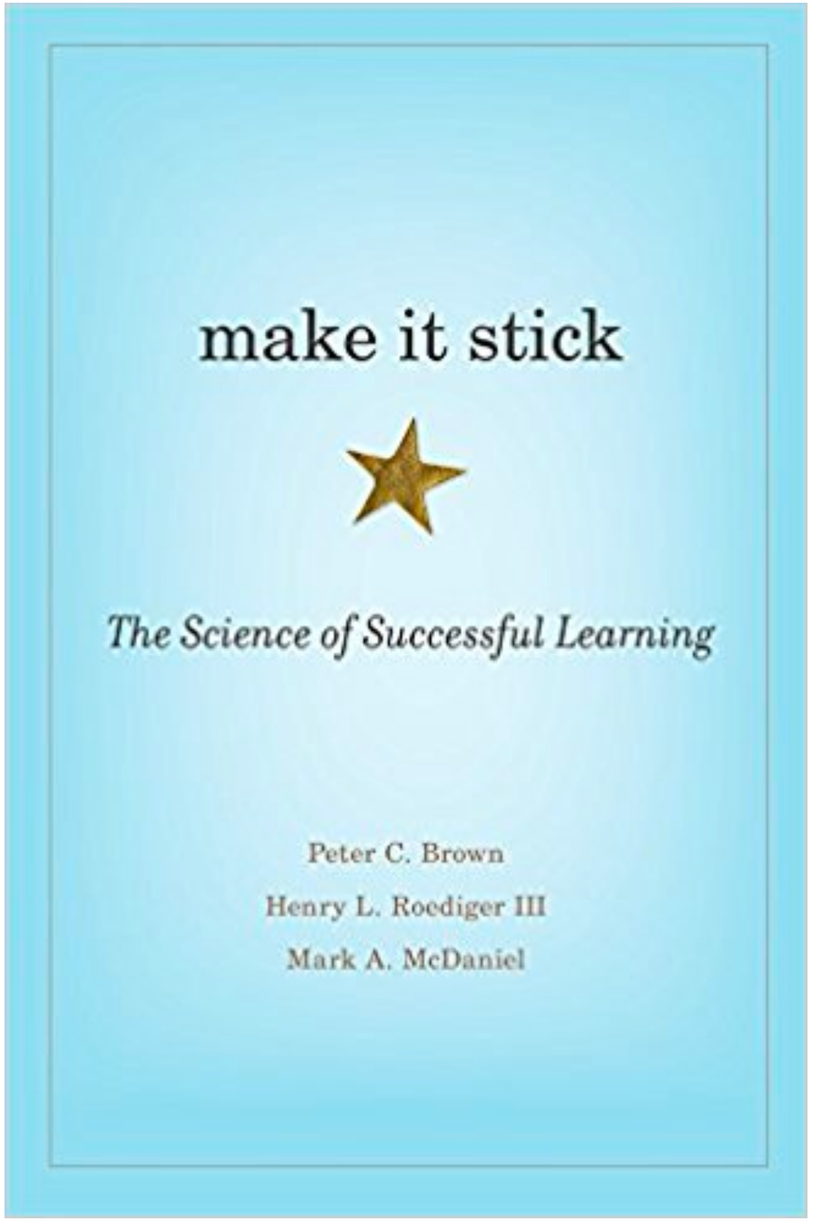
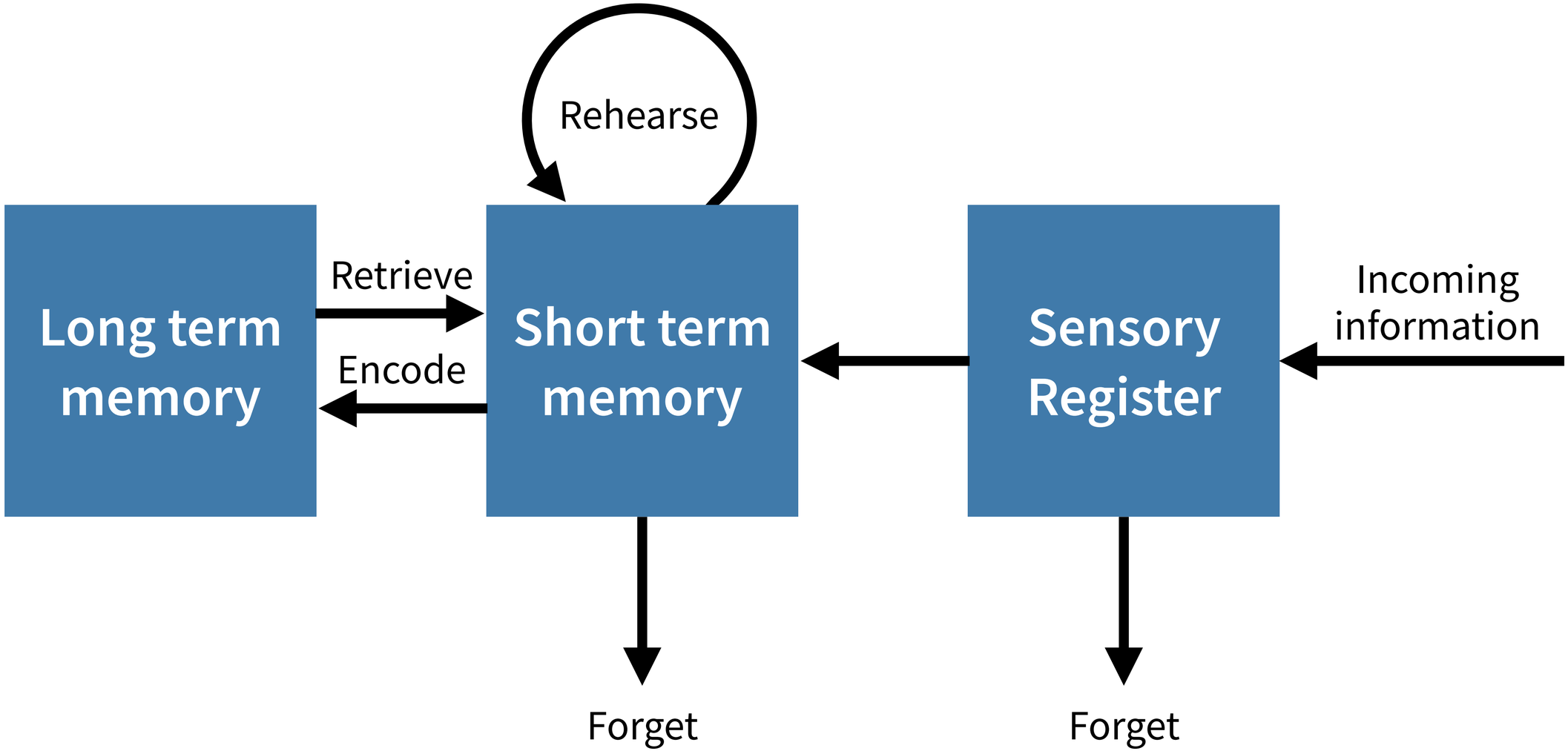
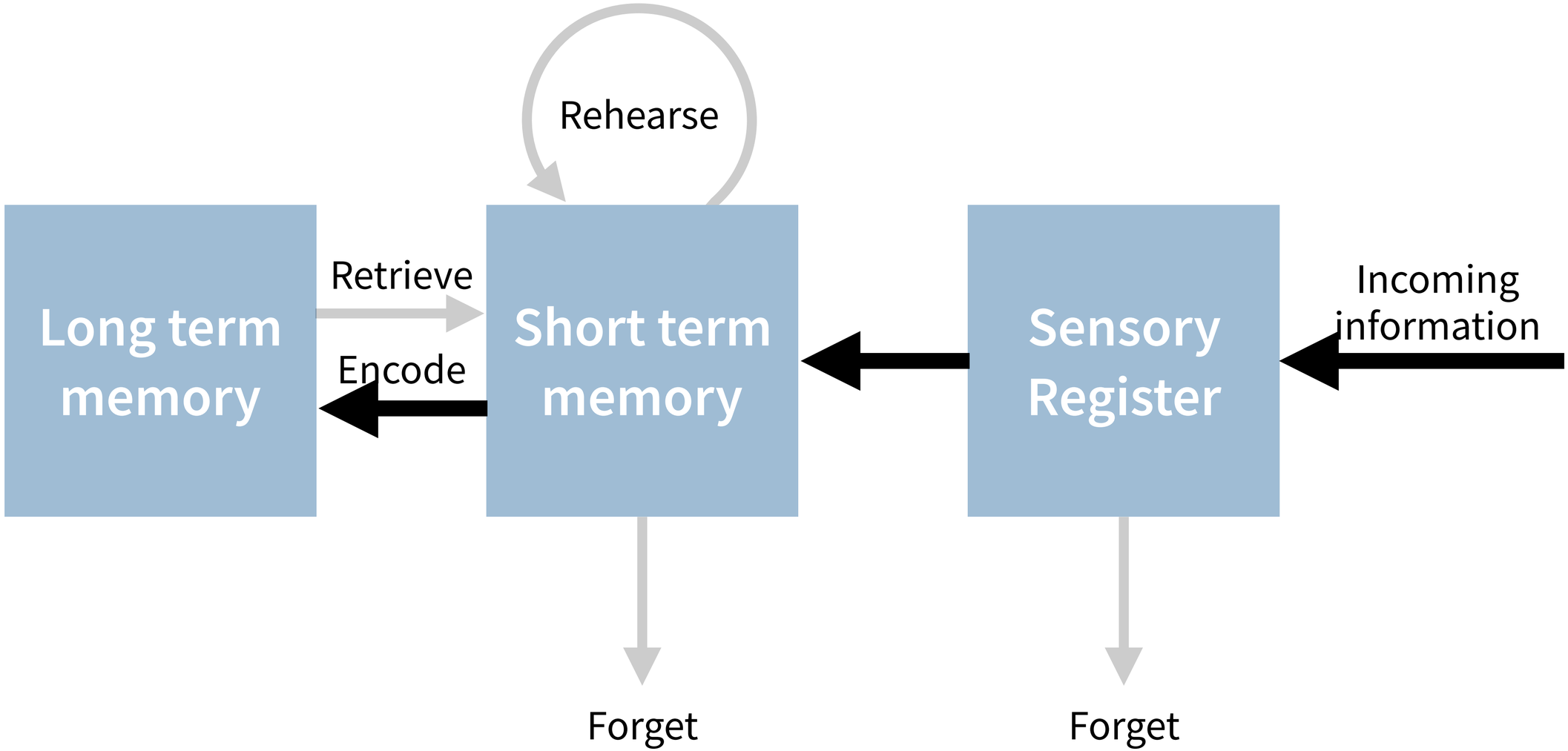
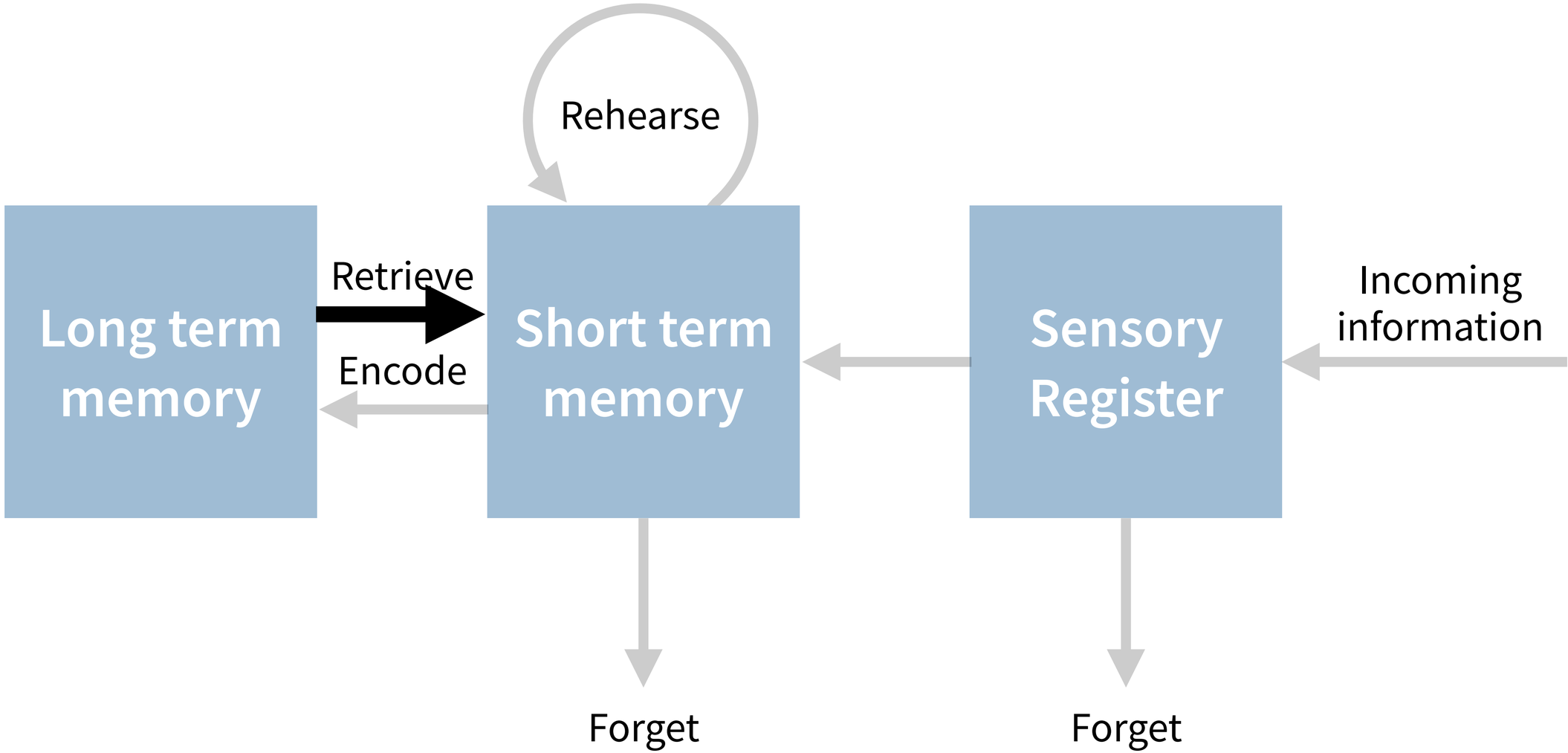
Make it Stick

