Introduction to Data Science Using R

Lecture 3

Ben Rogers, Sept 13, 2022

Review

Last Class

- Parts of a data science project
- Types of data science questions
- Introduction to R
  - Data types
  - Objects
  - R as a calculator
  - Relational/logical operators
  - Vectors

Today’s lecture

- Lists
- Dataframes
- Functions
- Scripts
- Introduction to R Markdown

Lists

- Lists are objects which contain elements of different types — strings, numbers, vectors, or even other lists.

```
list1 <-  list(name="Mike", gender="M", company="A")
list(x = c(1,2,3), gender="M", company="A")
list1$name
```

Data Frames

- Data frames are essentially rectangular spreadsheets
- Rows correspond to observations
- Columns correspond to variables that describe the observations

```
name <- c("Mike", "Lucy", "John")
age <- c(20, 25, 30)
student <- c(TRUE, FALSE, TRUE)
df <-  data.frame(name, age, student)
```

Data Frames

Subsetting

- You can subset a data frame similar to a vector, but with a data frame there are 2 dimensions:

```
df[2,2]
df[,2]
df[2,]

df[ df$age<25,]
```

Functions

- Functions perform tasks in R
- They take inputs called arguments and return outputs.
- Often you can either manually specify a function’s arguments or use the function’s default values
- R has a large collection of built-in functions that are called like this:
  ```
  function_names(arg1 = val1, arg2 = val2, … )
  ```

Function example

Sequence function

- The function ‘seq()’ makes regular sequences of numbers

```
> seq(1,10)

 [1]  1  2  3  4  5  6  7  8  9 10

> seq(1,4)

[1] 1 2 3 4

> seq(from = 4, to = 1)

[1] 4 3 2 1
```

Note our use of ‘=‘ as a syntax token to pass values to named arguments

Something to watch out for

- Quotation marks and parnetheses must always come in a pair.
- RStudio does its best to help you, but it is still possible to mess up and end up with a mismatch.
- If this happens, R will show you the continuation character ‘+’

```
> x <- seq(1,3

+
```

The ‘+’ tells you that R is waiting for more input; it does not think you are done yet. Either add missing character, or press ESCAPE to abort the expression and try again.

Clearing your environment

- Look at your environment in the upper right pane. You can see all of the objects that you have created.
- You can remove objects from the environment using rm(), and remove all the objects using

```
rm(list=ls())
```

You can clear the console using Ctrl + l

Your turn

- Press Alt + Shift + K (option + shift + k on Mac). What happens?
- Here are some examples that appear in the help page of seq(), explain the results:
  - seq(0, 1, length.out = 11)
  - seq(1, 6, by = 3)
- Generate the sequence 1, 1.5, 2, 2.5, …, 5
- Calculate $1^3 + 2^3 + \cdots + 20^3$
- Calculate $(1 + 2 + \cdots + 20)^2$

Some other useful functions

- sqrt() computes the square root
- log() and exp() compute the logarithm and exponential
- mean() computes the mean of a vector
- min() and max() compute the minimum and maximum
- range() gives the range

Scripts

- To have more room to work, we can use the script editor.
- Open it up either by clicking the File menu -> New File -> R script, or by using the keyboard shortcut Cmd/Ctrl + Shift + N
    - Now you'll see four panes
- The script editor is a great place to put code you care about
    - You can experiment in the console, but once you have written code that works and does what you want, put it in the script editor.
- RStudio will automatically save the contents of the script editor when you quit RStudio, and will automatically load it when you re-open (You should still regularly save your scripts!!)
- The script editor will also highlight syntax errors with a red squiggly line and a cross in the side bar
    - Hover mouse over the cross to see what the problem is.

