube video. You can do this using the shortcode by providing a `title` parameter. If no title is provided, a default of "YouTube Video" will be used.

example-youtube-input-with-title.md

Copy

```
{{< youtube id="w7Ft2ymGmfc" title="A New Hugo Site in Under Two Minutes"
```



## Example `youtube` Output [↗](#)

Using the preceding `youtube` example, the following HTML will be added to your rendered website's markup:

example-youtube-output.html

Copy

```
<div style="position: relative; padding-bottom: 56.25%; height: 0; overflow: hidden">
  <iframe src="https://www.youtube.com/embed/w7Ft2ymGmfc?autoplay=1" style="width: 100%; height: 100%; border: none;" allow="autoplay; fullscreen; picture-in-picture" />
</div>
```

## Example `youtube` Display [↗](#)

Using the preceding `youtube` example (without `autoplay="true"`), the following simulates the displayed experience for visitors to your website. Naturally, the final display will be contingent on your style sheets and surrounding markup. The video is also include in the [Quick Start of the Hugo documentation](#).

A New Hugo Site in Under 2 Minutes



## Privacy Config [↗](#)

To learn how to configure your Hugo site to meet the new EU privacy regulation, see [Hugo and the GDPR](#).

## Create Custom Shortcodes [↗](#)

To learn more about creating custom shortcodes, see the [shortcode template documentation](#).



# See Also

- [markdownify](#)
- [.RenderString](#)
- [anchorize](#)
- [Shortcode Variables](#)
- [Markdown Render Hooks](#)

Last updated: December 18, 2022: [Remove redundant markdown and fix a few typos \(#1936\)](#) ([ce207e141](#))

IMPROVE THIS PAGE

By the **Hugo** Authors



[File an Issue](#) [Get Help](#) [Discuss Source Code](#)

[@GoHugoIO](#) [@spf13](#) [@bepsays](#)

[Menu](#)

[Docs Menu](#)



## Hugo Sponsors



The Hugo logos are copyright © Steve Francia 2013–2023.

The Hugo Gopher is based on an original work by Renée French.