**Starting a tutorial (without coding)**

~~As discussed in chapter 3, a user must be able to start a tutorial seamlessly, and without needing to install or code.~~

~~To allow for this I installed the control package, learnr.dashboard, in the R system library ( /usr/local/lib/R/site-library ). This package contains the function to specify which R libraries to use:~~

~~`learnr.dashboard::learnr\_setup("/home/siebrenf/git/edu/learnr.proto/renv/library/R-3.6/x86\_64-pc-linux-gnu")`~~

~~This function will also look up the renv-system-library (which contains the minimal set of R libraries), and symlink those in the /tmp directory. The specified library, and this /tmp library are then set as the only available libraries. Finally, the function attempts to unload all loaded libraries (including itself) to avoid conflicts.~~

To prevent having to install anything to the system library, I’ve changed the above system.

Users start by adding the renv library to their list with:  
`.libPaths("/path/to/renv/lib")`

Next, the user can start a tutorial with one of two functions that handle everything internally:  
```  
learnr.dashboard::start\_tutorial("fg1") # available tutorials: fg1, fg2, fg3, fg4 and test  
learnr.dashboard::start\_background\_tutorial("fg2") # available tutorials: fg1, fg2, fg3, fg4 and test  
```

The latter function will start a new R session. This allows the user to keep using Rstudio! The tutorials have been fitted with code that will cause a timeout, and will shut down when the browser is closed, in order to save the server from overuse.

Note: The tutorial cannot be edited, nor easily found. Therefore, it also fills the criteria of preventing accidents/cheats!

**In summary:**

```

# in case it all goes wrong

.rs.restartR()

# minimal startup functions

.libPaths("/home/siebrenf/git/edu/learnr.proto/renv/library/R-3.6/x86\_64-pc-linux-gnu")  
learnr.dashboard::start\_background\_tutorial("fg1") # available tutorials: fg1, fg2, fg3, fg4 and test

```