

# Org-mode assignment

This file contains the first Org-mode assignment, a glossary of terms, and a sample solution in the programming language R instead of C.

The main challenge is not the programming language but managing the complexity of writing a "literate" program, which includes layout, documentation, source code and result elements.

Note that the sample solution also lists relevant references at the end.

## Assignment

Create an Org-mode file with the following characteristics:

1. A headline with the text `My first Org-mode file`.
2. A short text that explains what the following code block does (including the block header), and any other observations or thoughts you might have.
3. A C source code block. The block should contain one line of code only. It should print the sentence `My first program`.
4. For the source code block header, use the settings:
  - `:main yes`
  - `:includes: <stdio.h>`
  - `:exports both`
5. For the source code, use
  - the C function `puts()`
  - the string `"My first program"`
  - the end statement `;`
6. Save the file as `YourName.org` (e.g. `MarcusBirkenkrahe.org`)
7. Upload the file to [this GitHub repo](#).

## Glossary

| TERM                       | EXPLANATION                                      |
|----------------------------|--|
| Org-mode                   | Major Emacs feature mode for organizing things   |
| Org-mode headline          | Starts with one ore more * characters            |
| Org-mode code block        | An executable (in some language) block of text   |
| Org-mode code block header | Language and optional arguments                  |
| Argument                   | Some data passed on for further processing       |
| Function                   | Algorithm $f(x)$ expecting an argument $x$       |
| String                     | Data type representing text                      |
| Begin/End statement        | Symbol to signal the start/end of a command      |
| File                       | Storage unit in a computer                       |
| Uploading                  | Sending so that file exists locally and remotely |

## Example solution in R (not C).

### My first Org-mode file

- The function `str()` prints all variables (column vectors) of a data structure.
- In the code block, `str()` is applied to the built-in data frame `mtcars`.
- The result shows that `mtcars` contains 32 rows (observations) of 11 variables.
- The code block header exports both source code and result, starts an R session, and prints the result to `stdout`.
- The syntax is not highlighted (language keywords and data are not distinguished visually)

```
str(mtcars)
```

```
'data.frame':  32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

### References

`mtcars`: Motor Trend Car Road Tests. [URL: rdocumentation.org](https://rdocumentation.org).

R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. [URL: r-project.org](https://r-project.org).

`str`: Compactly Display the Structure of an Arbitrary R Object. [URL: rdocumentation.org](https://rdocumentation.org).

Created: 2021-12-28 Tue 10:03

[Validate](#)