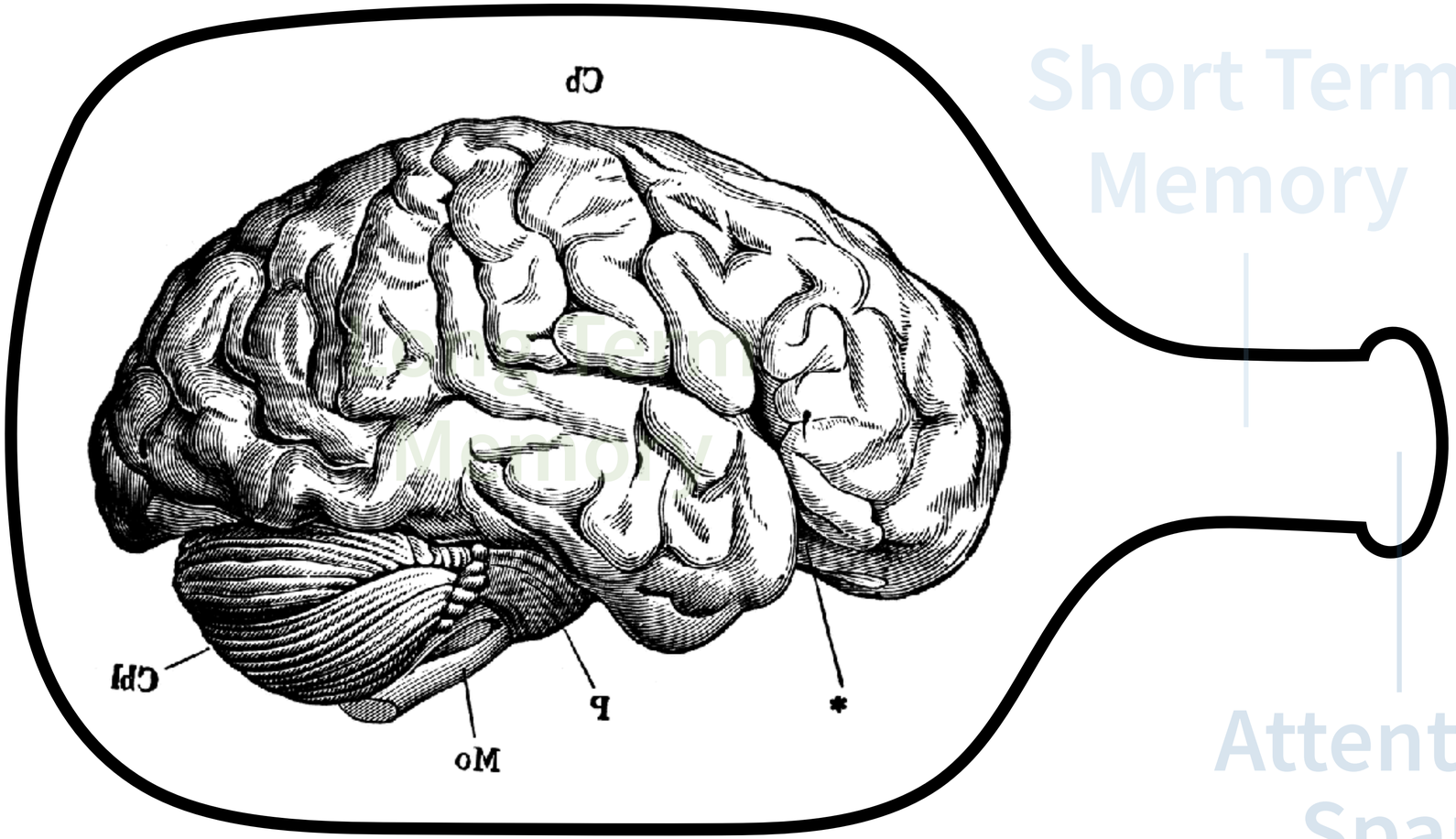
The brain revisited

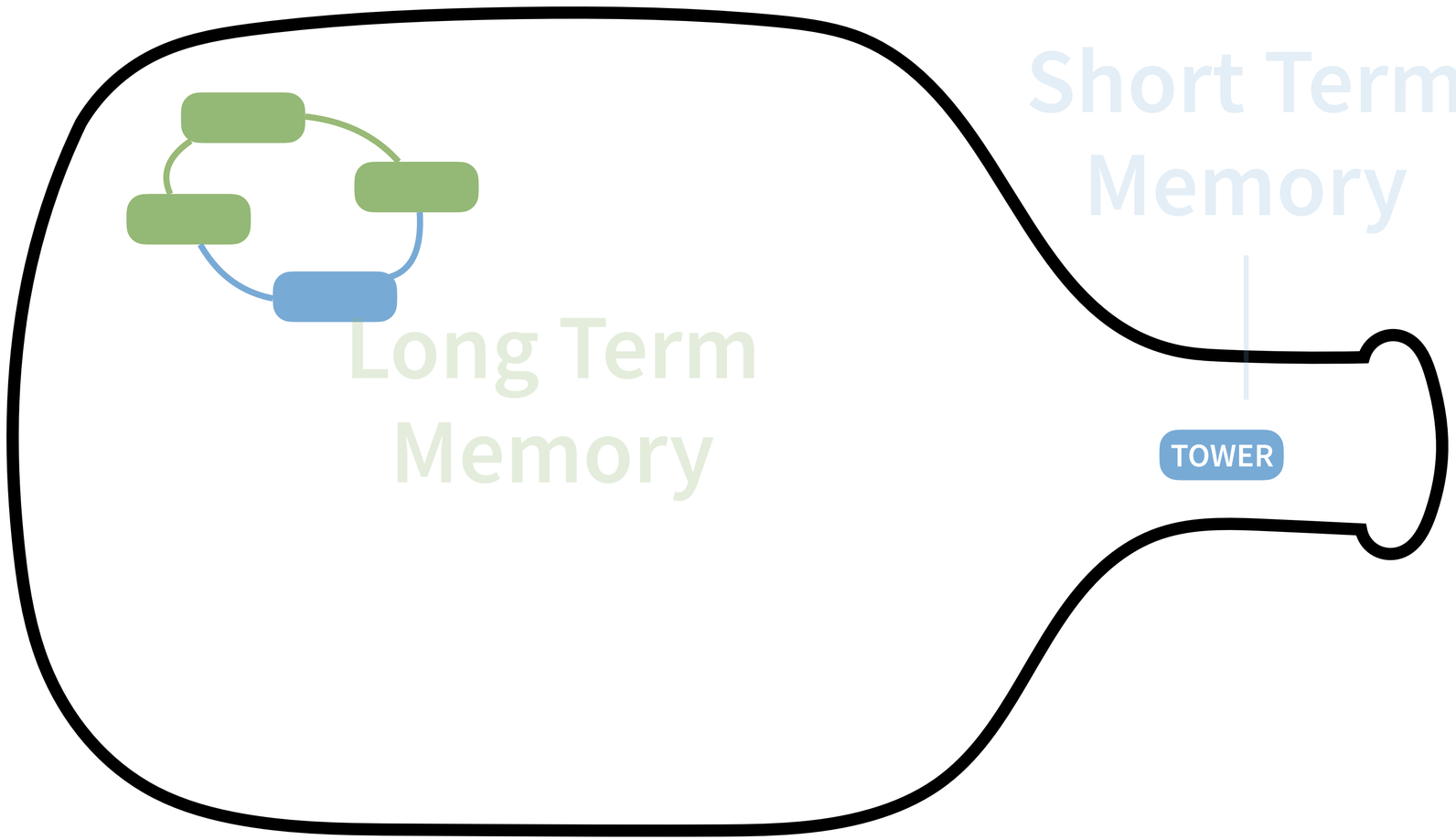


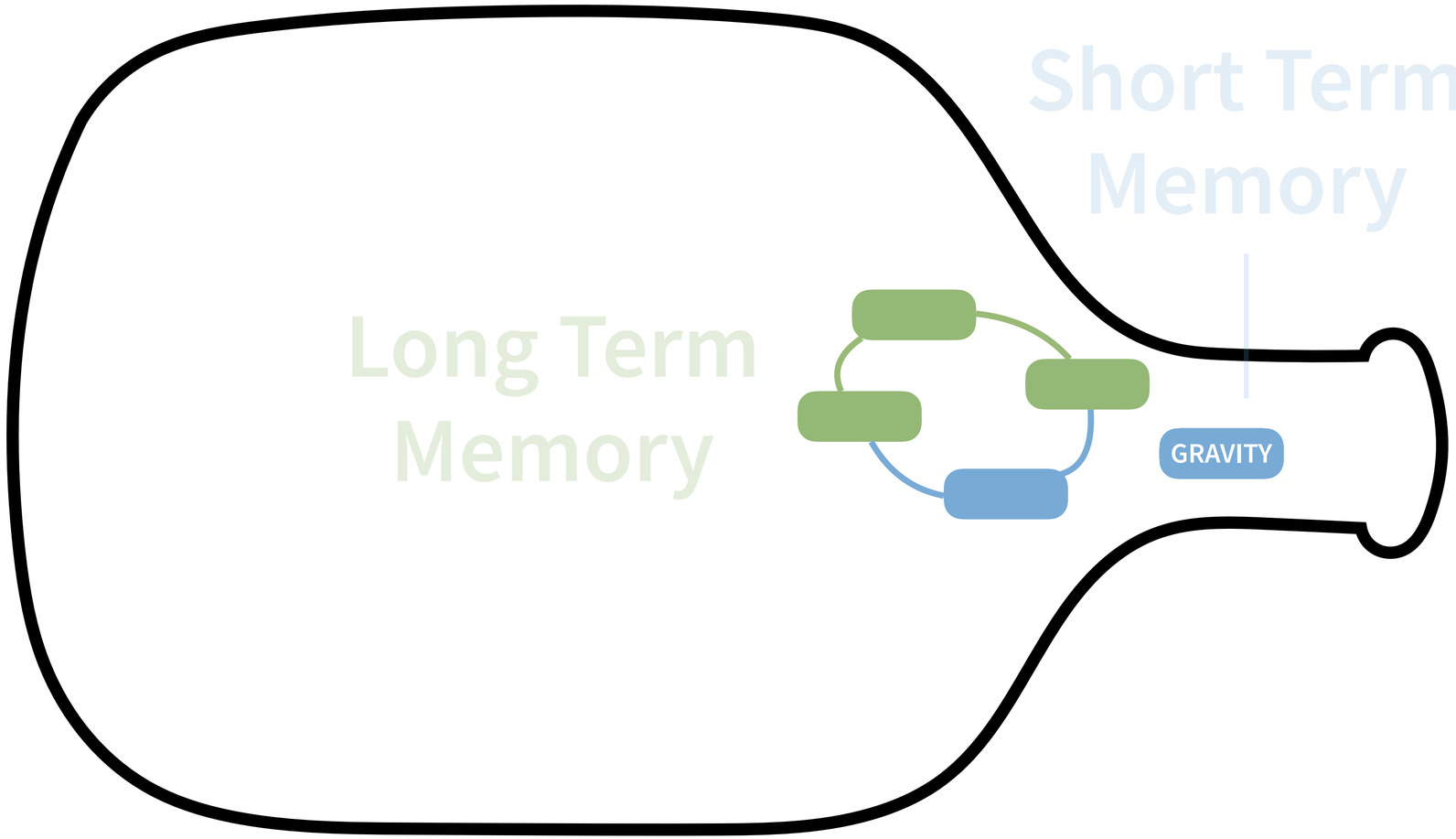


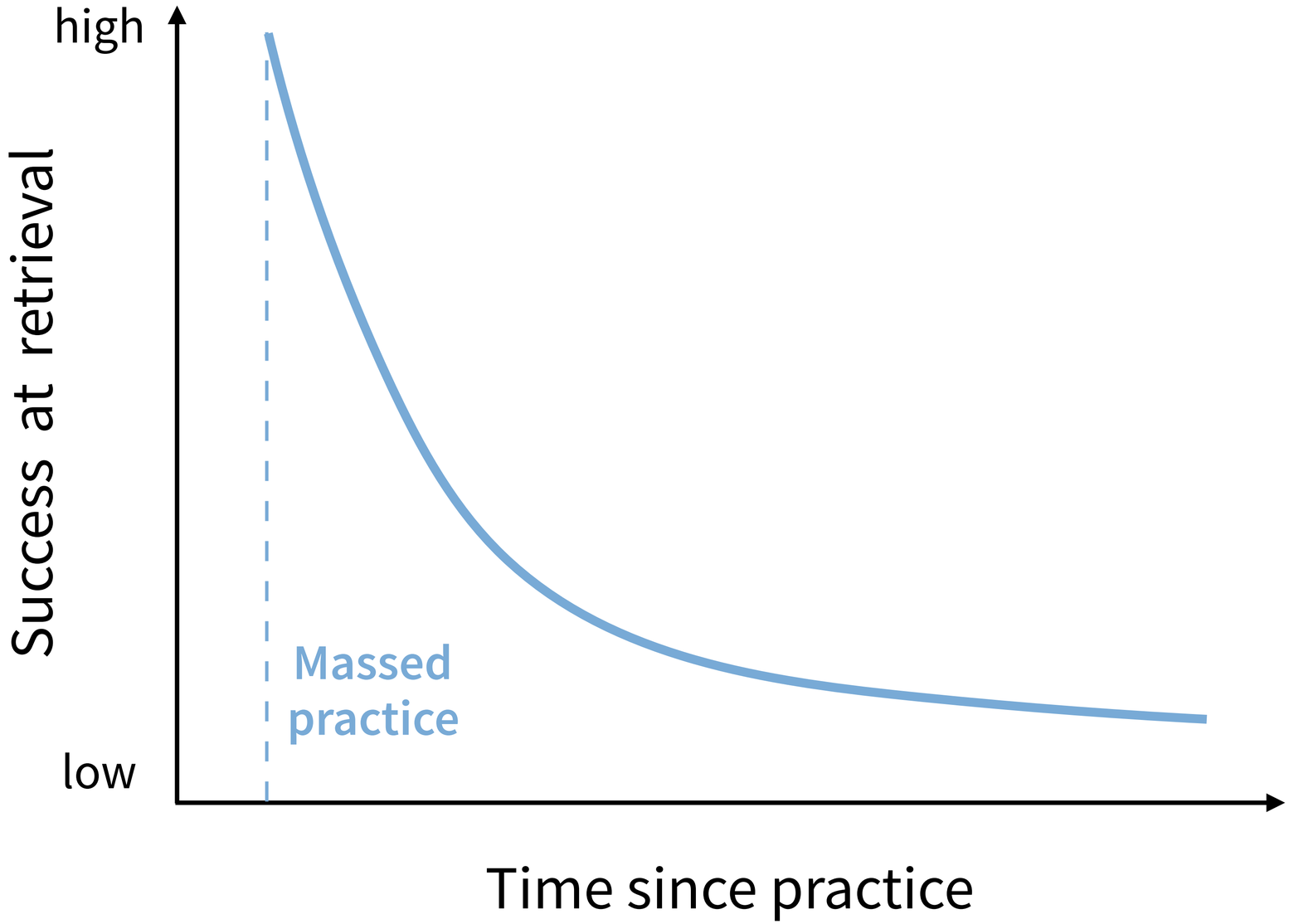


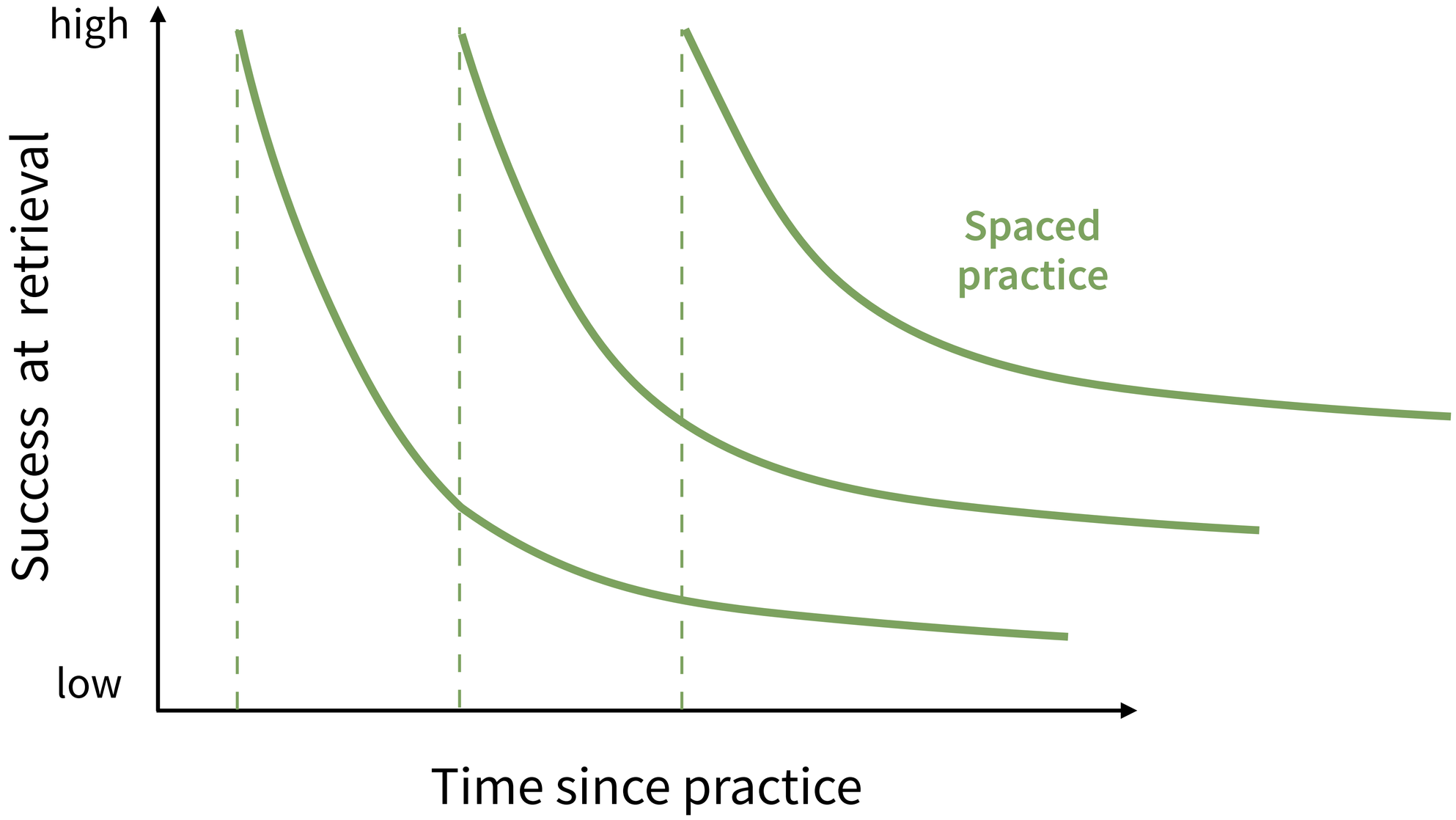


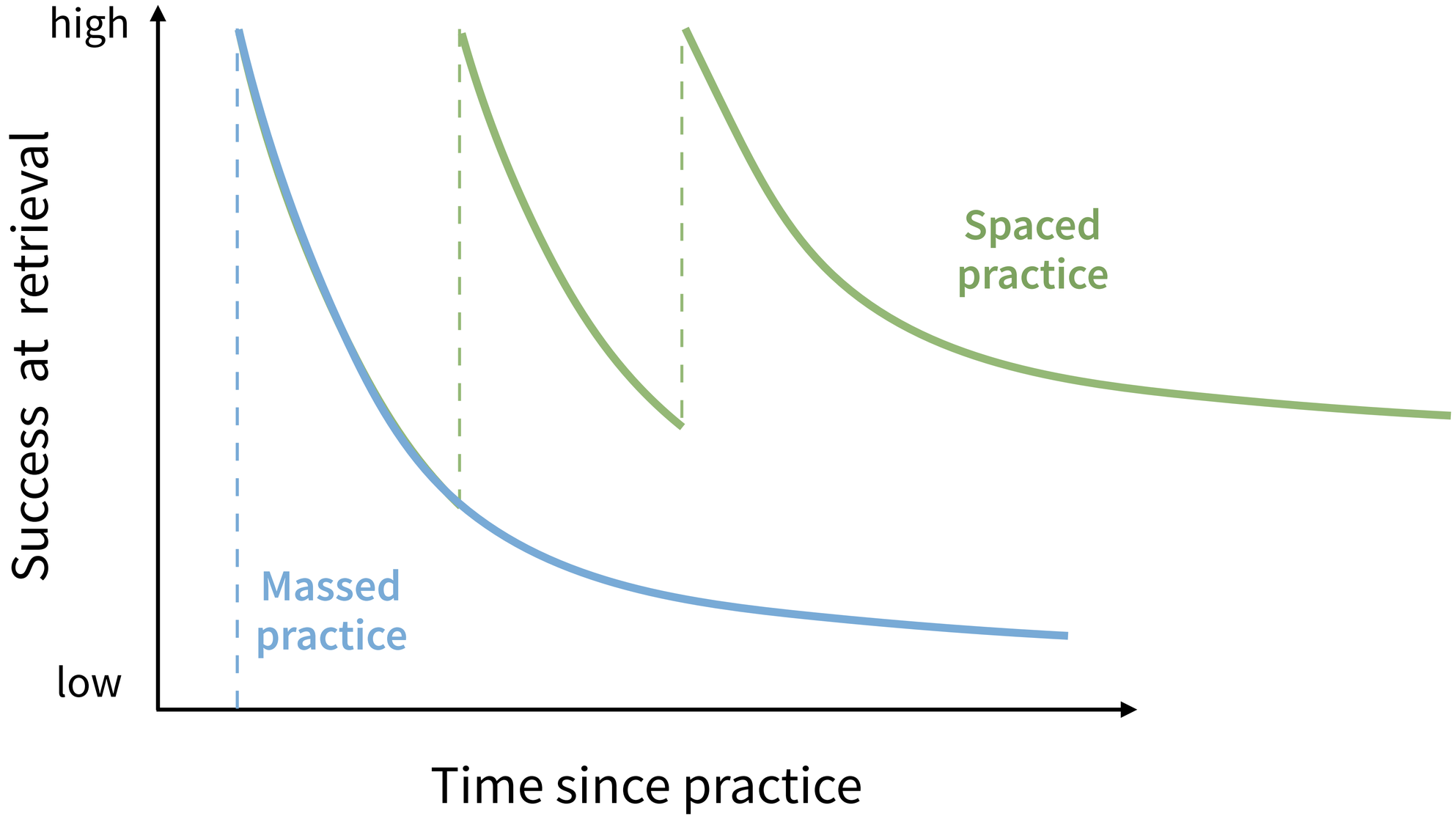












How to teach a workshop

1. Teach the **mental model(s)**

    Explain them. Practice the skills they support.

2. Put useful details in a **handout**

3. Follow up with **post-workshop quizzes**