Your turn

R script

- Create a script in RStudio and do the following step by step

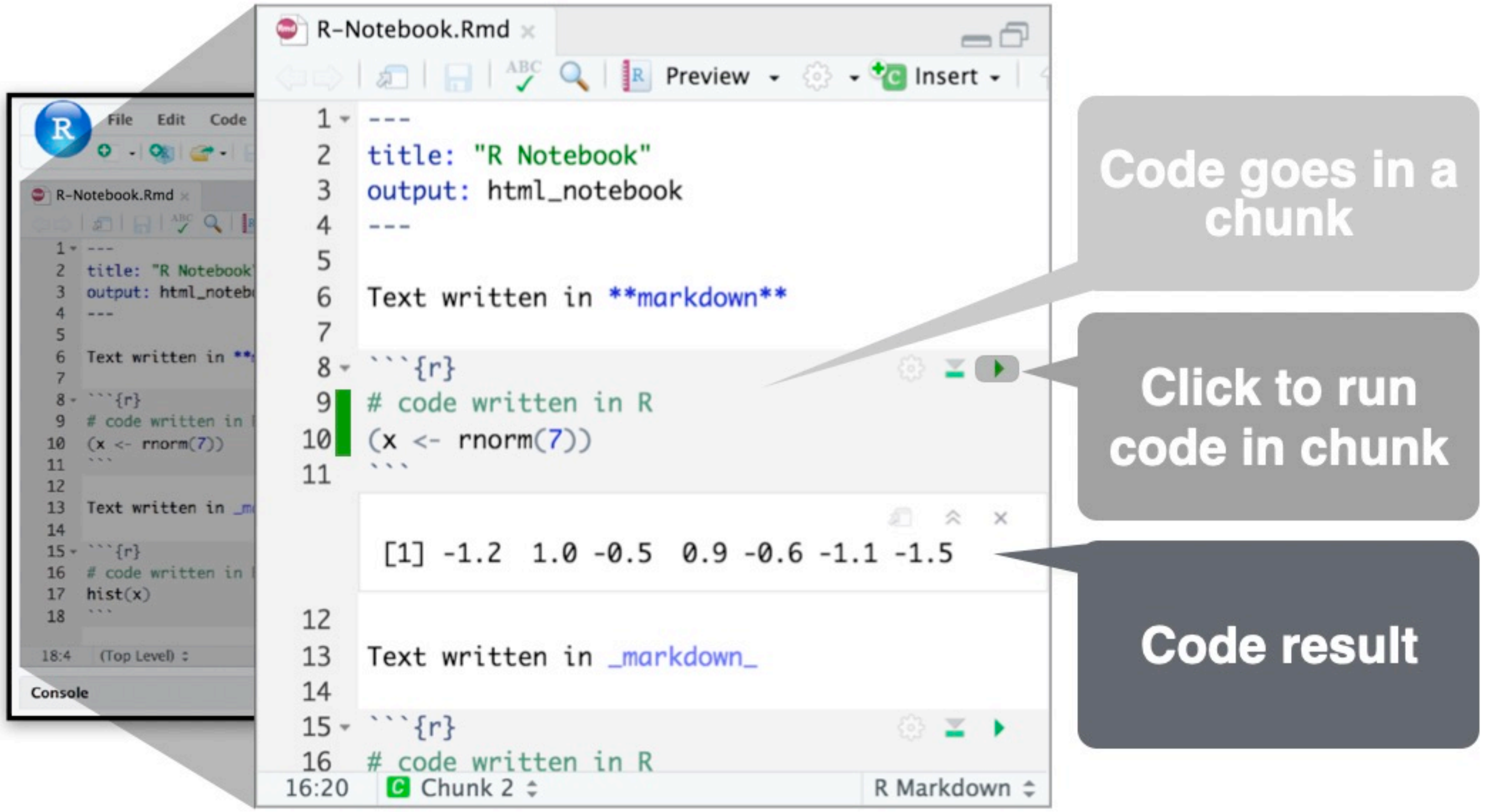
1. Calculate $\frac{\pi^2}{6}$ and save it in x.
2. Calculate $\frac{1}{1} + \frac{1}{2^2} + \cdots + \frac{1}{10^2}$
3. Take the difference between the two numbers obtained from the last two steps, save it in d1
4. Replace 10 with 20 in step 2 and do step 3 again, save the difference in d3
5. Replace 20 with larger numbers, do the same calculation and save them in d3, d4, d5, …
6. Combine d1, d2, d3, … in a vector d, did you find anything interesting in d?

Errors, warnings and messages

- R reports errors, warnings and messages in a glaring red font, which makes it seem like it is scolding you. R will show red text in the console pane in three different situations.
  - **Errors**: Generally when there's an error, the code will not run
  - **Warnings**: Generally, when there's a warning, your code will still work, but there may be something wrong.
  - **Messages**: When the red text doesn't start with either "Error" or "Warning", it's just a friendly message

Introduction to Markdown

R Markdown



~~~
---
title: "R Notebook"
output: html_notebook
---

Text written in **markdown**

```{r}
# code written in R
(x <- rnorm(7))
```
~~~

```
[1] -1.2  1.0 -0.5  0.9 -0.6 -1.1 -1.5
```

~~~
Text written in _markdown_

