

## Assignment #6—It's Like Magick

---

**Due: March 01, 2023 at 11:59pm**

This week's assignment will be shorter than usual, since we'll also be publishing the final project guidelines this week and we want to give you some extra time to start thinking about that.

This assignment consists of one component:

1. Practice using Pandoc and ImageMagick to convert documents and images

We expect this assignment to take 0-1 hours depending on your proficiency level with the tools. If you find yourself unproductively stuck or unproductively struggling, ask on Ed and/or go to office hours!

Remember to install ImageMagick and Pandoc using the instructions from the [software page](#).

### **Part I: Practice with Pandoc and ImageMagick (3 points)**

In this part of the assignment, you will be building a *static site generator*, a piece of software that takes content written in a simple format and converts it into a website. Don't worry, you don't have to write any code! We'll be taking advantage of the powers of Make, Pandoc, and ImageMagick to do all the hard work for us.<sup>1</sup>

The CS 45 team has started to put together a little blog describing the different tools we use. You can [clone our repository](#) and see what we've come up with so far.

To save time, we've been writing our content in Markdown files. Markdown is a simple markup language—a way to express rich text, like bolding and italicizing, within an ordinary text file. It's really popular—you might have used it before on websites like GitHub, Ed, or Reddit. Even Google Docs has some support for it nowadays! However, web browsers only understand HTML, a more complicated markup language, so we need to convert our Markdown files into HTML files somehow.<sup>2</sup>

Our posts also reference strips from the webcomic [xkcd](#), which are strangely almost always relevant to whatever we're talking about. However, the images we downloaded from the xkcd website were PNG files. To save space and bandwidth, we want to convert them to compressed JPG files.

To start, take a look at the Makefile we set up. There's a lot of comments explaining everything that's going on, but in summary: we need you to figure out how to convert our Markdown files into

---

<sup>1</sup> This is actually a very realistic use case—we use a static site generator to generate [cs45.stanford.edu](https://cs45.stanford.edu).

<sup>2</sup> You don't need to write any Markdown for this assignment, but if you want to learn, here's a tutorial: <https://www.markdowntutorial.com/>.

HTML and compress our images! There are two places in the Makefile you need to edit; if you search the file for "TODO" you'll find them:

- The first is a Markdown-to-HTML rule, which converts one or more Markdown files into an HTML file. We'd recommend using Pandoc for this.
- The third is a rule to convert PNG images to JPG images. In addition to simply converting the formats, you'll also want to *compress* the JPG images by lowering their quality to about 50%. We'd recommend using ImageMagick for this.
  - Note: the autograder is running Ubuntu, which comes with ImageMagick 6 (not ImageMagick 7, which is the latest version). This isn't a big deal, but the `magick` wrapper command won't work; instead, run the `convert` command directly.

The Makefile has a few built-in rules to help you as you work:

- `all` builds everything which needs to be updated (but this won't work correctly when you're editing the Makefile itself; see below for more info)
- `clean` deletes all the build "artifacts", i.e., the output, which forces everything to be rebuilt the next time you run `make`
- `browse` tries to open the blog in your default browser; this may not work on all computers.
- `serve` tries to run a web server on your local computer, and gives you a URL which you can open in your browser

If neither `browse` nor `serve` work, you can also click on `index.html` in a file browser to open it. We've posted [an example of what the blog should look like](#), so you can make sure your version looks like ours.

One thing to be careful about—Make isn't smart enough to realize that it might need to rebuild things when the *Makefile itself* changes, so it might falsely think all the files are "up-to-date" if you edit the Makefile. Every time you edit the Makefile, rerun `make clean && make` to rebuild all the files with the updated Makefile.

For this part, submit your Makefile.

### **Feedback Survey (0.5 points)**

Once you have completed the assignment, you can earn an additional 0.5 points by completing our anonymous feedback survey. Given this is the first offering of the course, we want to collect as much feedback as possible to improve the course in the future. You can complete the survey [here](#). Once you complete the survey, you will receive a completion code which you should place in a text file named `survey.txt`. The survey is anonymous so submitting the completion code is the only way to verify that you completed the survey. *Please do not share this code with anyone, as that would constitute a breach of the honor code.*

**Submitting Your Assignment**

Once you have finished this assignment, you will need to upload your files to [Gradescope](#). Make sure to upload your files to the Assignment 6 submission page. You should also upload `survey.txt` if you completed the survey.

Unset

```
zip -jv assign6_submission.zip ./Makefile ./survey.txt
```

*All files must have the same name as specified above.*