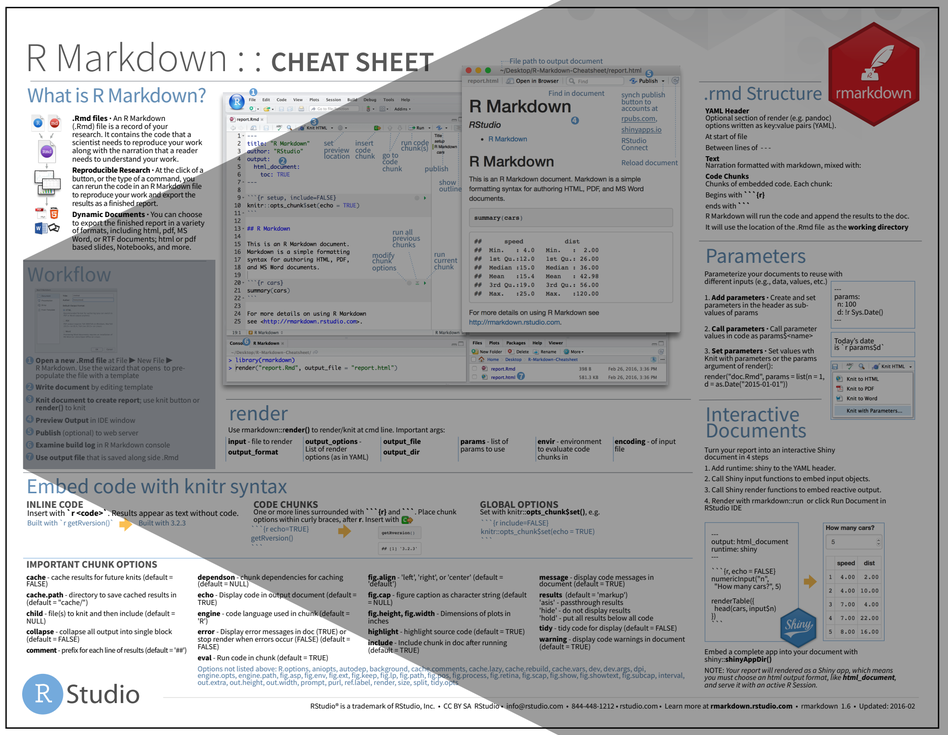
Use cognitive theory as a guide

Your Turn

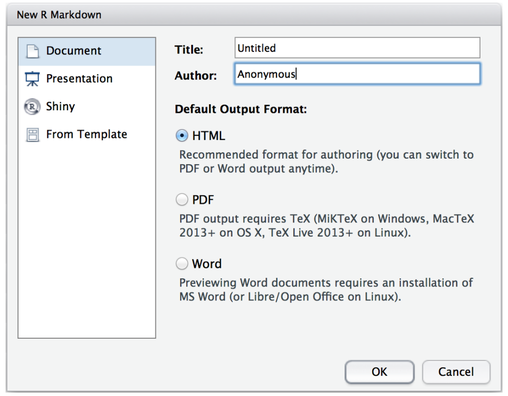
Type out three multiple choice questions about your topic and one coding exercise to test whether students have retained your learning objective(s).

R Markdown

What is R Markdown?



Workflow



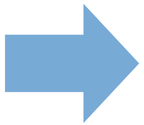
1. **Open a new .Rmd file** at File ▶ New File ▶ R Markdown. Use the wizard that opens to pre-populate the file with a template
2. **Write document** by editing template
3. **Knit document to create report**; use knit button or **render()** to knit
4. **Preview Output** in IDE window
5. **Publish** (optional) to web server
6. **Examine build log** in R Markdown console
7. **Use output file** that is saved along side .Rmd

Your Turn

Create a New Project and open an R Markdown file.
Knit the file into finished HTML output. Read it.

