Stylesheet for GitHub Slides

DataONE Community Engagement & Outreach Working Group

It's Working!

Each lesson is a single markdown document that is converted to HTML by remark javascript.

Each slide starts with a `# title`, and ends with `---`:

```
# This is the Slide Title

This is the content of the slide.

---
# This is the Title of the Next Slide

This is the content of the next slide.
```

Spaces are important in Markdown. For example, if you are not seeing a new slide, then check that there are no spaces to the left of the slide terminator `---`.

General Info

Lesson title and metadata go in the `yaml` header:

```
---
title: "Lesson title"    <-- The Title of the Presentation
update: Sept. 20, 2016   <-- When the slides where last edited
layout: slides           <-- How the slides are rendered

---

# Title of the first content slide
```

Everything *below* that is markdown.

The `yaml` header is the first element of a presentation, and will give the title, date of latest update, and (possibly) other data. The line `layout: slides` is **very important** because it controls how the slides are rendered.

Markdown 101

- `*italics*` is *italics*
- `**bold**` is **bold**
- `***bold italics***` is ***bold italics***
- `[DataONE](https://www.dataone.org)` will appear as DataONE

```
- List item 1
- List item 2
  - Nested list item

1. Enumerated list
2. Second item
  - We can mix both
```

- List item 1
- List item 2
    - Nested list item

1. Enumerated list
2. Second item
    - We can mix both

See the remark wiki for more.

Headings

```
# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6
```

Heading 1
Heading 2
Heading 3
Heading 4
Heading 5
Heading 6

Heading levels are specified by the number of # at the beginning of a line. Six levels of headings are supported.

Quotes

```
> Text goes here
```

> *Text goes here*

Syntax Highlighting

```
~~~ R
<your code here>
~~~
```

renders as

```r
random_thing <- function(x, r, ...) {
  return(r(x, ...))
}

plot(random_thing(100, runif))
```

You can also put code inline, using backticks:

```
This is `inline` code.
```

Tables

Markdown can do tables:

```
|   Animal | Diet       | Fuzzy? |
|---------:|:-----------|:------:|
| Hedgehog | rings      |   no   |
|   Racoon | garbage    |  meh   |
|      Cat | hairballs  |  yup   |
```

This renders as:

Animal	Diet	Fuzzy?
Hedgehog	rings	no
Racoon	garbage	meh
Cat	hairballs	yup

DataONE

Notes

Notes are everything that comes below ???. Press **P** to toggle presenter mode.

```
Notes are everything that comes below `???`.
Press **P** to toggle presenter mode.

???

These are the notes, and *they can* be in
markdown too.
```

Columns

It is possible to have columns or various widths in the presentation. The columns should be wrapped the following way:

```
.one-third[
  content
]
```

You can chain columns in any way you want, *e.g.* 1/3 then 2/3, 1/4 then 1/2 then 1/2. As long as it sums to one, it's fine.

Possible values

- `.one-third`
- `.two-third`
- `.one-half`
- `.one-fourth`
- `.three-fourth`
- `.full-width`

Nested Columns

This is a three-fourth column, and splitted in two.

It works!

```
for (i in c(1:10)) {
  print(i)
}
```

```
for i in 1:10
  print(i)
end
```

Then the column resumes after the split.

Image Captions

```
![D. Lafrenière et al., ApJ Letters]
(/dataone_lessons/lessons/00_markdown/images/data-loss.jpg)
*D. Lafrenière et al., ApJ Letters*
```

D. Lafrenière et al., ApJ Letters