Oefeningen LaTeX-cursus Week 3

TEXniCie

17 oktober 2022

Deel 1: Efficiëntie met VS Code

☐ **Oefening 1** (Compilatie)**.** Kopieer de code van `https://vkuhlmann.com/latex/example` in Visual Studio Code, en compileer het bestand met Ctrl+S.

☐ **Oefening 2** (PDF-viewer)**.** Ga naar het TeX-tabje in de activity bar links in VS Code. Probeer de verschillende opties onder 'View LaTeX PDF'. Wat vind je het fijnste werken? Verander de `latex-workshop.view.pdf.viewer` optie in de settings als je een andere default wil.

☐ **Oefening 3** (Inline math shortcut)**.** Stel een shortcut in voor het invoegen van inline math. Bijvoorbeeld door het volgende toe te voegen aan je `keybindings.json`:

```
{
    "key": "ctrl+shift+m",
    "when": "editorTextFocus && editorLangId == latex",
    "command": "editor.action.insertSnippet",
    "args": {
        "snippet": "\\$ ${1:} \\$$0"
    }
},
```

Check dat dit werkt.

☐ **Oefening 4** (Errors en warnings)**.** Maak een error door een align met een witregel erin, en daarna een warning door `\label` twee keer te gebruiken met hetzelfde argument. Waar zie je de errors en warnings in Visual Studio Code?

☐ **Oefening 5** (LaTeX Workshop snippets)**.** Ga naar de volgende URL:

`https://github.com/James-Yu/LaTeX-Workshop/wiki/Snippets`.

Stel de `editor.suggest.snippetsPreventQuickSuggestions` in zoals aangegeven op de pagina. Probeer vervolgens een figure, een section en een `\textbf` te maken met de default snippets en shortcuts die erop vermeld staan.

☐ **Oefening 6** (Environment snippet)**.** Stel een snippet in voor het toevoegen van een environment. Kies als default environment naam wat je denkt het meest te zullen gebruiken (bijvoorbeeld align).

☐ **Oefening 7** (VS Code algemene shortcuts)**.** Ga naar `https://code.visualstudio.com/docs`, klik op 'Keyboard Shortcut Reference Sheet' en download de PDF voor jouw besturingssysteem. Probeer wat shortcuts uit. Welke zouden voor jou handig kunnen zijn?

☐ **Oefening 8** (Basisdocument snippet)**.** Maak een snippet die een basisdocument voor LaTeX voorziet, met alle `\usepackage`'s die je meestal nodig hebt.

☐ **Oefening 9** (Python)**.** Als je Python kent, maak een Python bestand in VS Code, en zoek hoe je het kan uitvoeren. Probeer ook de interactive console.

Deel 2: Efficiëntie in LaTeX code

☐ **Oefening 10** (Stelsel in matrix revisited)**.** Stelsels lineaire vergelijkingen kunnen opgelost worden door ze te schrijven als een matrix en Gauss eliminatie toe te passen. Repliceer dit typische stelselmatrix:

$$\left(\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array}\right)$$

De eerste rij komt overeen met de vergelijking $2x + y - z = 8$.
Getallenvoorbeeld van:
`https://en.wikipedia.org/wiki/Gaussian_elimination`

Maak een environment hiervoor. Zorg dat het environment een argument heeft voor hoeveel kolommen er voor de verticale streep staan.

Kan je dit een optioneel argument maken?

☐ **Oefening 11** (Vector)**.** Definieer een commando die drie argumenten neemt, en er een kolommatrix van maakt.

☐ **Oefening 12** (Commando \input)**.** Kopieer het `.tex`-bestand van je vorige inleveropgave, en plaats de preamble ervan in een ander bestand, dat je bijvoorbeeld `preamble.tex` noemt. Gebruik `\input{preamble.tex}` in je eigenlijke `.tex`-bestand. Kan je het nog steeds compileren?

☐ **Oefening 13** (Eigen documentclass)**.** Maak je eigen documentclass zoals aangegeven in de slides. Werkt het als je in het kopie van je vorige inleveropgave de preamble vervangt door `\documentclass{inleveropgave}`?

☐ **Oefening 14** (aux-directory)**.** In plaats van dat alle hulpbestanden zoals `.aux`, `.toc`, `.out` je mapje onoverzichtelijk maken, is het mogelijk de locatie ervoor te veranderen naar een ander mapje. Vraag Vincent als je benieuwd bent.

In VS Code, onder het TeX-tabje, gebruik 'Clean up auxiliary files' onder 'Build LaTeX project'.

Typ `"latex-workshop.latex.tools"` in je settings.json bestand, en gebruik de auto-complete. Je krijgt een hele lijst met 'tools'. Voeg deze tool toe:

```
{
    "name": "pdflatexDirs",
    "command": "pdflatex",
    "args": [
        "-synctex=1",
        "-interaction=nonstopmode",
        "-file-line-error",
        "-aux-directory=auxdir",
        "%DOC%"
    ],
    "env": {}
},
```

Typ nu `"latex-workshop.latex.recipes"` in je settings.json bestand, en gebruik weer auto-complete. Voeg bovenaan de recipes toe:

```
{
    "name": "pdflatexDirs",
    "tools": [
        "pdflatexDirs"
    ]
},
```

Sla op, en compileer je bestand. Je zou nu een nieuw mapje 'auxdir' moeten zien, en hierin staan al je auxiliary files.

☐ **Oefening 15** (Snelle compilatie)**.** Maak een `.tex`-bestand en compileer het manueel met het `pdflatex`-commando in je terminal.

Eenmaal dat is gelukt, voeg deze lijn bovenaan je `.tex`-bestand toe

```
%&document_format
\documentclass{article}

...
```

en voer dit commando uit in je terminal:

```
pdftex -ini -jobname="document" "&pdflatex" mylatexformat.ltx document.tex
```

Als het goed is zie je nu een `document.fmt`-bestand. Dit is een cache van het moment dat de preamble helemaal was ingeladen. Als je nu je `.tex`-bestand weer compileert (kan ook via Visual Studio Code), zou dit veel sneller moeten gaan.

Maar let op! De cache kijkt niet of je preamble is veranderd, dus als je je preamble verandert moet je je `document.fmt`-bestand verwijderen, en opnieuw maken. Je kan het commando voor het maken van dit `.fmt`-bestand ook instellen in VS Code. Dit gaat analoog aan Oefening 14, als

```
{
    "name": "mylatexformat",
    "command": "pdftex",
    "args": [
        "-ini",
        "-jobname=\"%DOC%_format\"",
        "&pdflatex",
        "mylatexformat.ltx",
        "%DOC%"
    ],
    "env": {}
},
```

Let erop dat in je document de eerste regel `%&document_format` is, en vervang `document_format` hier door de naam van je bestand zonder de `.tex`, en met `_format` erachter.

Dit is een moeilijke opgave, en het kan zijn dat ik iets mis in de instructies. Vraag Vincent als het niet meteen lukt.