Headers

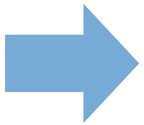
```
# Header 1
## Header 2
### Header 3
#### Header 4
##### Header 5
###### Header 6
```



Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

Text

```
Text
_italics_
__bold__
`code`
```



Text

italics

bold

`code`

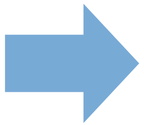
Lists

```
Bullets

* bullet 1
* bullet 2

Numbered list

1. item 1
2. item 2
```



Bullets

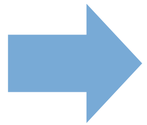
- bullet 1
- bullet 2

Numbered list

1. item 1
2. item 2

Hyperlinks

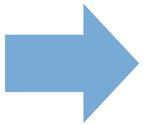
```
This is a
[link](www.git.com).
```



This is a link.

Equations

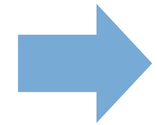
```
According to
Einstein,
$E=mc^{2}$
```



According to Einstein, $E = mc^2$

Images

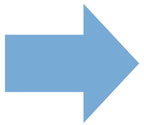
```
![](images/1.png)
The RStudio logo.
```



code chunks

Code chunks

````
Here's some code
```{r}
dim(iris)
```
````

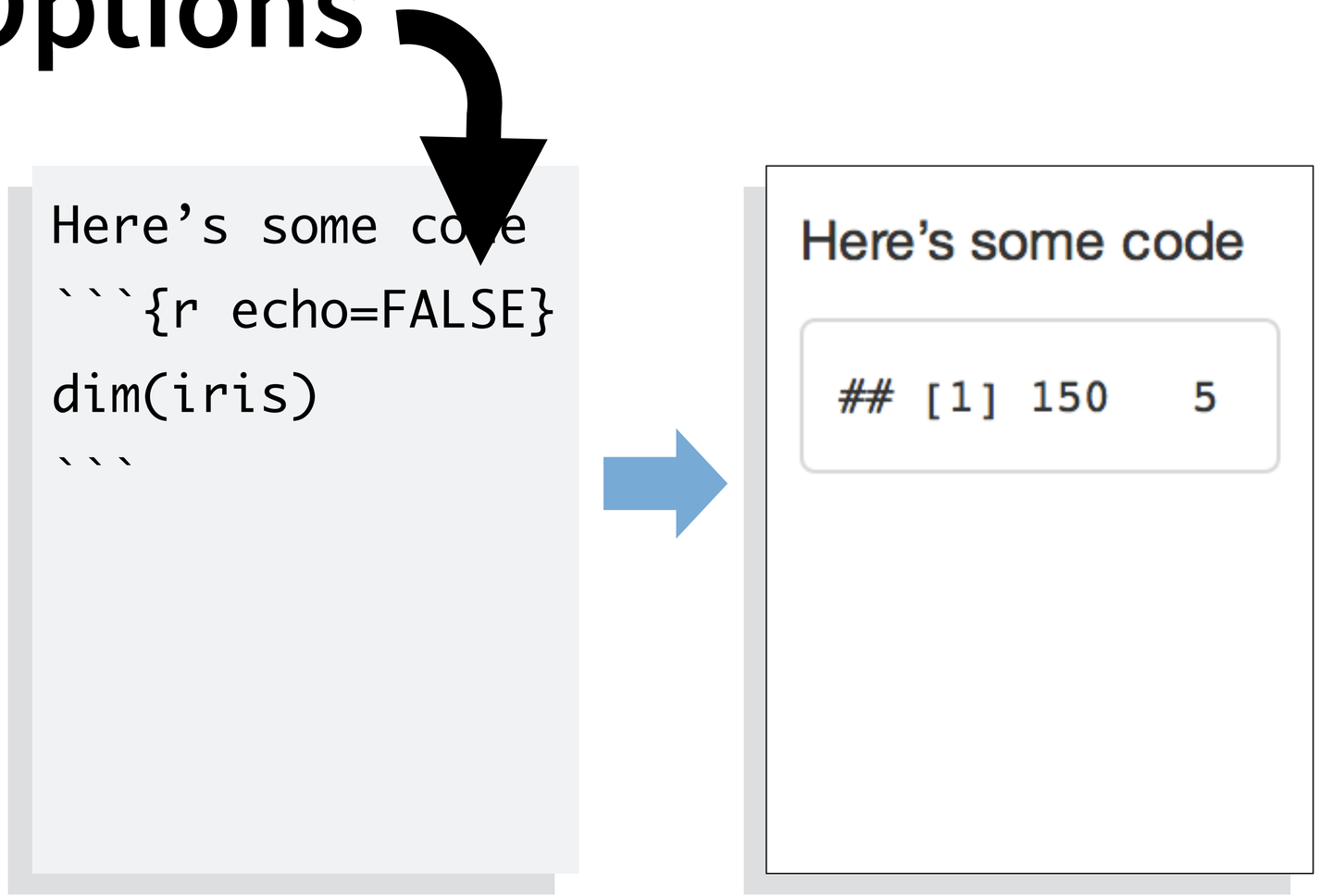


Here's some code

```
dim(iris)
```

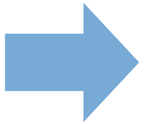
```
## [1] 150   5
```

Chunk Options



Chunk Options - echo = FALSE

````
Here's some code
```{r echo=FALSE}
dim(iris)
```
````



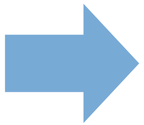
Here's some code

```
## [1] 150   5
```

Displays code results, but not code

Chunk Options - eval = FALSE

````
Here's some code
```{r eval=FALSE}
dim(iris)
```
````



Here's some code

```
dim(iris)
```

Displays code, but not results (code is not run)

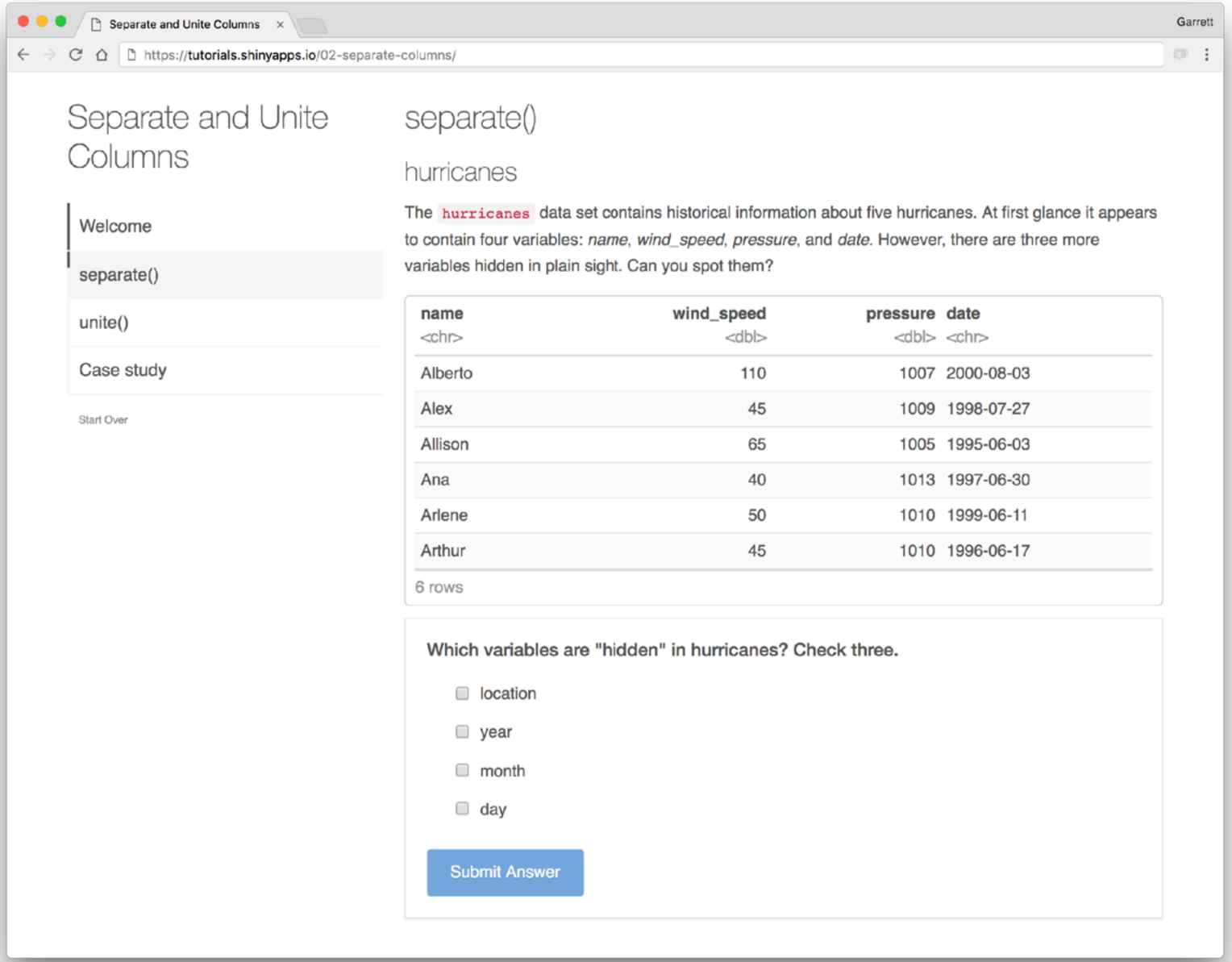
Your Turn

Use your R Markdown document to write a half page summary of your topic that includes bullet points and at least one chunk of example code.

