



# Project authoring

Be the Change you Want for 42

Kai [kai@42.us.org](mailto:kai@42.us.org)

*Summary: Practice with LaTeX and YAML, speak on-camera, and complete an original project of your choice*

# Contents

<b>I</b>	<b>Brainstorm &amp; Execute</b>	<b>2</b>
<b>II</b>	<b>LaTeX it</b>	<b>3</b>
<b>III</b>	<b>Formatting Guidelines</b>	<b>4</b>
<b>IV</b>	<b>Build a Grading Scale</b>	<b>5</b>
<b>V</b>	<b>It's Your Time to Shine :)</b>	<b>7</b>
<b>VI</b>	<b>Turn-in for Grading</b>	<b>8</b>

# Chapter I

## Brainstorm & Execute

Take a look at the project tree for H2S, including which projects do not yet have an uploaded PDF. What topic would you like to add to? Would you like to create a new branch, or a bud off a current one? Consult the linked wishlist for project ideas that have already been requested.

Aim to create a project complexity that will take a student only a month of once-or-twice a week work to complete. Try to do something that is unique or has a twist that makes it different than the dozens of similar solutions which may have been posted online. You may only copy another organization's project idea if they published it with Creative Commons license, and if you do adapt someone else's idea you should link to your inspiration/tutorial/source in the PDF.

Sketch out your project goals before you begin, as a series of bullet points. Goal setting will help you stay on track.

Next, take about a week of work to implement your own solution to the coding challenge. Note which requirements were easy and which ones might be stretch goals (bonuses).

# Chapter II

## LaTeX it

Download the latex-template.zip file and unzip it. This contains all the tools you need to build both a PDF and a grading scale.

Walkthrough:

- Within the main folder, we have `folder_template` and `resources`.
- Inside `resources` you'll find the source code for our signature LaTeX template and the program `scaleValidator`, which checks if your scale is ready for upload.
- Most of what you need is inside `folder_template/subject` and `folder_template/scale`.

Take a look at the `YOUR_PROJECT_NAME.en.tex` document. This is an excellent cheatsheet for most of the commands that you need in order to put a 42 PDF together.

Go ahead, try it out: Edit the Makefile inside of `folder_template/subject` so that the `TARGETS = YOUR_PROJECT_NAME.en.pdf`. Do not change anything else in the Makefile. Save it, navigate to this folder in the terminal, and "make".

View the .pdf output to understand what each of the example commands does.

Other commands that are common to LaTeX can be researched online and used alongside our template. For example, adding an image is not detailed there, but the syntax works:

```
\includegraphics[width=0.6\textwidth]{filename.png}
```

# Chapter III

## Formatting Guidelines

Project PDFs can be written however you feel is appropriate. Make it look nice and as crisp/professional as you are able. It's good to include these elements:

- A Title page with authors (you!) and a brief summary of purpose.
- A funny Foreward that may or may not be related to the topic.
- However many Introduction/Goals/General\_Instruction pages that you need to introduce the topic.
- Some technical tips on how to begin (it's okay to be easier than 42, they are kids).
- A Mandatory Part that describes what they will be graded on.
- A Bonus Part that sets stretch goals for them to go further if they find it easy.
- A Turn-in or Formatting section that gives them specific instructions on how to structure their Vogsphere repo. It is nice if we give them very exact folder structures to follow so that we can extract their files via automated scripts later, or run Moulinette on the submission.
- Inspiring, colorful images of what they can expect to produce (these should be screenshots from your own demo of the project!)



Draft your PDF and then run "make" to see if it works. You may have to debug some error messages, including gems that need to be installed at the beginning. Watch out for special characters in LaTeX there are many of them which need to be escaped with a backslash, including underscores and most math symbols.

# Chapter IV

## Build a Grading Scale

Here are the skills to pick from as you are writing a grading scale. For each project you should choose 1-4 most relevant skills.

- Basics
- Web
- Imperative programming (Use for command-line games etc)
- Graphics
- Parallel computing (Use for AI projects if applicable)
- Security
- DB & Data
- Objectoriented programming
- Group & interpersonal
- Technology integration (Use this one for mobile device projects)
- Algorithms & AI (Use for algorithms that solve a puzzle of any kind)

Look at the template in `/folder_template/scale/YOUR_PROJECT_NAME.en.yml`.

The comments in this file are pretty detailed and will explain what you need to do if you read them carefully. Fill out the first parts, and then copy and paste the modular sections to build up as many grading questions as need to be assessed. Here are some notes:

- Set the number of corrections to 2.
- Notice that the numbering for the "position" variable resets at the beginning of each section, and always starts from 1.

- Questions can be a boolean (yes or no), or multi (a slider from one to five), or text (just a text box for commentary).
- Each skill needs to have percentage points adding up to 100 for "standard" questions, and can have up to 25 points under "bonus" questions. The percentage points should be given as integers.

When you have drafted your scale, navigate to the `/resources/scaleValidator` folder and run `"ruby scale_validator.rb ../../folder_template/scale/whatevermyfilenameis.yml"`. Read the output of the validation program and fix your file accordingly.

# Chapter V

## It's Your Time to Shine :)

Ready for 15 minutes of fame?

Write a script for a brief video or series of videos that you would like to post in the intra video library to help students with this project.

They can be short, just 3-5 minutes, or long if you really want to. If you record 20 minutes of content we will consider if it should be broken up into some smaller attention spans.

These will be modeled after the Intra videos that we relied on for reference during the piscine. Teach the essential concepts and commands that you learned about while making this project. Try to make your script brief, cute and interesting as well as useful.

Get in touch with Kai to schedule a recording time. We often record with the AV team on Fridays. You will sit at a desktop computer and be recorded by the iSight camera while your on-screen activity is also captured - we stitch them together later in post processing.



# Chapter VI

## Turn-in for Grading

Include the following files in your Vogsphere repo:

- Your code from solving your own programming challenge
- The .tex and .pdf document from your Subjects folder - and any images that you used to generate them
- The valid .yaml scale
- The script for your video (in case we want to dub it in translation later)
- If helpful, any extra README that I should include on the project description in Intra.