```{r}
# code written in R
~~~

Code goes in a chunk

Click to run code in chunk

Code result

R Markdown

- R Markdown files are designed to be used in three ways:

    1. For communicating to decision makers, who want to focus on the conclusions, not the code behind the analysis.

    2. For collaborating with other data scientists, who are interested in both your conclusions, and how you reached them (i.e. the code).

    3. As and environment in which to do data science, as a modern day lab notebook, where you can capture not only what you did, but also what you were thinking.

R Markdown tools and references

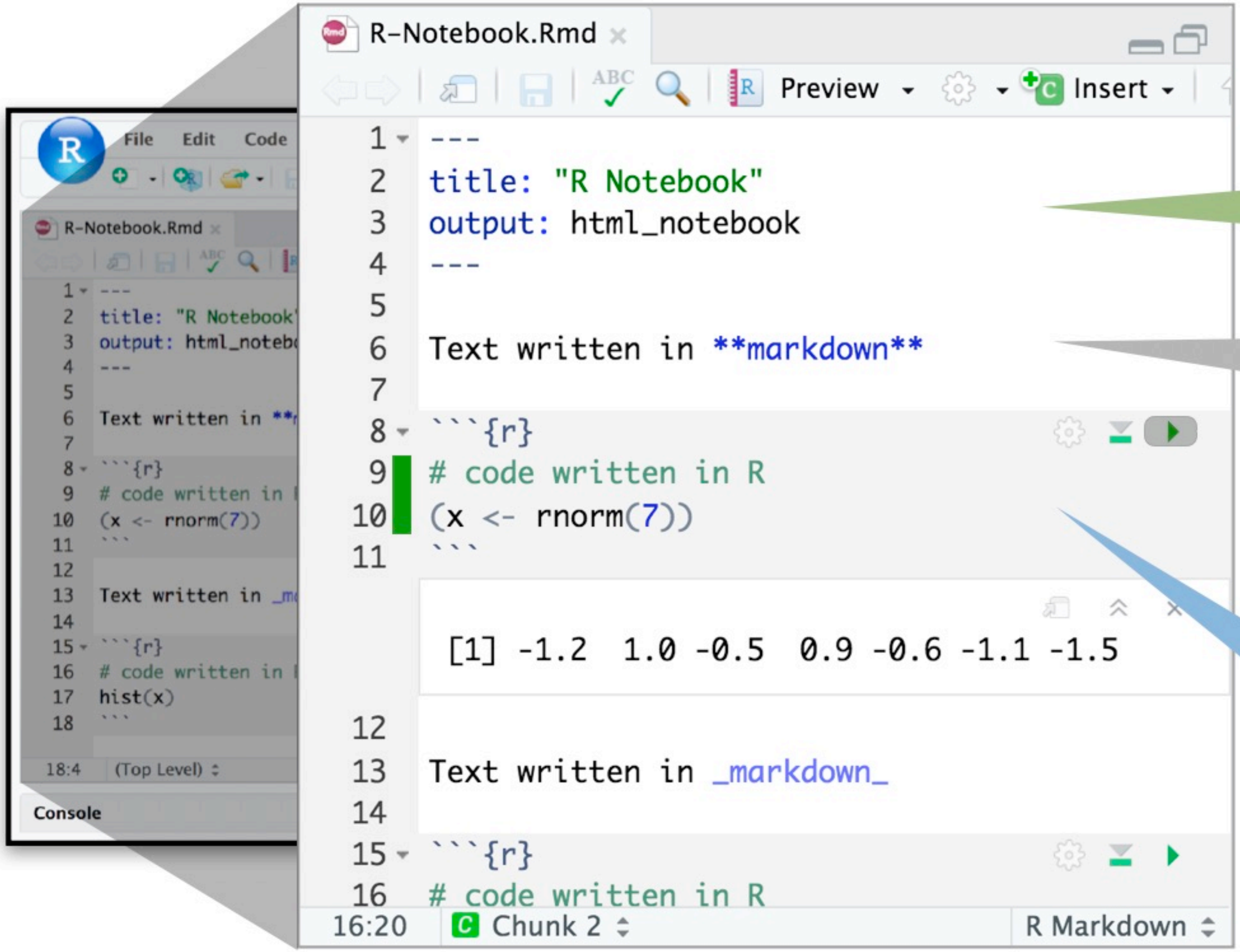
- R Markdown integrates a number of R packages and external tools. When you use R Markdown, keep these resources close at hand:
  - R Markdown Cheat Sheet: Help -> Cheatsheets -> R Markdown Cheat Sheet
  - R Markdown Reference Guide: Help -> Cheatsheets -> R Markdown Reference Guide

Getting started in R Markdown

- To use R Markdown, you will need to install the rmarkdown package:
    - install.packages(“rmarkdown”)

Create an R Markdown File

Plain text file with 3 types of content:



~~~
---
title: "R Notebook"
output: html_notebook
---

Text written in **markdown**

```{r}
# code written in R
(x <- rnorm(7))
```

[1] -1.2  1.0 -0.5  0.9 -0.6 -1.1 -1.5


Text written in _markdown_

```{r}
# code written in R
hist(x)
```
~~~

A YAML header surrounded by — — —

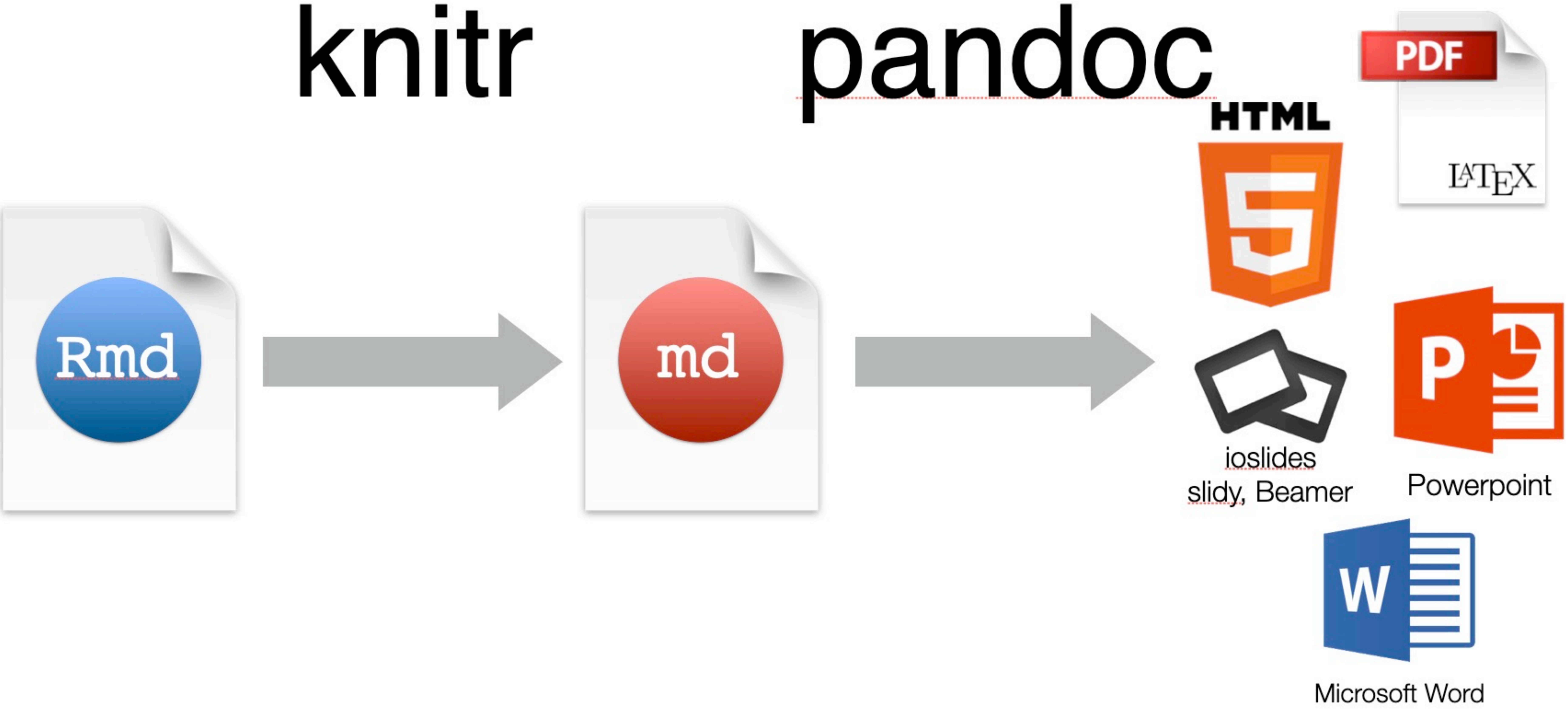
Text in markdown

Code chunks surrounded by ```

R Markdown

- You can run each code chunk by clicking the Run icon, or by pressing Cmd/Ctrl + Shift + Enter