Knit the document into an html file.

learnr

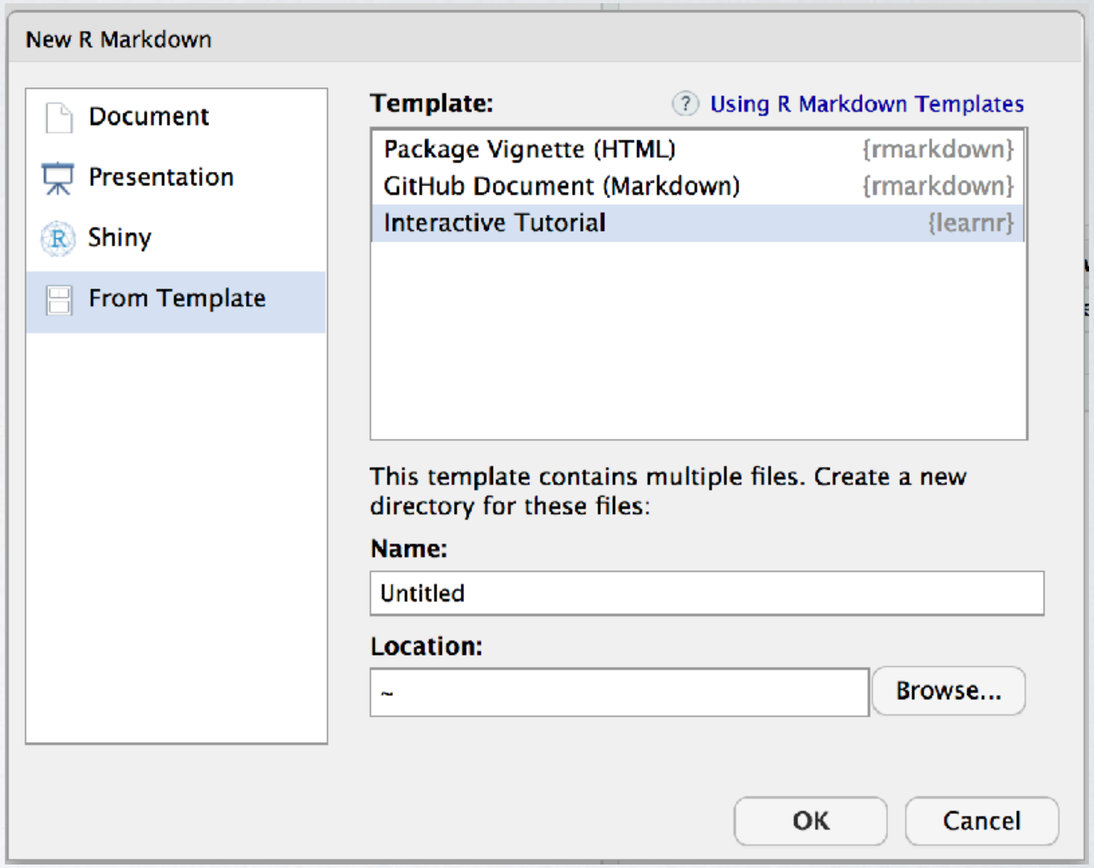
tutorials.shinyapps.io/02-separate-columns/

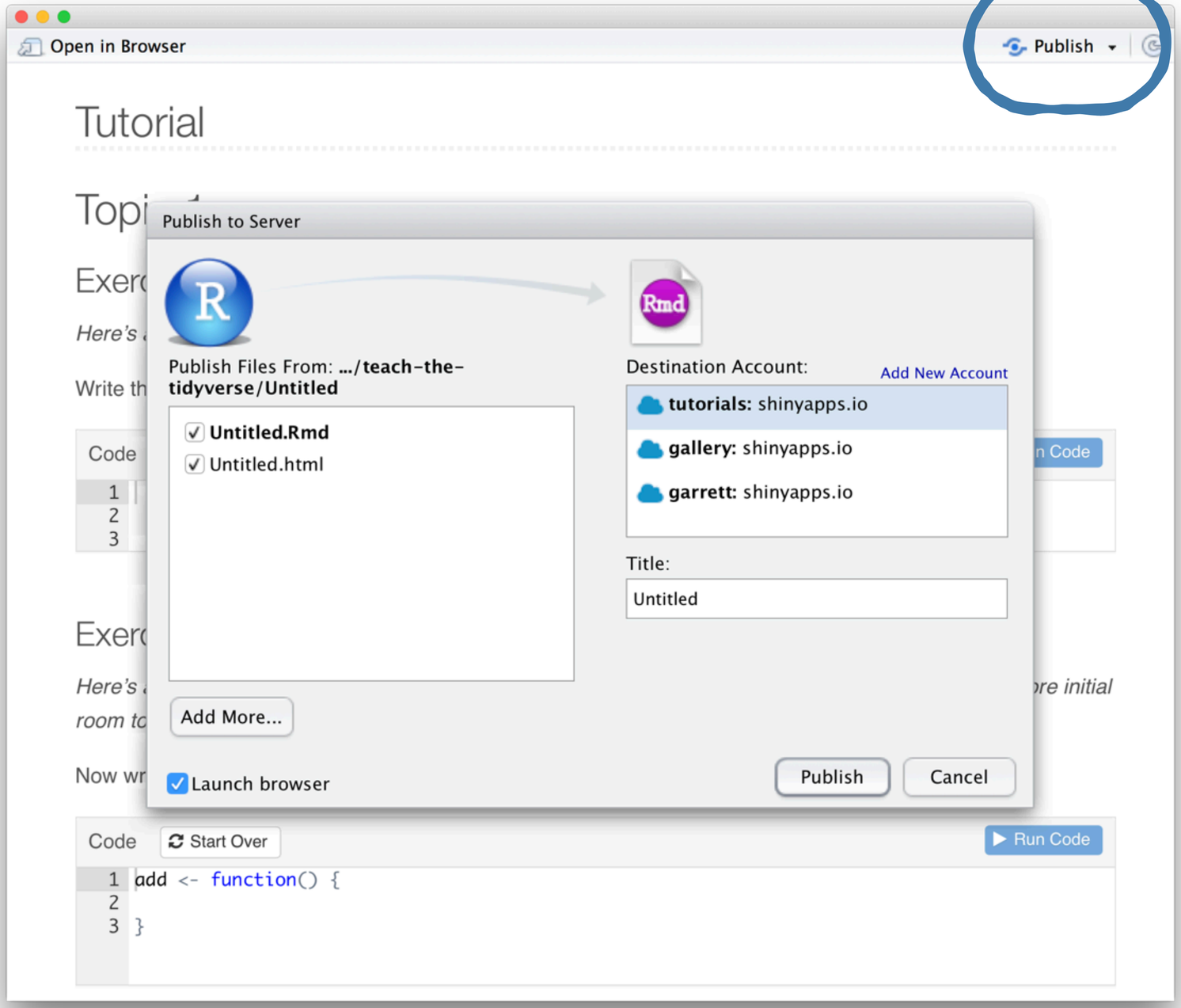


Your Turn

In your Teach the Tidyverse Project (rstudio.cloud/project/11787). Open a new R Markdown file and select From Template > Interactive Tutorial.

Read the source code in the file and run the file.





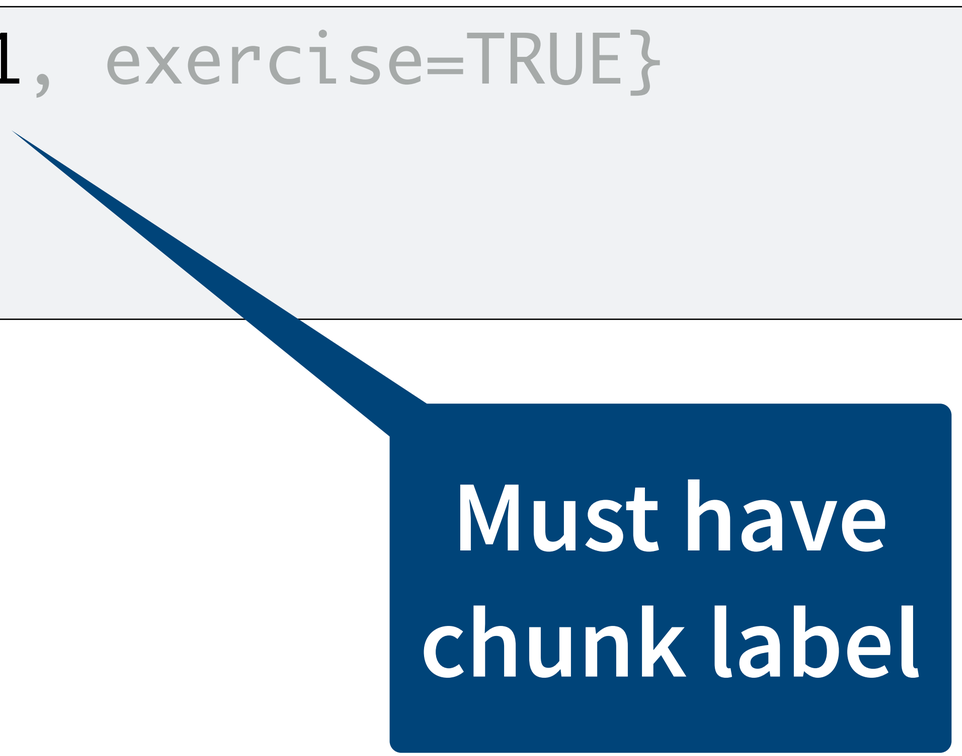
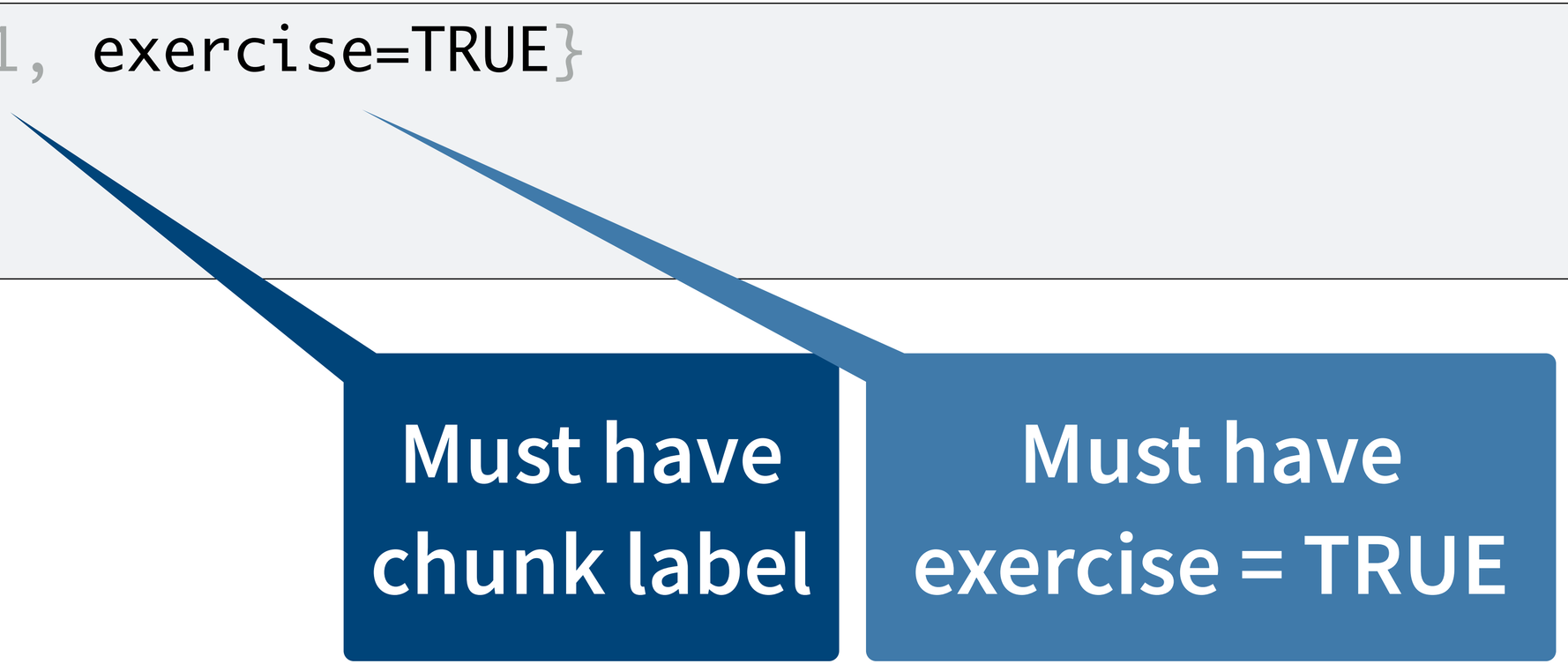
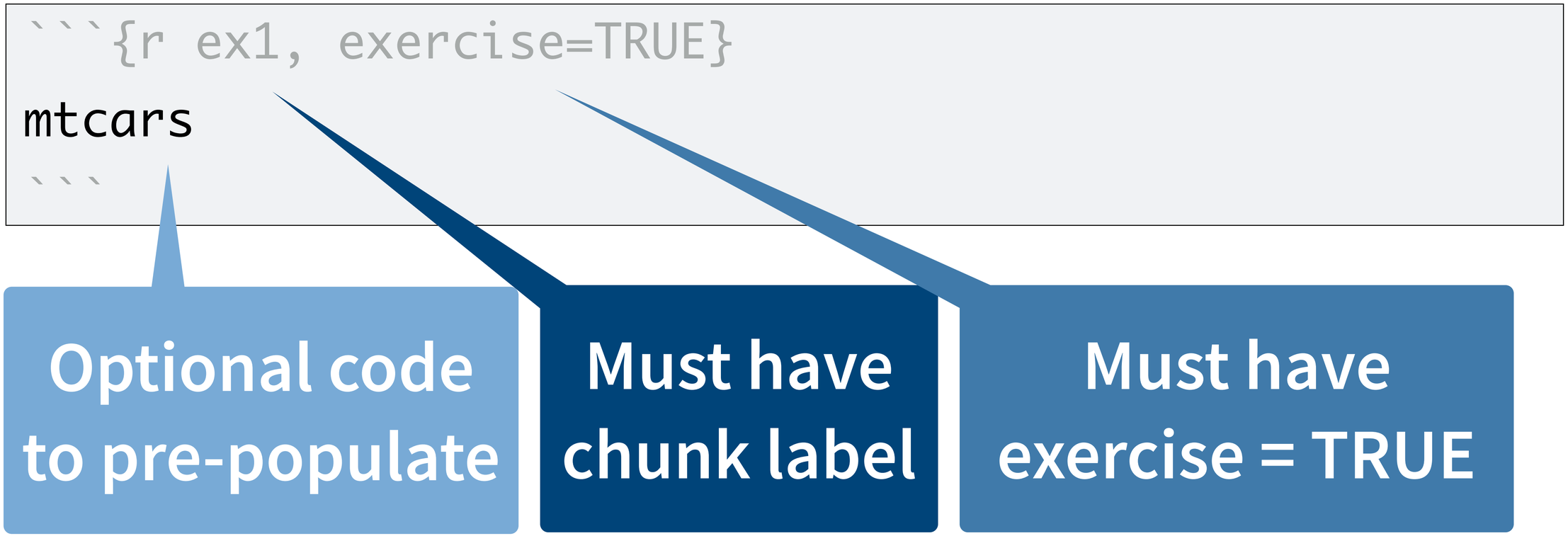
Learner setup

````
---
title: "Tutorial"
output: learnr::tutorial
runtime: shiny_prerendered
---


