

## Contents

<b>1 LAHP2019</b>	<b>1</b>
1.1 Outline for the day . . . . .	1
1.1.1 Bits and Bytes . . . . .	1
1.1.2 Command Line . . . . .	1
1.1.3 Grep . . . . .	2
1.1.4 Regular Expressions (RegEx) . . . . .	2
1.1.5 Pandoc . . . . .	2

## 1 LAHP2019

“Working with Texts,” part II: 30 May 2019.

### 1.1 Outline for the day

	Topic	Time
	Introduction to bits, bytes, encoding etc	30 mins, Marty
	Intro the CLI: principles, navigations	15 mins, Christopher
	Grep	15 mins, Jonathan
	Grep questions	15 mins
	Regex, using grep	20 mins, Christopher
	Regex questions	15 mins
	CLI tool for document conversion: Pandoc	10 mins, Christopher

#### 1.1.1 Bits and Bytes

[See Marty’s slides]

#### 1.1.2 Command Line

Make sure you have downloaded Git (<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>).

1. What are command line tools, why do we use them, and which do we use most?
  - **command line tools** are . . . They are based on “standard” Unix commands, but also include other commands from downloadable packages (e.g., Git and Pandoc).
  - **we use CLI tools** to hack our system, to have full control . . .

- the CLI tools we use most are navigation, conversion, and system debugging...

- **\*\*navigation:**

```
cd pwd mkdir echo
```

2. Exercise: Get a copy of this repository.

- start with `cd + ENTER`.
- make a new directory called “gitspace”
- navigate to the gitspace directory
- use the `git clone` function.

```
git clone https://github.com/cmohge1/LAHP2019.git
```

For a very good (and broad) overview of command line tools, see <https://github.com/jlevy/the-art-of-command-line>.

### 1.1.3 Grep

### 1.1.4 Regular Expressions (RegEx)

### 1.1.5 Pandoc

Install Pandoc [here](#).

Pandoc is a very handy universal document converter. It is used on the command line following a basic syntax:

```
pandoc FILE-TO-CONVERT -o CONVERTED-FILE
```

We invoke pandoc first, then type out (and tab) the file we want to convert, use the `-o` option (which stands for output), and name the output file. That’s it.

Let’s go back to our Terminal. Navigate to our git repo, type `ls`.

If we take this file (README.md), and we want to convert it to a PDF, how would we do that?

```
pandoc README.md -o LAHP-working-textsII.pdf
```

Most pandoc operations look like the above. There are more precise ways to guide the conversion: So if you want an html file of this README file,

```
pandoc README.md -f markdown -t html -s -o README.html
```

works like a charm, for the most part. The `-f` options specifies that you are converting from markdown syntax, and `-t` says that you are converting to html. The `-s` option specifies that it is a standalone file with a proper html header. In most cases, however, pandoc only needs the file extension to guide the conversion,

which is why the `-o` option is all you need in most cases. But sometimes you do need more precision, such as with PDF conversions (note that you'll need to have LaTeX installed. See MacTeX on OS X, MiKTeX on Windows, or install the texlive package in linux).

```
pandoc README.md -o README.pdf
```

What happens here?

Try this instead:

```
pandoc README.md --pdf-engine=xelatex -o README.pdf
```

I have already extolled the virtues of Markdown, and I hope you can see that you can use Pandoc to easily convert files from Markdown.