    - RStudio executes the code and displays the results in line with the code.
- To produce a complete report containing all text, code, and results, click “Knit” or press Cmd/Ctrl + Shift + K.
    - This will display the report in the viewer pane, and create a self-contained HTML file that you can share with others.

Knitting an R Markdown file



knitr

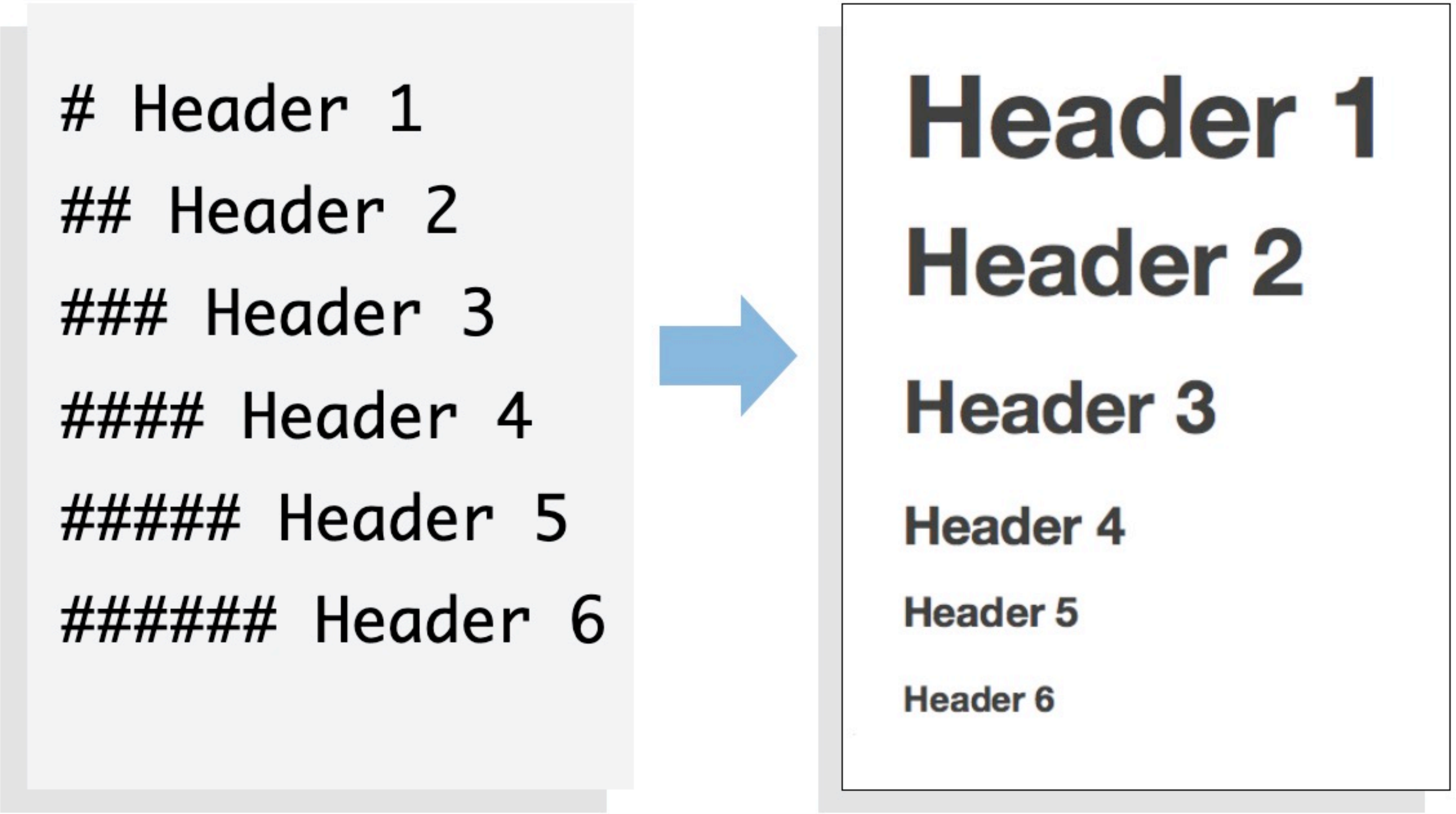
pandoc

ioslides
slidy, Beamer

Powerpoint

Microsoft Word

Text formatting with Markdown



