Save and load user-defined functions

1 README

- `[X]` Work through this notebook at your own pace. When you're done, check a task off by typing `C-c C-c` on the line with the bullet point. Check this task off for practice!
- `[X]` You can close bullet points with the `<TAB>` key on your keyboard to reduce the clutter on the screen. Close this long paragraph now by going to the line with the bullet point `*` in it, then check the task off.
- `[X]`

  Make sure that you can run R in this buffer by executing the code block below (`C-c C-c` with the cursor ON the block) and then save the file (`C-x C-s`).

  ```
  print("hello world")
  ```

- `[X]` Sometimes, you want to check what happened in the R session buffer. Make sure that you find that buffer (e.g. `C-x C-b` for `list-buffers`).
- `[X]`

  Practice: go to the buffer `*R*` now, and run the command shown below, then come back here again.

  ```
  search()
  ```

  - `[X]` Did you know what this command would do? Do you recognize the output in the session buffer? If not, type `?search` for help, then return here again to continue.
- `[X]` In case you have multiple buffers open: remember that `C-x 1` closes all but the current buffer, and `C-x o` cycles through them. Try that now!
- `[X]` If you have any trouble with executing the `hello world` program, analyze the problem by yourself first. Typical sources of errors are:
  - Can you write to the current directory? (Windows may forbid it)
  - Is this file an Org-file? (Syntax highlighting/modeline check)
  - Did you use the correct key sequence? (Check caps lock e.g.)
  - Do you have the right code block header arguments? (You only need `:session :results output` for now.
  - Are you working with an old `*R*` session and writing to some far-flung directory? (check with `getwd()` and reset with `setwd()`).
- `[X]` Now, we're going to do learn how to save and load our own functions, and then we're going to write two functions and test it.

2 Saving and loading user-defined functions

- `[ ]`

  Functions can be saved for later use. The code in 1 defines and saves the function `hello world` as a binary to the current directory. Try it: replace the `???` in the code block by your own name.

  ```
  hello <- function(name) print(paste("hello",name ))
  hello("???")
  ```

```
save(hello, file='hello')
```

- [ ]

  Now shut down the R session (type `q()` in the session buffer and kill the buffer with `C-x k`). Try to execute the function.

  ```
  hello("error")
  ```

- [ ]

  You can load the function as shown in 1. This will require you to open a new R session.

  ```
  load('hello')
  ```

- [ ]

  Now you can execute the function.

```
hello("loaded")
```

- [ ] Do you have any questions or corrections for me about this? If there's anything you don't understand (including technical terms), please ask and don't be shy!
- If not, let's practice!

3 Practice

- The R prompt can be changed on the console with the built-in utility function `options`.
- [ ]

  Write a function that changes the prompt to `"R> "`. Execute it and change the prompt to `"R> "`.

  ```
  fprompt <- function(x) options(prompt=x)
  fprompt("R> ")
  ```

  Check your result in the session buffer.

- Change the prompt back to `"> "` by calling the function again.

  ```
  fprompt("> ")
  ```

  Check your result in the session buffer again.

- Save your function in a file as `'fprompt'`.

  ```
  load('frompt')
  ```

- As an alternative to shutting down the R session buffer, Empty the object list with `rm(list=ls())` then try to call the function.

  ```
  rm(list=ls())
  frompt("error")
  ```

- Load the function by loading `'frompt'` , and change the prompt to `"hi> "`.

  ```
  load('frompt')
  fprompt("hi> ")
  ```

  Check your result in the session buffer.

- Change the prompt back in the session buffer without using the function.

Author: Marcus Birkenkrahe

Created: 2022-03-11 Fri 13:18

Validate