


Lesson Title

Content from [introduction](#)

Last updated on 2024-05-31 | [Edit this page](#) 

[Download Chapter PDF](#)

OVERVIEW

Questions

- How do you write a lesson using R Markdown and [sandpaper](#)?

Objectives

- Explain how to use markdown with the new lesson template
- Demonstrate how to include pieces of code, figures, and nested challenge blocks

Introduction

This is a lesson created via The Carpentries Workbench. It is written in [Pandoc-flavored Markdown](#) for static files (with extension `.md`) and [R Markdown](#) for dynamic files that can render code into output (with extension `.Rmd`). Please refer to the [Introduction to The Carpentries Workbench](#) for full documentation.

Questions and objectives

What you need to know is that there are three sections required for a valid Carpentries lesson template:

1. questions are displayed at the beginning of the episode to prime the learner for the content.
2. objectives are the learning objectives for an episode displayed with the questions.
3. keypoints are displayed at the end of the episode to reinforce the objectives.

CHALLENGE 1: CAN YOU DO IT?

What is the output of this command?

```
paste("This", "new", "lesson", "looks", "good")
```

R < >

Solution

```
[1] "This new lesson looks good"
```

OUTPUT < >

CHALLENGE 2: HOW DO YOU NEST SOLUTIONS WITHIN CHALLENGE BLOCKS?

Solution

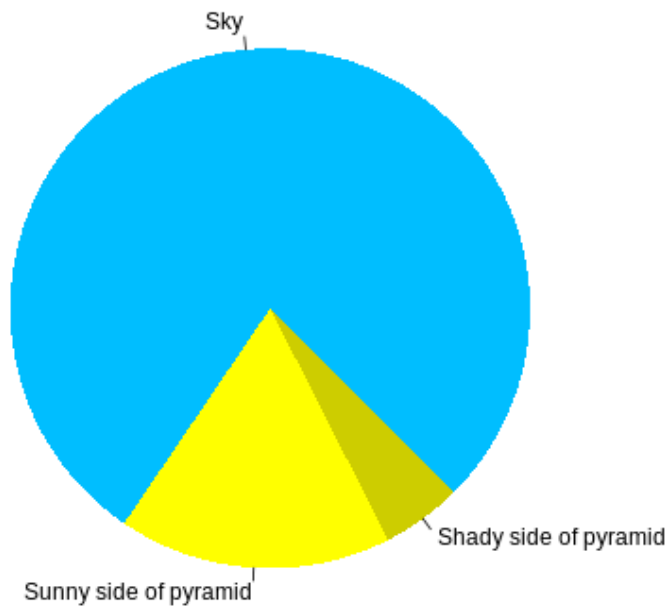
You can add a line with at least three colons and a `solution` tag.

Figures

Including figures from R Markdown

You can include figures generated from R Markdown:

```
pie(  
  c(Sky = 78, "Sunny side of pyramid" = 17, "Shady side of pyramid" = 5),  
  init.angle = 315,  
  col = c("deepskyblue", "yellow", "yellow3"),  
  border = FALSE  
)
```



Sun arise each and every morning

Including static figures

Or you can use pandoc markdown for static figures with the following syntax:

![optional caption that appears below the figure](figure url){alt='alt text for accessibility purposes'}



You belong in The Carpentries!

Math

One of our episodes contains *L^AT_EX* equations when describing how to create dynamic reports with {knitr}, so we now use mathjax to describe this:

`\alpha = \dfrac{1}{(1 - \beta)^2}` becomes: $\alpha = \frac{1}{(1 - \beta)^2}$

Cool, right?

KEY POINTS

- Use `.md` files for episodes when you want static content
- Use `.Rmd` files for episodes when you need to generate output
- Run `sandpaper::check_lesson()` to identify any issues with your lesson
- Run `sandpaper::build_lesson()` to preview your lesson locally

Content from [Not yet released](#)

Last updated on 2024-05-31 | [Edit this page](#) 

[Download Chapter PDF](#)

OVERVIEW

Questions

- How do you write a lesson using R Markdown and [sandpaper](#)?
-

Objectives

- Explain how to use markdown with the new lesson template
- Demonstrate how to include pieces of code, figures, and nested challenge blocks

Introduction

This is a lesson created via The Carpentries Workbench. It is written in [Pandoc-flavored Markdown](#) for static files (with extension `.md`) and [R Markdown](#) for dynamic files that can render code into output (with extension `.Rmd`). Please refer to the [Introduction to The Carpentries Workbench](#) for full documentation.

What you need to know is that there are three sections required for a valid Carpentries lesson template:

1. `questions` are displayed at the beginning of the episode to prime the learner for the content.
2. `objectives` are the learning objectives for an episode displayed with the questions.
3. `keypoints` are displayed at the end of the episode to reinforce the objectives.

CHALLENGE 1: CAN YOU DO IT?

What is the output of this command?

```
paste("This", "new", "lesson", "looks", "good")
```

R < >

Solution

```
[1] "This new lesson looks good"
```

OUTPUT < >

CHALLENGE 2: HOW DO YOU NEST SOLUTIONS WITHIN CHALLENGE BLOCKS?