```
# Header 1
## Header 2
### Header 3
#### Header 4
##### Header 5
###### Header 6
```

Header 1

Header 2

Header 3

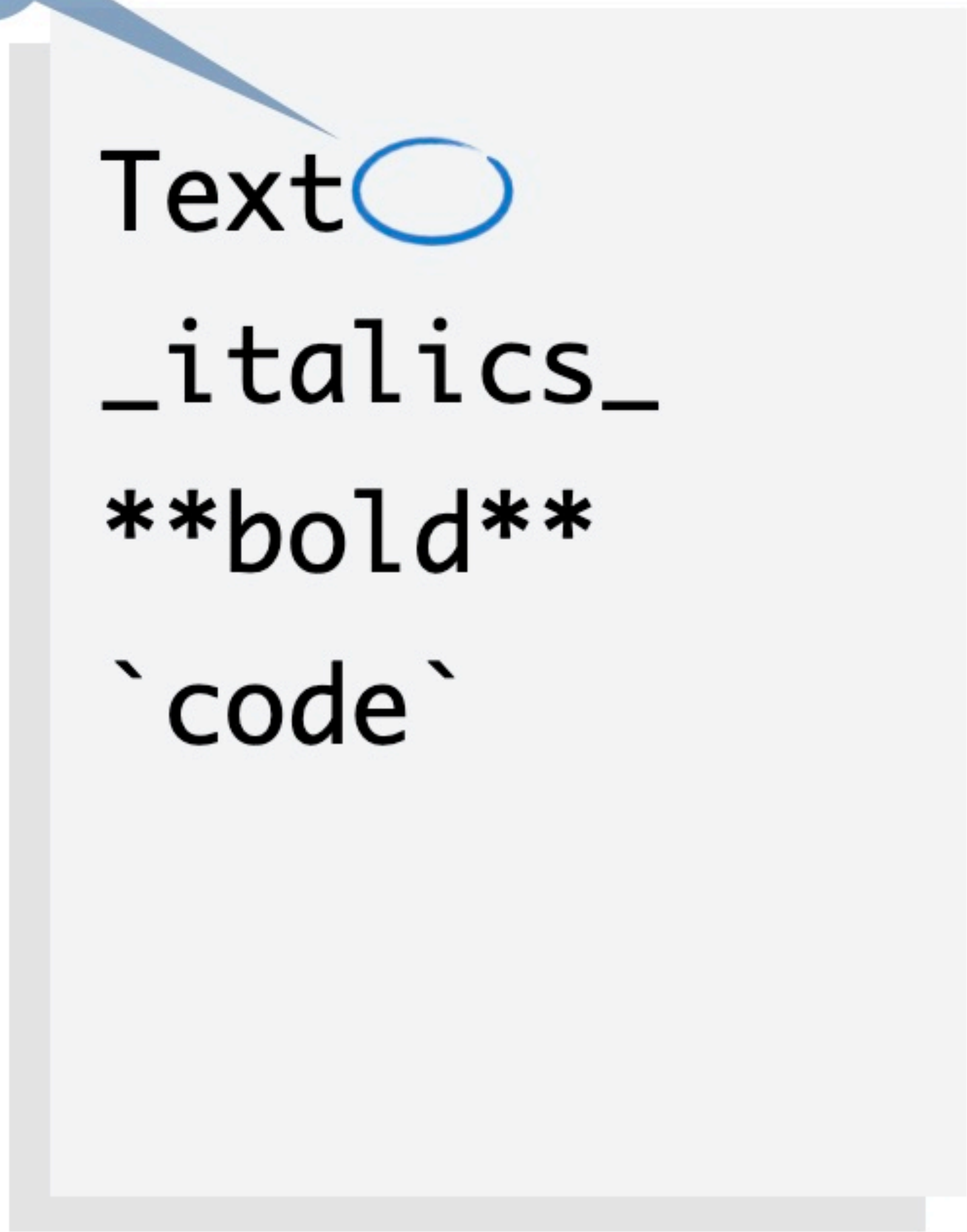
Header 4

Header 5

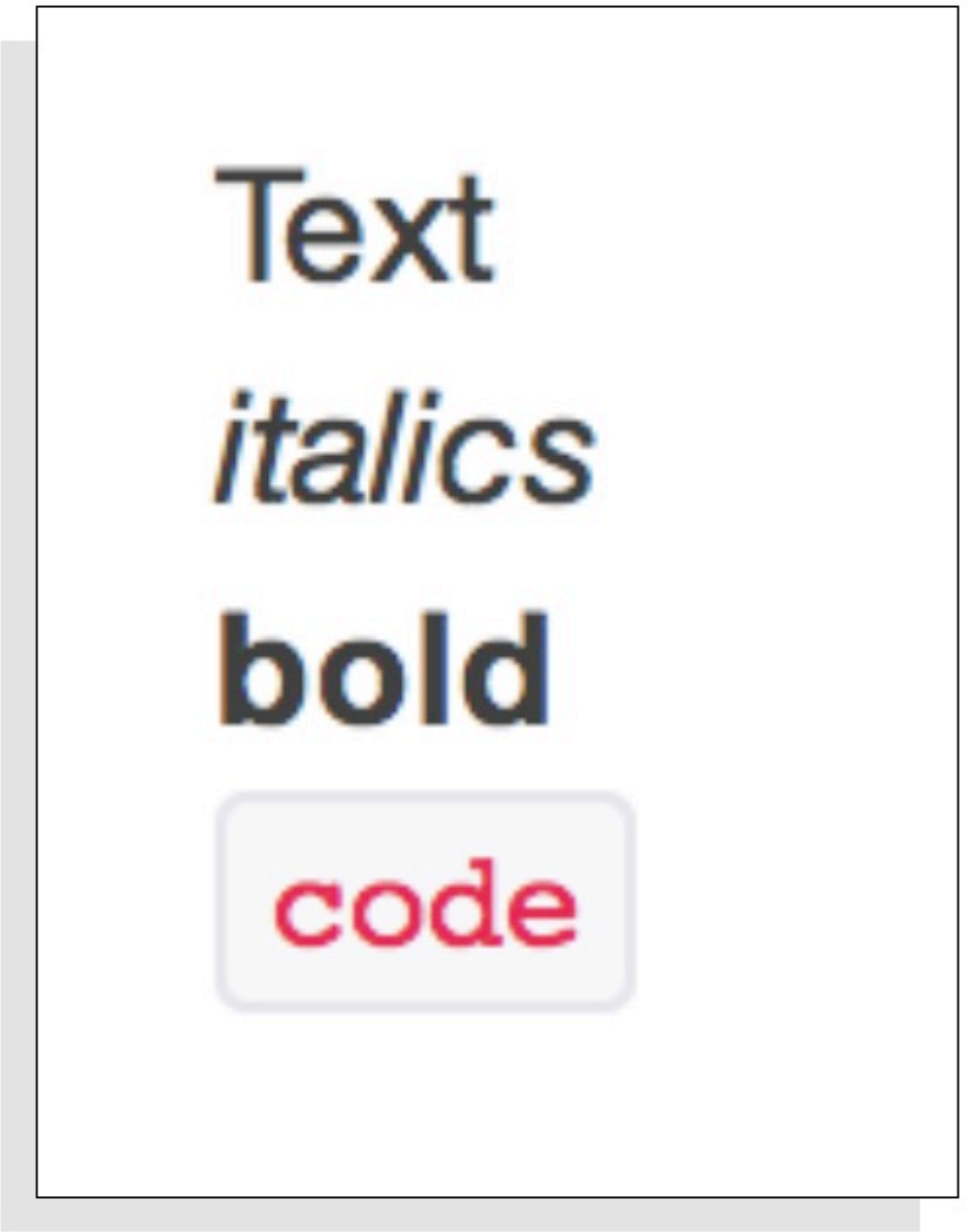
Header 6

Text

Add two spaces at the end of a line to start a



