

English 3844: Writing and Digital Media Guidebook

Andy Lautenschlager

2018-08-31

Contents

1	Introduction	5
2	Readings	7
2.1	From “The Rhetorical Situation” by Lloyd Bitzer (1968)	7
3	Atom	9
3.1	Introducing Atom	9
3.2	Installing Atom	9
3.3	Installing packages	10
3.4	Optional packages	12
4	Markdown	13
4.1	Cheat sheet	14

Chapter 1

Introduction

Welcome to English 3844: Writing and Digital Media! In this class we write blogs, create podcasts and videos, and then curate all of this content on our own websites.

This booklet contains instructions and resources related to English 3844: Writing and Digital Media. Inside you'll find instructions on how to install and use Atom text editor, GitHub Desktop, and GitHub pages, as well as a few readings and a collection of audio and video development resources.

I'll add more and more content to this booklet as the semester progresses.

Chapter 2

Readings

We won't have many readings this semester, but I have compiled a few excerpts from longer works below.

2.1 From "The Rhetorical Situation" by Lloyd Bitzer (1968)

The study of rhetoric dates back thousands of years, predating even Socrates. Since then, countless scholars have tried to answer the question, "What makes discourse rhetorical?" Lloyd Bitzer offered one of the clearest answers to that question in 1968. Below is a collection of excerpts from his essay "The Rhetorical Situation".

Rhetoric alters reality

In order to clarify rhetoric-as-essentially-related-to-situation, we should acknowledge a viewpoint that is commonplace but fundamental: a work of rhetoric is pragmatic; it comes into existence for the sake of something beyond itself; it functions ultimately to produce action or change in the world; it performs some task. In short, rhetoric is a mode of altering reality, not by the direct application of energy to objects, but by the creation of discourse which changes reality through the mediation of thought and action. The rhetor alters reality by bringing into existence a discourse of such a character that the audience, in thought and action, is so engaged that it becomes mediator of change. In this sense rhetoric is always persuasive.

The rhetorical situation

Let us now amplify the nature of situation by providing a formal definition and examining constituents. Rhetorical situation may be defined as a complex of persons, events, objects, and relations presenting an actual or potential exigence which can be completely or partially removed if discourse, introduced into the situation, can so constrain human decision or action as to bring about the significant modification of the exigence. Prior to the creation and presentation of discourse, there are three constituents of any rhetorical situation: the first is the exigence; the second and third are elements of the complex, namely the audience to be constrained in decision and action, and the constraints which influence the rhetor and can be brought to bear upon the audience. Any exigence is an imperfection marked by urgency; it is a defect, an obstacle, something waiting to be done, a thing which is other than it should be.

Exigence

In any rhetorical situation there will be at least one controlling exigence which functions as the organizing principle: it specifies the audience to be addressed and the change to be effected. The exigence may or may not be perceived clearly by the rhetor or other persons in the situation; it may be strong or weak depending upon the clarity of their perception and the degree of their interest in it; it may be real or unreal depending on the facts of the case; it may be important or trivial; it may be such that discourse can completely remove it, or it may persist in spite of repeated modifications; it may be completely familiar - one of a type of exigences occurring frequently in our experience - or it may be totally new, unique. When it is perceived and when it is strong and important, then it constrains the thought and action of the perceiver who may respond rhetorically if he is in a position to do so.

Audience

The second constituent is the audience. Since rhetorical discourse produces change by influencing the decision and action of persons who function as mediators of change, it follows that rhetoric always requires an audience - even in those cases when a person engages himself or ideal mind as audience. It is clear also that a rhetorical audience must be distinguished from a body of mere hearers or readers: properly speaking, a rhetorical audience consists only of those persons who are capable of being influenced by discourse and of being mediators of change.

