Coloured Terminal Listings in remark
Challenge

How do we capture ANSI coloured console output and display it in remark?

```bash
josh@brick ~ $ ls -l --color
total 20363
drwxrwxr-x+ 96 root admin 3264 Feb 14 13:05 Applications
drwxrwxr-x 18 root admin 612 Aug 14 2011 Developer
drwxrwxr-t+ 63 root admin 2142 Apr 28 2012 Library
drwxr-xr-x  2 root wheel  68 Jul 10 2009 Network
drwxr-xr-x  4 root wheel 136 Jul 30 2011 System
drwxr-xr-x  6 root admin  204 Jun 27 2012 Users
drwxrwxrwt+  3 root admin 102 Feb 14 18:12 Volumes
drwxr-xr-x 39 root wheel 1326 Oct 23 19:41 bin
```

Solution

Use the `aha` Ansi HTML Adaptor to capture output and paste the result as code to be interpreted by the highlight.js engine used in remark.
The Code

The results produced by aha are translated via the highlight.js engine (using the [language definition](#)) and a small jQuery script to substitute the embedded style elements after the page has rendered.

To achieve this, add the following to the end of the remark presentation:

```html
...<script src="https://gnab.github.io/remark/downloads/remark-latest.min.js"></script>
<script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>
<script type="text/javascript">
  var hljs = remark.highlighter.engine;
</script>
<script src="terminal.language.js" type="text/javascript"></script>
<script type="text/javascript">
  var slideshow = remark.create({
    highlightStyle: 'monokai'
  });
  // extract the embedded styling from ansi spans
  $('code.terminal span.hljs-ansi').replaceWith(function(i, x) {
    return x.replace(/<\?\?(\w+).*?\?>/g, '<$1>'});
</script>
...
```

The terminal language definition assumes a prompt of the form:

```
user@host /path/x $
```

If you would like a different prompt style, get your regex on and alter it in the [language definition](#).
Worked Example to Capture Coloured Command Output (e.g. *git*)

1. Build and install **aha**: 

```
josh@brick ~/repos $ git clone https://github.com/theZiz/aha; cd aha
josh@brick ~/repos/aha $ make && cp aha /usr/local/bin
```

2. Capture output with **aha** (for dark background highlight.js styles such as solarized_dark, use **aha -b**):

```
josh@brick ~/repos/aha $ git log -2 --color | aha -b -n | pbcopy
```

To capture directly to clipboard, use:
- *pbcopy* for OS X
- *xsel --clipboard* for Linux
- *getclip* for Cygwin
Paste into remark document with the `header code` (see `source` for more details):

```terminal
josh@brick ~/repos/aha $ git log -2
<span style="color:olive;">commit 36b0a3af174e204c8d0a7a993ad467cd7be39bca</span>
Author: Ziz <zizsd1@googlemail.com>
Date:   Fri Aug  3 10:29:43 2012 +0200

    Small version changes, make clean, merging, etc.
...
...
```

Note: code must be in a [GFM](https://github.github.com/gfm) fenced block with "terminal" as the syntax-type as above.
The Result

Putting it all together, we have the desired result:

```bash
josh@brick ~/repos/aha $ git log -2
commit 36b0a3af174e204c8d0a7a993ad467cd7be39bca
Author: Ziz <zizsdl@googlemail.com>
Date:   Fri Aug 3 10:29:43 2012 +0200
   Small version changes, make clean, merging, etc.
commit 86abaa46c786c1e5dca33ad5907b8067d0e47a20
Merge: ed4f29b d0dc6c1
Author: Ziz <zizsdl@googlemail.com>
Date:   Fri Aug 3 10:28:59 2012 +0200
   Merge branch 'master' of github.com:theZiz/aha
```

Bold and Reverse

Bold and reverse work nicely, too (be careful of the background colour of your highlight.js theme):

```bash
josh@brick ~ $ zsh -c "print -P '%Bbold%Sreverse%b%s'"
bold reverse
```
The Colourful aha Help Message

The output of aha  -h give a good example of the results of this method:

```
josh@brick ~/repos/aha $ aha -h
Ansi Html Adapter Version 0.4.6.1
aha takes SGR-colored Input and prints W3C conform HTML-Code
use: aha <options> [-f file]
    aha (--help | -h [2])
aha reads the Input from a file or stdin and writes HTML-Code to stdout
options: --black,     -b: Black Background and White "standard color"
    --pink,      -p: Pink Background
    --iso X,   -i X: Uses ISO 8859-X instead of utf-8. X must be 1..16
    --title X, -t X: Gives the html output the title "X" instead of "stdin"
    or the filename
    --line-fix,  -l: Uses a fix for inputs using control sequences to
        change the cursor position like htop. It's a hot fix,
        it may not work with any program like htop. Example:
        echo q | htop | aha -l > htop.htm
    --word-wrap, -w: Wrap long lines in the html file. This works with
        CSS3 supporting browsers as well as many older ones.
    --no-header, -n: Don't include header into generated HTML,
        useful for inclusion in full HTML files.
Example: aha --help | aha --black > aha-help.htm
    Writes this help text to the file aha-help.htm

Copyleft Alexander Matthes aka Ziz 2012
    ziz SDL@gmail.com
    http://ziz.delphigl.com/tool_aha.php
This application is subject to the MPL or LGPL.
```