

Moving the lessons to github

DataONE Community Engagement & Outreach

It's working!

- Each lesson is its own `lesson_name.md` file in the github repository
- The maintainer ensures that the github pages are built from the master branch
- That's it

All slides start with a title, and end with a `---`:

```
# This is a slide title
```

This is the content of the slide.

General info

Lesson title and metadata go in the yaml header:

```
---  
title: "Lesson title"  
update: Sept. 20, 2016  
layout: slides  
---
```

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 allows to render the slides themselves.

Everything *below* that is markdown.

Markdown 101

- `*italics*` is *italics*
- `**bold**` is **bold**
- `***bold italics***` is ***bold italics***

Lists:

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

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

Quotes

> Text goes here

Text goes here

Syntax highlighting

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

renders as

```
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

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 of 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



CC image by momboleum on Flickr



CC image by Sharyn Morrow on Flickr

D. Lafrenière et al., ApJ Letters