```
Text  
_italics_
**bold**
`code`
```



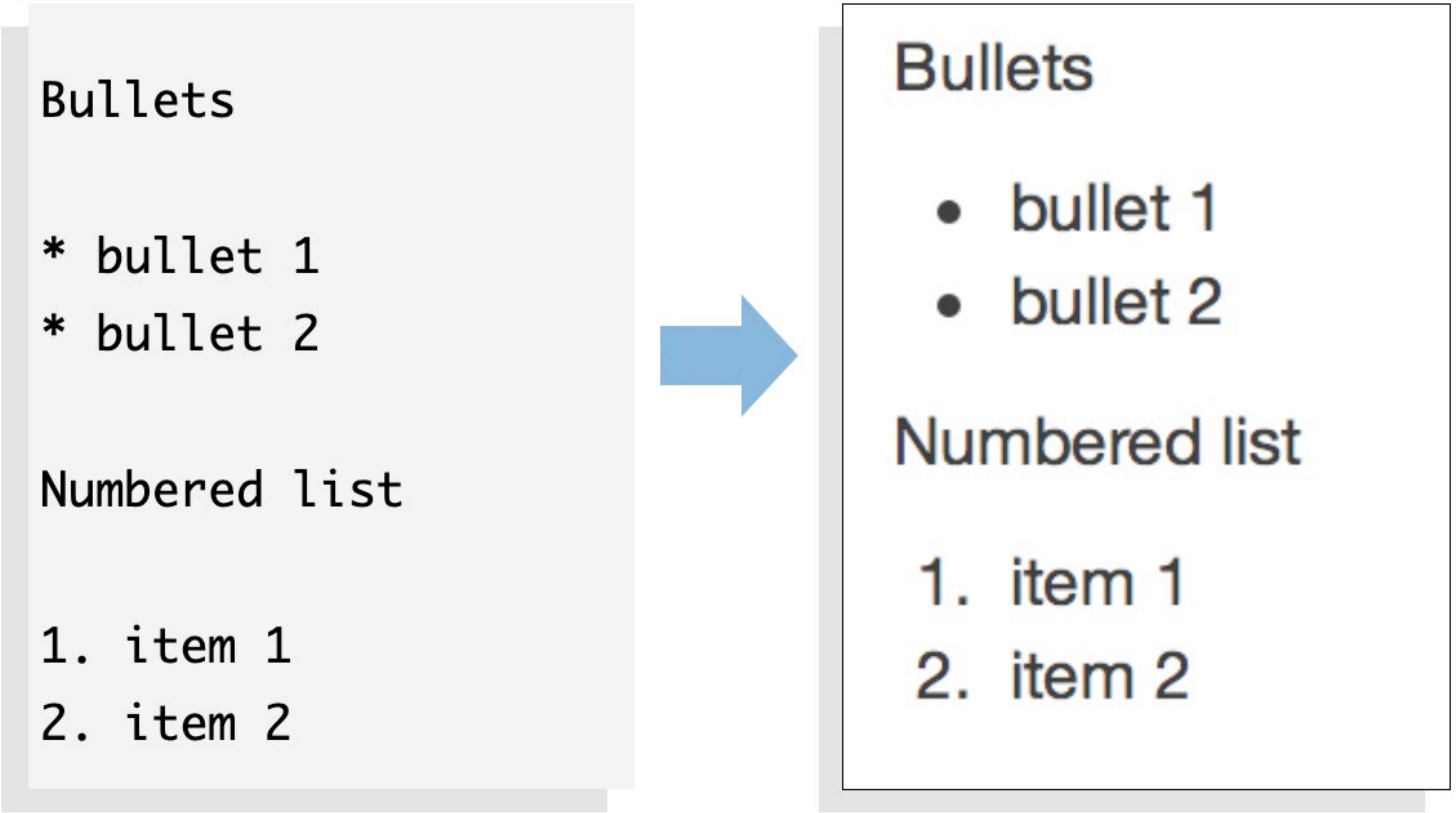
Text

italics

bold

`code`

Lists



