

Cheatsheet User Guide

This document is a cheatsheet on how to use cheatsheets. Cheatsheets are Markdown files with the `.cheatmd` extension. You may also check the source of this document as a reference.

I'm a H2 title with 2 columns

I'm a H3 title

I'm a H4 title.

And this is a paragraph.

Code

```
# hello.exs
defmodule Greeter do
  def greet(name) do
    message = "Hello, " <> name <> "!"
    IO.puts message
  end
end

Greeter.greet("world")
```

Paragraphs

Paragraphs are also supported and can quote `code`.

Several paragraphs are visually distinct.

Lists

- Element 1

- Element 2

- `piece of code`

Table

Date (H4 header)	
EXAMPLE	OUTPUT
<code>%m/%d/%Y</code>	<code>06/05/2013</code>
<code>%A, %B %e, %Y</code>	<code>Sunday, June 5, 2013</code>
<code>%b %e %a</code>	<code>Jun 5 Sun</code>

Variants

Adding variants

```
## H2
{: .col-2}

### H3
{: .list-6}
```

- * example 1
- * example 2

Variants customize how your cheatsheet looks like via Markdown's inline attribute notation.

Header variants

H2

`.col-2`

two-columns

`.col-3`

three-columns

`.col-2-left`

two-columns (1/3 - 2/3)

H3

`.list-4`

four-columns for lists

`.list-6`

six-columns for lists

Code

Code with headings

<code>index.ex</code>
<pre>Path.join(["~", "foo"]) "~/foo"</pre>
<code>other.ex</code>
<pre>Path.join(["foo"]) "foo"</pre>
Code blocks can have headings.

Long lines

<pre>defmodule MyTracer do def trace({:remote_function, _meta, module, name, arity}, env) do IO.puts "#{env.file}:#{env.line} #{inspect(module)}.#{name}/#{arity}" :ok end def trace(_event, _env) do :ok end end</pre>
Long lines show scrollbars.

Line wrapping

<pre>defmodule MyTracer do def trace({:remote_function, _meta, module, name, arity}, env) do IO.puts "#{env.file}:#{env.line} #{inspect(module)}.#{name}/#{arity}" :ok end def trace(_event, _env) do :ok end end</pre>
Add <code>wrap</code> to wrap long lines.

Lists

Bullet lists

- This is

- a list

- with a few items

Ordered lists

1. I'm first;

2. You are second;

3. We are third;

Here's an extra paragraph after the list.

With headings and code links

Part 1

- `is_atom/1`

- `is_binary/1`

- `is_number/1`

Part 2

- `length/1`

- `elem/2`

List columns

Six columns

- One
 - Two
 - Three
 - Four
 - Five
 - Six
-
- Seven
 - Eight
 - Nine
 - Ten
 - Eleven

Add `{: .list-6}` after the H3 title to make large lists.

Four columns

- One
 - Two
 - Three
 - Four
-
- Five
 - Six
 - Seven
 - Eight
-
- Nine
 - Ten
 - Eleven

Add `{: .list-4}` after the H3 title to make large lists.

Full page

This is the default.

Half width

I make tables look nicer

Time	
EXAMPLE	OUTPUT
<code>%H:%M</code>	<code>23:05</code>
<code>%I:%M %p</code>	<code>11:05 PM</code>
Add <code>{: .width-50}</code> after the H2 title to use only half width.	

Column left reference

One

```
# hello.exs
defmodule Greeter do
  def greet(name) do
    message = "Hello, " <> name <> "!"
    IO.puts message
  end
end

Greeter.greet("world")
```

Three

- Yet

- Another

- List

Two

```
user = %{
  name: "John",
  city: "Melbourne"
}

IO.puts "Hello, " <> user.name
```

Four

I'm a grid, with 1/3 - 2/3 width.

Built using [ExDoc](#) (v0.29.0) for the [Elixir programming language](#)