



## Project Authoring

Be the Change you Want for 42

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

*Summary: Break down a concept into individual parts.*



*This work is licensed under a [Creative Commons Attribution-Noncommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).*

# Contents

<b>I</b>	<b>Designing a H2S Project</b>	<b>2</b>
<b>II</b>	<b>Outline</b>	<b>4</b>
<b>III</b>	<b>Build a PDF</b>	<b>5</b>
III.1	Avoiding Compiler Errors . . . . .	6
<b>IV</b>	<b>Build a Grading Scale</b>	<b>7</b>
<b>V</b>	<b>Turn-in for Grading</b>	<b>9</b>

# Chapter I

## Designing a H2S Project

So you want to write a 42 project! Congratulations, this marks a humanitarian and intellectual challenge that you are willing to take on. We admire your effort already!

**Topic:** Pick a topic that is a natural progression in terms of difficulty from what HackHighSchool students are studying already. Consult the [curriculum concepts map](#) for an overview of what is currently taught at HackHighSchool. Decide which HackHighSchool projects will be prerequisites for yours.

**Scope:** You should aim to create a project that will take a high school student 1 to 4 HackHighSchool classes to complete, that is, 4 sessions of 4 hours effort. If you have more content than that, you can create a sequel that builds on the concepts from your first one, and earn the T2 authorship credit instead of T1.

Here is my recommended template for a H2S project. This differs from a 42 project in that we offer more straightforward explanation of new concepts, rather than hints and mystery!

- Title page
- Humorous Introduction (Preface)
- Serious Introduction (Introduction)
- Key Concept 1: Overview, External links, Example, & Challenge
- Key Concept 2: Overview, External links, Example, & Challenge
- Key Concept 3 ... up to Key Concept 6: Overview, External links, Examples, & Challenges
- Overview of main/larger project which will make use of all the above concepts
- Mandatory Part of main project
- Bonus Part of main project

- Inspiring screenshots of what this project will look like when it's done

You will also need to turn in the following:

- A document describing which HackHighSchool projects should be prerequisites for this one.
- A grading scale.
- Any supporting files or skeleton code that you wish to provide the students.

The chapters that follow have instructions for how to use the 42 PDF maker and grading scale maker. We do not use an autograder for HackHighSchool assignments so far, although you can write a script that acts as a filechecker.

# Chapter II

## Outline

You may wish to start by composing your PDF in Google Docs or some other word editor, without the 42 formatting. We don't want the formatting to be a barrier to you writing content. If you feel mystified by the LaTeX and YAML instructions below, you can turn in an unformatted project and unformatted grading scale for 75% credit.

# Chapter III

## Build a PDF

To create 42-branded PDFs, we use a custom LaTeX library written by David Giron (Thor). You need admin privileges in order to install some of the dependencies, so I have set up the LaTeX repository on the three computers at the HackHighSchool check-in table and recommend that you use these to compile your PDF.

Post in the `#h2s_projectauthorship` Slack channel about your intended PDF topic to receive the login information for these check-in computers.

In each of the HackHighSchool computers, there is a folder called `42PDFMAKER` in the home directory. This contains all the tools you need to build both a PDF and a grading scale.

### PDF Walkthrough:

- Within the main folder, PDF templates are stored in `folder_template/subject`.
- There is a Makefile and various folders containing `.tex` files, `.pdf` files and other intermediate files that are generated by LaTeX. The `.tex` files are the source code and the `.pdfs` are the compiled final product.
- Open the Makefile and change `TARGETS` to name a PDF that you wish to produce. Use the name of `.tex` template that exists in this folder. For example, if you set `TARGETS = examples/YOUR_PROJECT_NAME.en.pdf`, the Makefile will look inside the `examples` folder for `YOUR_PROJECT_NAME.en.tex`, and compile it.
- After you have edited the Makefile, use a terminal to navigate to the subject folder and type `"make"`. `"make clean"` will also clean up your folder by deleting all the unnecessary `.aux`, `.log`, `.out`, `.toc` and `.pyg` files. The compiled PDF will be generated in the folder containing the Makefile, not the folder where the `.tex` was stored.
- Try this with `YOUR_PROJECT_NAME.en.pdf` for a tutorial on what formatting options are available!

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}
```

## III.1 Avoiding Compiler Errors

LaTeX is sensitive to many special characters which must be escaped with a backslash, including underscore(`\`), tilde(`~`), minus sign(`-`), percent(`%`), dollar sign(`$`), ampersand(`&`), and presumably many more that I am forgetting.

If you get Build FAILED, scroll up a little bit in terminal and pay attention to which line number caused the fatal error. If the last line says Build OK, then you are good!

# 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 out of the list of skills that are tracked by the H2S cursus.

- 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 1.
- 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

## Turn-in for Grading

Include the following files in your Vogsphere repo:

- A text file listing the prerequisites for your project
- The .tex and .pdf document from your Subjects folder - and any images that you used to generate them.
- The valid .yaml scale.