```
Bullets

* bullet 1
* bullet 2

Numbered list

1. item 1
2. item 2
```

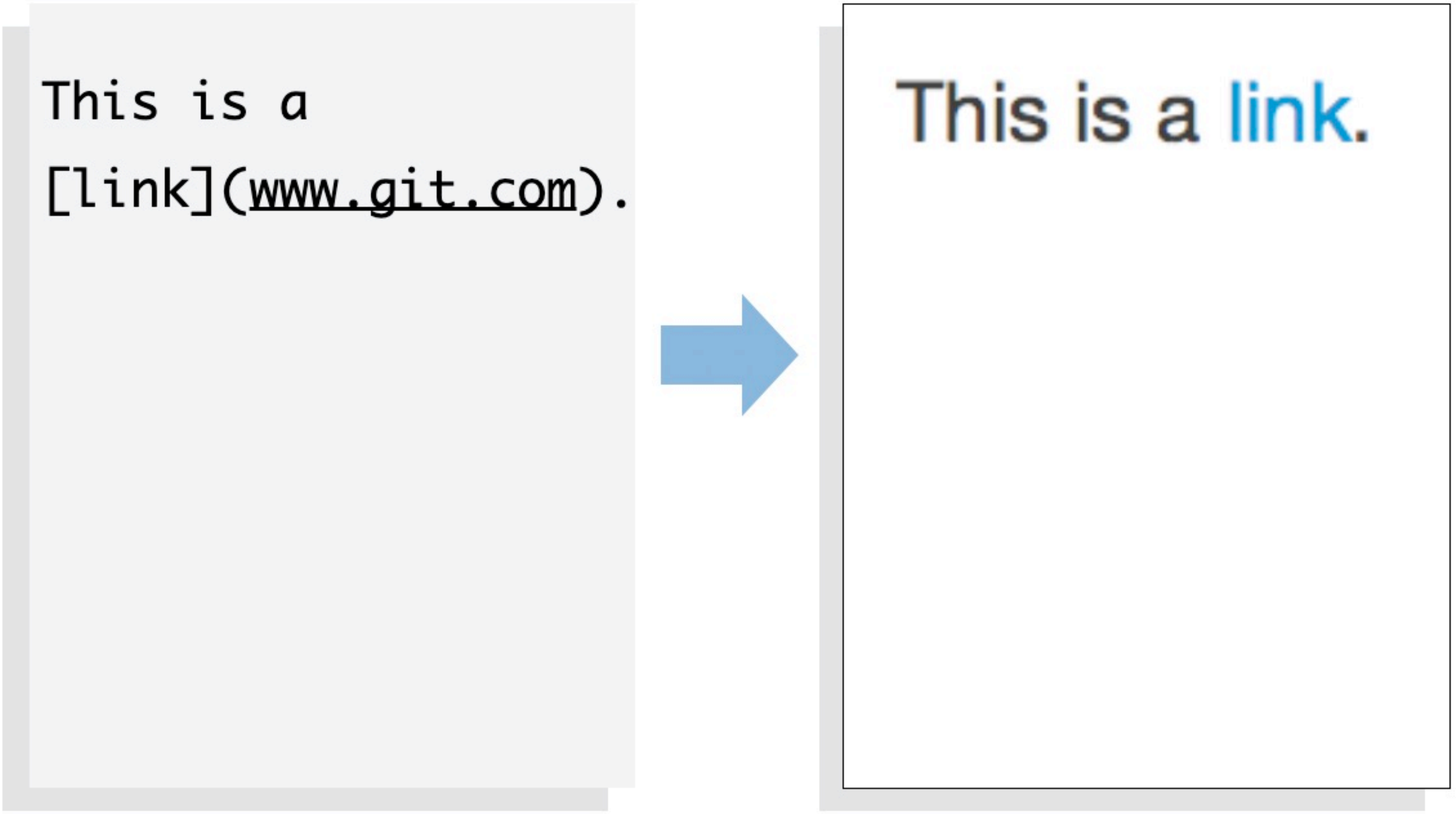
Bullets

- bullet 1
- bullet 2

Numbered list

1. item 1
2. item 2

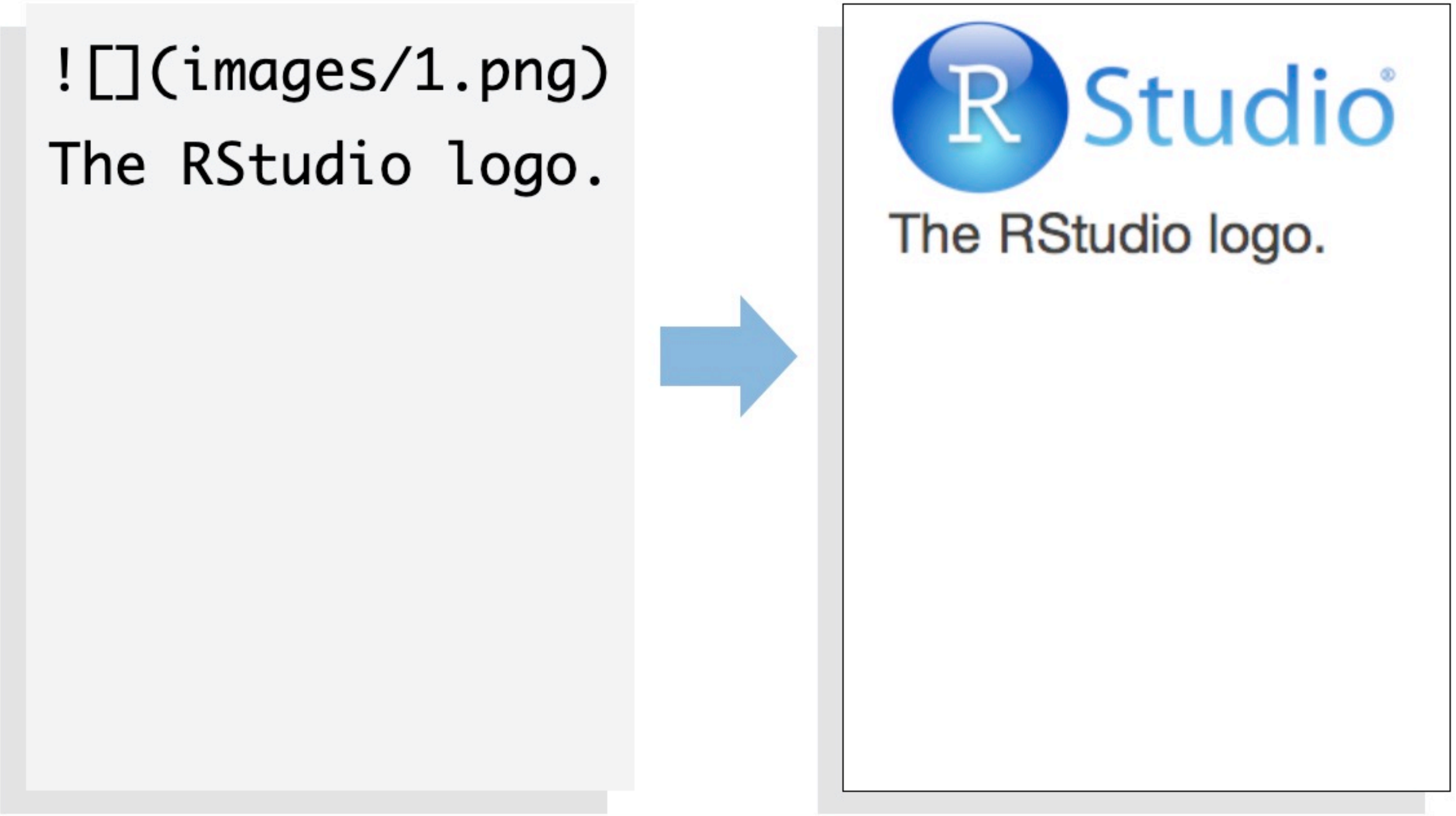
Links



```
This is a
[link](www.git.com).
```

This is a link.

Images



```
![](images/1.png)
The RStudio logo.
```

The RStudio logo.

Tables

```
First Header  | Second Header
------------- | -------------
Content Cell  | Content Cell
Content Cell  | Content Cell
```

Your turn

Using the R Markdown Quick Reference…

- Add a block quote.
- Add a horizontal rule.
- Add a subscript.

That’s all for today!