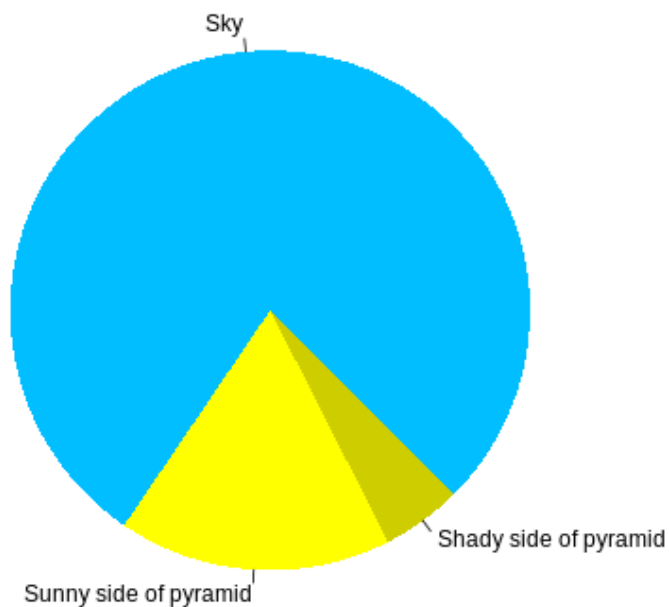
Solution

You can add a line with at least three colons and a `solution` tag.

Figures

You can include figures generated from R Markdown:

```
R < >  
pie(  
  c(Sky = 78, "Sunny side of pyramid" = 17, "Shady side of pyramid" = 5),  
  init.angle = 315,  
  col = c("deepskyblue", "yellow", "yellow3"),  
  border = FALSE  
)
```



Sun arise each and every morning

Or you can use pandoc markdown for static figures with the following syntax:

```
![optional caption that appears below the figure](figure url){alt='alt  
text for accessibility purposes'}
```



You belong in The Carpentries!

Math

One of our episodes contains *L^AT_EX* equations when describing how to create dynamic reports with {knitr}, so we now use mathjax to describe this:


`\alpha = \dfrac{1}{(1 - \beta)^2}` becomes: $\alpha = \frac{1}{(1 - \beta)^2}$

Cool, right?

KEY POINTS

- Use `.md` files for episodes when you want static content
- Use `.Rmd` files for episodes when you need to generate output
- Run `sandpaper::check_lesson()` to identify any issues with your lesson
- Run `sandpaper::build_lesson()` to preview your lesson locally

Content from [Also not yet released](#)

Last updated on 2024-05-31 | [Edit this page](#) 

[Download Chapter PDF](#)

OVERVIEW

Questions

- How do you write a lesson using R Markdown and [sandpaper](#)?
-

Objectives

- Explain how to use markdown with the new lesson template
- Demonstrate how to include pieces of code, figures, and nested challenge blocks

Introduction

This is a lesson created via The Carpentries Workbench. It is written in [Pandoc-flavored Markdown](#) for static files (with extension `.md`) and [R Markdown](#) for dynamic files that can render code into output (with extension `.Rmd`). Please refer to the [Introduction to The Carpentries Workbench](#) for full documentation.

What you need to know is that there are three sections required for a valid Carpentries lesson template:

1. `questions` are displayed at the beginning of the episode to prime the learner for the content.
2. `objectives` are the learning objectives for an episode displayed with the questions.
3. `keypoints` are displayed at the end of the episode to reinforce the objectives.

CHALLENGE 1: CAN YOU DO IT?

What is the output of this command?

```
paste("This", "new", "lesson", "looks", "good")
```

R < >

Solution

```
[1] "This new lesson looks good"
```

OUTPUT < >

CHALLENGE 2: HOW DO YOU NEST SOLUTIONS WITHIN CHALLENGE BLOCKS?

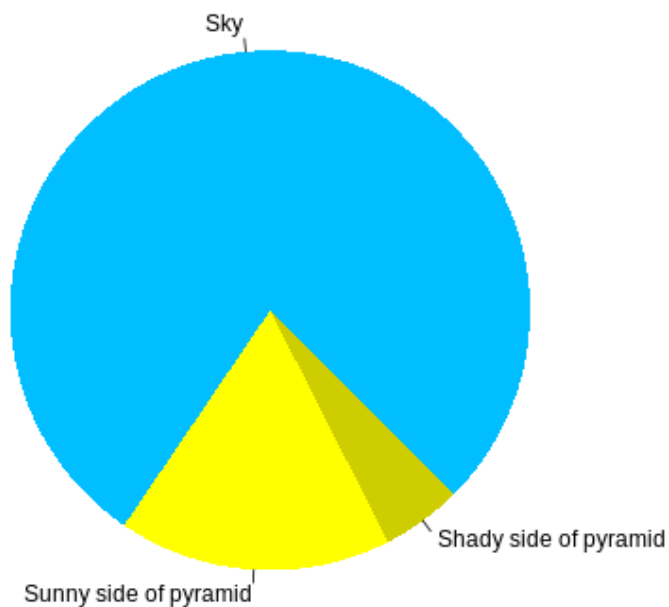
Solution

You can add a line with at least three colons and a `solution` tag.

Figures

You can include figures generated from R Markdown:

```
R < >  
pie(  
  c(Sky = 78, "Sunny side of pyramid" = 17, "Shady side of pyramid" = 5),  
  init.angle = 315,  
  col = c("deepskyblue", "yellow", "yellow3"),  
  border = FALSE  
)
```



Sun arise each and every morning

Or you can use pandoc markdown for static figures with the following syntax:

![optional caption that appears below the figure](figure url){alt='alt text for accessibility purposes'}



You belong in The Carpentries!

Math

One of our episodes contains *L^AT_EX* equations when describing how to create dynamic reports with {knitr}, so we now use mathjax to describe this:

`$\alpha = \dfrac{1}{(1 - \beta)^2}$` becomes: $\alpha = \frac{1}{(1 - \beta)^2}$

Cool, right?

KEY POINTS

- Use `.md` files for episodes when you want static content
- Use `.Rmd` files for episodes when you need to generate output
- Run `sandpaper::check_lesson()` to identify any issues with your lesson
- Run `sandpaper::build_lesson()` to preview your lesson locally