Constraints

Besides exigence and audience, every rhetorical situation contains a set of constraints made up of persons, events, objects, and relations which are parts of the situation because they have the power to constrain decision and action needed to modify the exigence. Standard sources of constraint include beliefs, attitudes, documents, facts, traditions, images, interests, motives and the like; and when the orator enters the situation, his discourse not only harnesses constraints given by situation but provides additional important constraints - for example his personal character, his logical proofs, and his style. There are two main classes of constraints: (1) those originated or managed by the rhetor and his method (Aristotle called these "artistic proofs"), and (2) those other constraints, in the situation, which may be operative (Aristotle's "inartistic proofs").

Chapter 3

Atom

3.1 Introducing Atom

Atom is a text editor. A text editor is a little bit like Microsoft Word, but for coding. With a little setup and practice, however, you may find yourself writing your English papers in Atom instead of Word. Atom is faster, simpler, prettier, and does most of what you need in terms of text production. At the very least, it's *much* better than Word for taking notes and writing things for the internet.

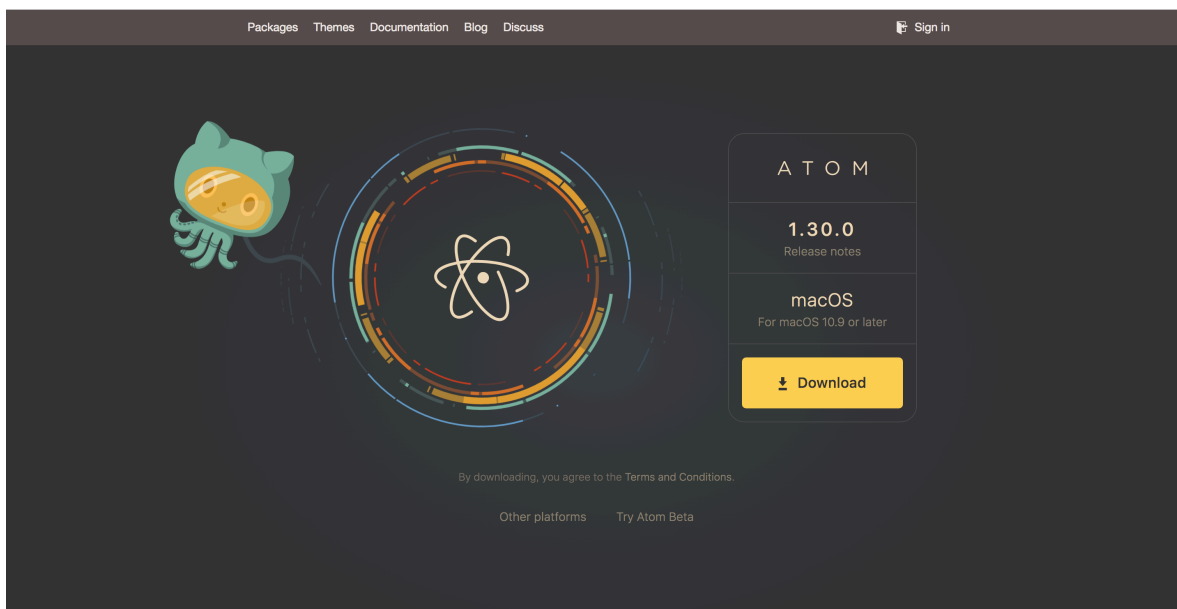
We'll eventually use it to code websites, but first we'll use it for writing blog posts. So let's install it and customize it for those tasks.

This process might be a little scary, but do the best you can. I'll include some links to help you if you get stuck.

3.2 Installing Atom

First, let's install the program and put it where it belongs on your computer.

1. If you don't have **Google Chrome** on your computer, download it. You don't *technically* need Chrome, but it's what we'll use to examine code later in the semester.
2. Open Chrome and go to atom.io. You should see a screen like the one below.



