

Documenting at Fontys ICT: my 4 cents

Fontys ICT Teachers

2023-09-24

Table of contents

Preface	3
1 Introduction	4
2 Premises	5
2.1 Canvas for structure vs documents for content	5
2.2 self-contained documents	5
3 How to do it...	6
3.1 How to create a git repo for a quarto book	6
3.2 How to create a new quarto book	6
3.3 How to put the files in git	6
3.4 How to build the book	6
4 Alternatives we used or considered	8
References	9

Preface

This is a Quarto book about how to write documentation with (amongst others) Quarto.

1 Introduction

This is a book about a possible way to create and structure courses and documents for/from/in canvas, meant as possible Guidelines on documentation. This is not an official document, but the discussion/proposal.

The book and its building blocks can be found in Fontys-ICT-Teachers (2023)

2 Premises

- At Fontys ICT we are mainly ICT persons, so documentation should be in version control.
- At this moment the de facto standard for version control is git.
- Canvas is our LMS, so we are looking for a maintainable way to get the lesson materials in canvas.

2.1 Canvas for structure vs documents for content

- Canvas tells us about the week structure, deadlines, where to upload what (and when).
- The course content is in documents (that are uploaded in canvas).
- From canvas module only some 3 clicks needed, preferably, to reach the info you are looking for.
- Standard document types are
 - pdf - text, theory, assignments, looks similar in different viewers.
 - pptx - for slides.
 - zip - containing programming material, or projects created by students.
 - url - to a git repo. If it contains student made work they uploaded, for the ‘toets-dossier’ this (also) has to be uploaded in a zip at the end of the semester.

2.2 self-contained documents

The content in a document should be more or less self-contained, which means:

- If the document contains an assignment or challenge, *all* info about the content of that assignment/challenge should ideally be in the document.
- To put documentation in version control, markdown is used a lot nowadays, and easy to learn. ms-Word is not very practical with version control, LaTeX is too big a hurdle for people that did not already get acquainted.
- Quarto looks promising to create documents by including content from markdown and other formats. The resulting documents can be several types: mainly pdf and/or a website seem relevant for us.

See Knuth (1984) for additional discussion of literate programming.

3 How to do it...

3.1 How to create a git repo for a quarto book

TODO

3.2 How to create a new quarto book

- Go to an empty directory. (Possibly within the git repo you created in the previous chapter)
- Open a CLI (Command Line Interface)
- In the CLI, run the command `quarto create`. From the options that pop-up, choose `Project`, and as the `Type` choose `book`.
- For `Directory` type `.` (current directory).
- Then choose if you want to open the `quarto project` you just created, or not.

The `book project` was created, now very time something changed you can create a new build, see chapter `How to build the book`, but first you may want to add your book to your git.

3.3 How to put the files in git

TODO

3.4 How to build the book

A book is build from `sources`, just like software. Quarto knows how to build from `qmd` (Quarto markdown) and a `_quarto.yml` file. You can edit the source documents, like the markdowns or the pictures. Probably you will want to add changes in git before building the book.

- Open `index.qmd`
- `ctrl+shift+P` for the vs code menu, then start typing `Quarto` (Qua should be enough to get `Quarto: Render PDF`. `Enter` starts the build.

To learn more about Quarto books visit <https://quarto.org/docs/books>.

4 Alternatives we used or considered

- From canvas: link to material, right into git?
 - github-pages work fine but, is only for public material.
 - maybe create a CI/CD script ourselves that creates html?
- alternative: create 1 big pdf (Latex?) with all material.
 - disadvantage: it's a big document, some teachers think that could be intimidating to students.
 - disadvantage: link in canvas still has to be updated manually. We have seen examples in the past with multiple links to (initially) the same document, but after some maintenance the links pointed to different pdfs.
 - disadvantage(?): no auto-update after the student downloads pdf.
 - pdf will nowadays probably be shown in the browser (unless the user configured it differently)
- alternative: use a link into a 'normal' (~non-github-pages) git repo.
 - disadvantage: you see the non-relevant git-info, like a detailed file list (and lots more) in the browser page. Sometimes you have to scroll down before being able to read the document.
 - when it is a git.fhict-repo you have to login; if it is a public repo (github.com et al) then the whole world can see it, which means we have to be sure about copyright issues.
 - pdf will nowadays probably be shown in the browser (unless the user configured it differently), but for extensions likes docx, pptx and other it really is machine dependent how it will be shown.
 - When linking to a specific version of a document this alternative could well be an option.
- another (very promising!) alternative is *Quarto*: need to look more into that!

References

- Fontys-ICT-Teachers. 2023. “”Documenting at Fontys ICT: My 4 Cents”” https://github.com/stasemsoft/Documentation_my_four_cents.
- Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. <https://doi.org/10.1093/comjnl/27.2.97>.