```{r setup, include=FALSE}
library(learnr)
```
````

Learner setup

```
---
title: "Tutorial"
output: learnr::tutorial
runtime: shiny_prerendered
---

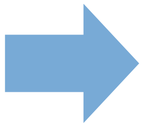

```{r setup, include=FALSE}
library(learnr)
```
```

Set output to learnr:: tutorial

Set runtime to shiny_prerendered

Load the learnr functions

Multiple choice questions

```
```{r q1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr")
)
```
```

Multiple choice questions

question()

```
    q1,
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr")
)
```

Multiple choice questions

question()

prompt

```
    q1 =
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr")
)
```

Multiple choice questions

question()

prompt

```
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr")
)
```

answers

Multiple choice questions

question()

prompt

```
    q1 =
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr")
)
```

answers

correct answer

Multiple choice questions

Use **incorrect** to return failure message

````
```{r q1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    incorrect = "Try Again"
)
```
````

Multiple choice questions

Use **message** to return messages for specific items

````
```{r q1}
question("Which package makes graphs?",
    answer("tidyr"),
    answer("dplyr"),
    answer("ggplot2", correct = TRUE),
    answer("purrr", message = "purrr is for lists")
)
```
````

Your Turn

1. Turn your R Markdown file into a learnr tutorial by placing in its yaml (then save):

   output: learnr::tutorial
   runtime: shiny_prerendered

2. In a new ``` chunk add the first of your multiple choice questions.

3. Run the document and try the question (you may need to click through to open in browser)

Quizzes

Combine several questions with **quiz()**

````
```{r quiz}
quiz(caption = "Pop Quiz",
  question("Which package makes graphs?",
    answer("dplyr"),
    answer("ggplot2", correct = TRUE)),
  question("Does forcats work with factors?",
    answer("yes", correct = TRUE), answer("no"))
)
```
````

Quizzes

Combine several questions with **quiz()**

````
```{r quiz}
quiz(caption = "Pop Quiz",
  question("Which package makes graphs?",
    answer("dplyr"),
    answer("ggplot2", correct = TRUE)),
  question("Does forcats work with factors?",
    answer("yes", correct = TRUE), answer("no"))
)
```
````

Your Turn

Add your last two multiple choice questions to the first *as a quiz*.

Run your document and take the quiz.

Coding Exercises

````
```{r ex1, exercise=TRUE}
mtcars
```
````

Coding Exercises

````
```{r ex1, exercise=TRUE}
mtcars
```
````



Coding Exercises

````
```{r ex1, exercise=TRUE}
mtcars
```
````



Coding Exercises

````
```{r ex1, exercise=TRUE}
mtcars
```
````



Coding Exercises

Add **exercise.eval=TRUE** to run initial code

````
```{r ex1, exercise=TRUE, exercise.eval=TRUE}
mtcars
```
````

Coding Exercises

Code in **<label>-setup** will be run before students code

````
```{r ex1, exercise=TRUE}
mtcars
```
````

````
```{r ex1-setup}
mtcars <- mtcars[1:10, ]
```
````

Coding Exercises

Code in **<label>-setup** will be run before students code

````
```{r ex1, exercise=TRUE}
mtcars
```
````

````
```{r ex1-setup}
mtcars <- mtcars[1:10, ]
```
````

Coding Exercises

Code in **<label>-solution** will be available as a solution

````
```{r ex1, exercise=TRUE}
mtcars
```
````

````
```{r ex1-solution}
head(mtcars)
```
````

Coding Exercises

Code in **<label>-solution** will be available as a solution

````
```{r ex1, exercise=TRUE}
mtcars
```
````

````
```{r ex1-solution}
head(mtcars)
```
````

Your Turn

Add your coding exercise to your learnr document as well as a solution. Test it out.

Headers

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



Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

RStudio Cloud | Tutorial

Secure | https://garrett.rstudio.cloud/8c34a197a84f46528d76aa11c8177deb/p/7519/Untitled.Rmd#section-topic-1

Tutorial

Topic 1

Topic 2

Start Over

Title

Topic 1

Exercise

Here's a simple exercise with an empty code chunk provided for entering the answer.

Write the R code required to add two plus two:

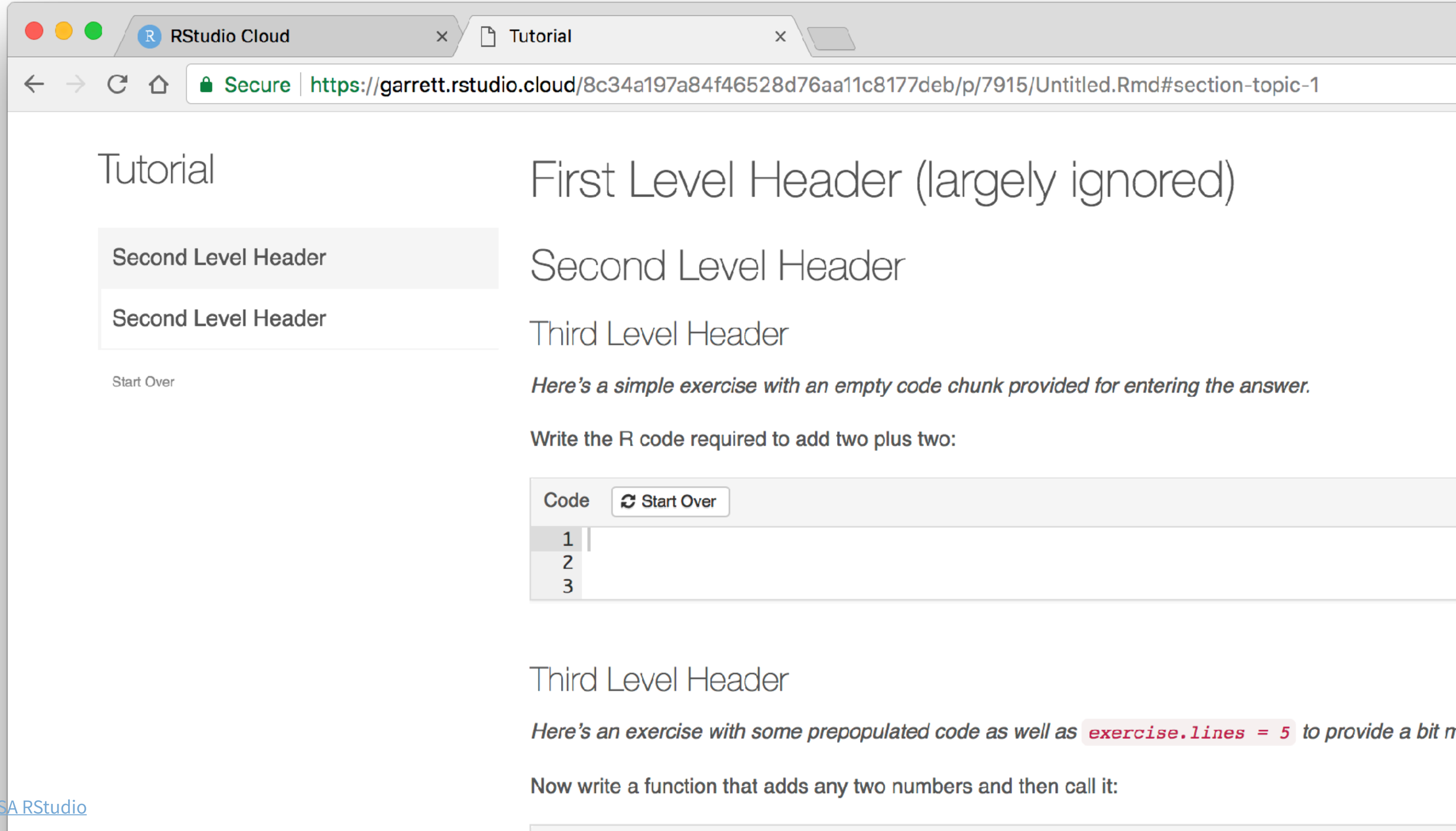
Code | Start Over

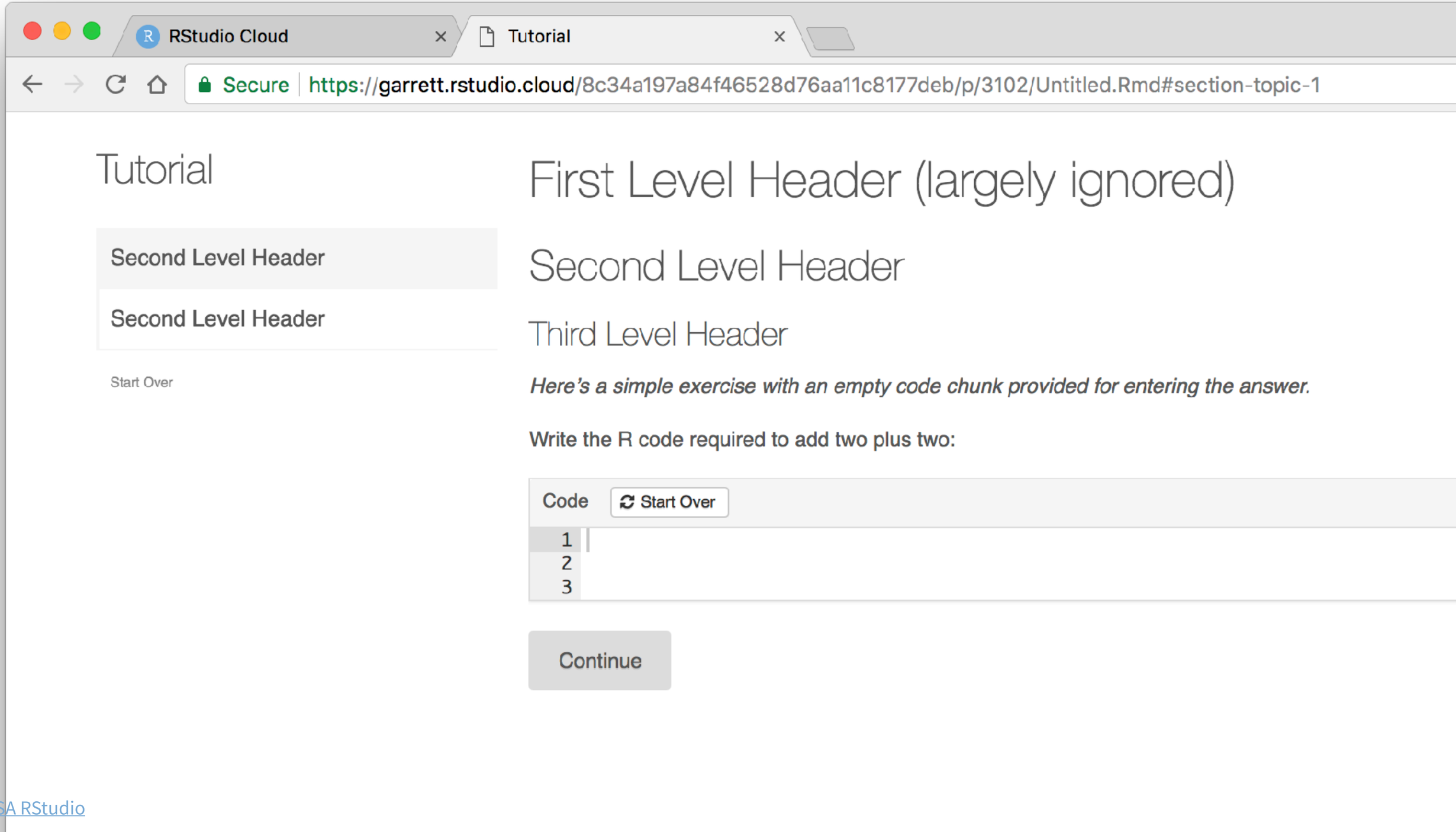
```
1
2
3
```

Exercise with Code

Here's an exercise with some prepopulated code as well as `exercise.lines = 5` *to provide a bit m*

Now write a function that adds any two numbers and then call it:





Progressive Reveal

Students will need to click "Continue" to move on to each third level header (###)

```
---
title: "Tutorial"
output:
  learnr::tutorial:
    progressive: true
runtime: shiny_prerendered
---
```

new line + indent two spaces + colon to add options to learnr::tutorial

indent four spaces

Progressive Reveal

Students will need to click "Continue" to move on to each third level header (###)

```
---
title: "Tutorial"
output:
  learnr::tutorial:
    progressive: true
runtime: shiny_prerendered
---
```

new line + indent two spaces + colon to add options to learnr::tutorial

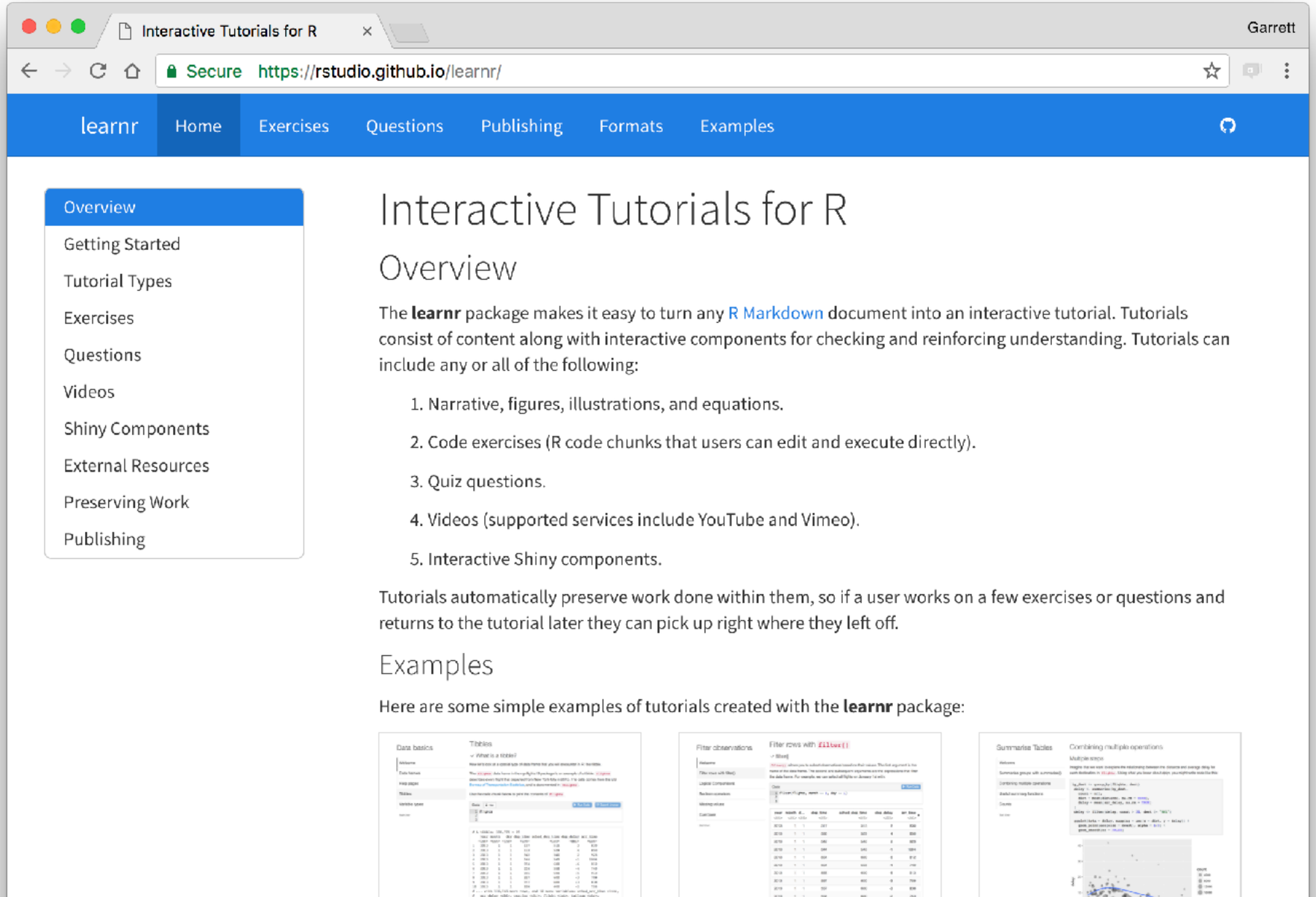
indent four spaces

Allow Skip

Add **allow_skip: true** to let students continue without finishing each exercise.

```
---
title: "Tutorial"
output:
  learnr::tutorial:
    progressive: true
    allow_skip: true
runtime: shiny_prerendered
```

rstudio.github.io/learnr/



Your Turn

Add **progressive: true** and **allow_skip:** to the yaml of the document. Then insert second and third level headers as needed to make a satisfying navigation experience.

Work through your complete tutorial.

Thank You

Thank You

Please take the class survey

www.surveymonkey.com/r/9L6PTBF