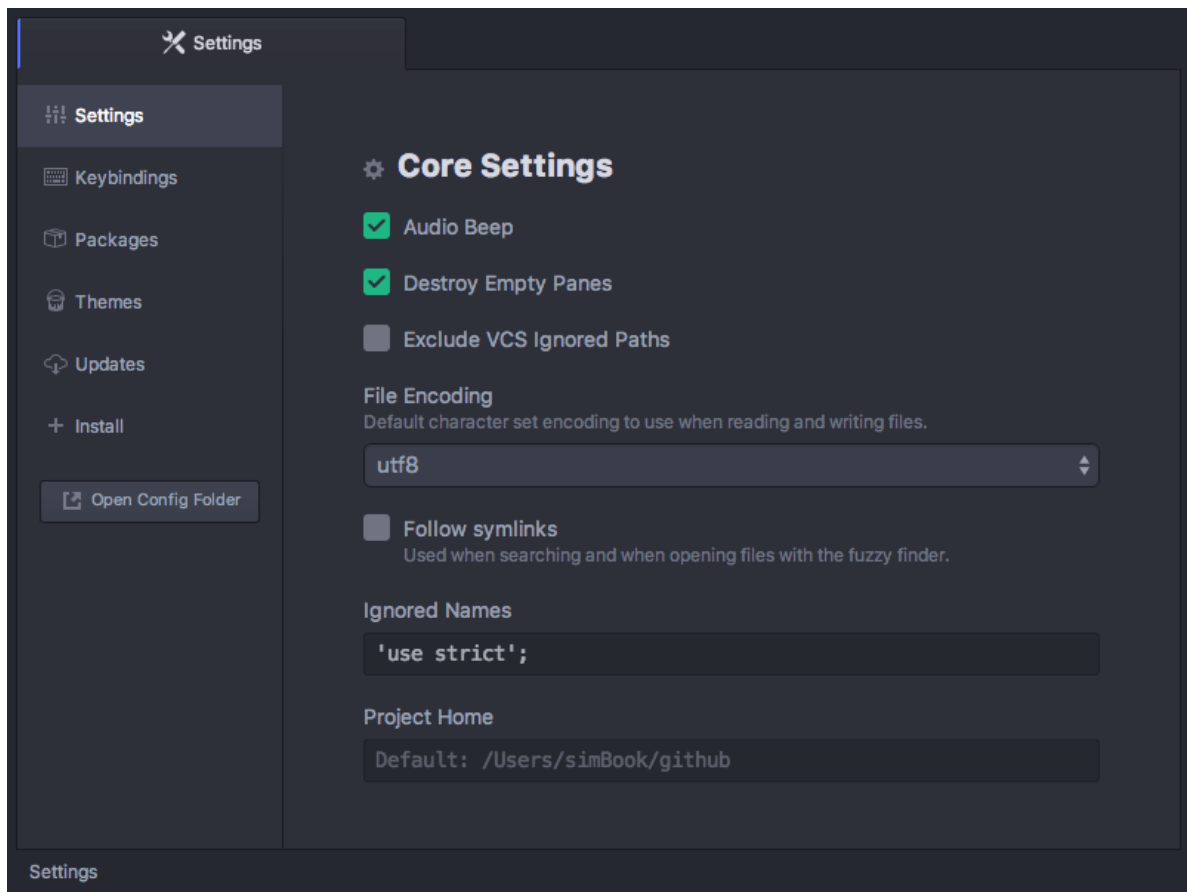
3. See the **Download** button? Click it. Your computer should then download a .zip (Mac) or .exe (Win) file. Some computers may automatically open and unpack the .zip file. If yours doesn't, then open the .zip file yourself. (If you don't know how to open .zip files on your computer, Google it.) Eventually, you should see the Atom icon.
4. If you're using a Mac, drag that icon to your **Applications** folder. If you're using Windows, Atom should automatically add an Atom shortcut to your **desktop** and your **Start menu**.
5. Click the Atom icon to launch Atom!

For more info/help, visit the Installing Atom section of the Atom documentation. Note that at the top of the page you can choose your operating system (Windows or Mac).

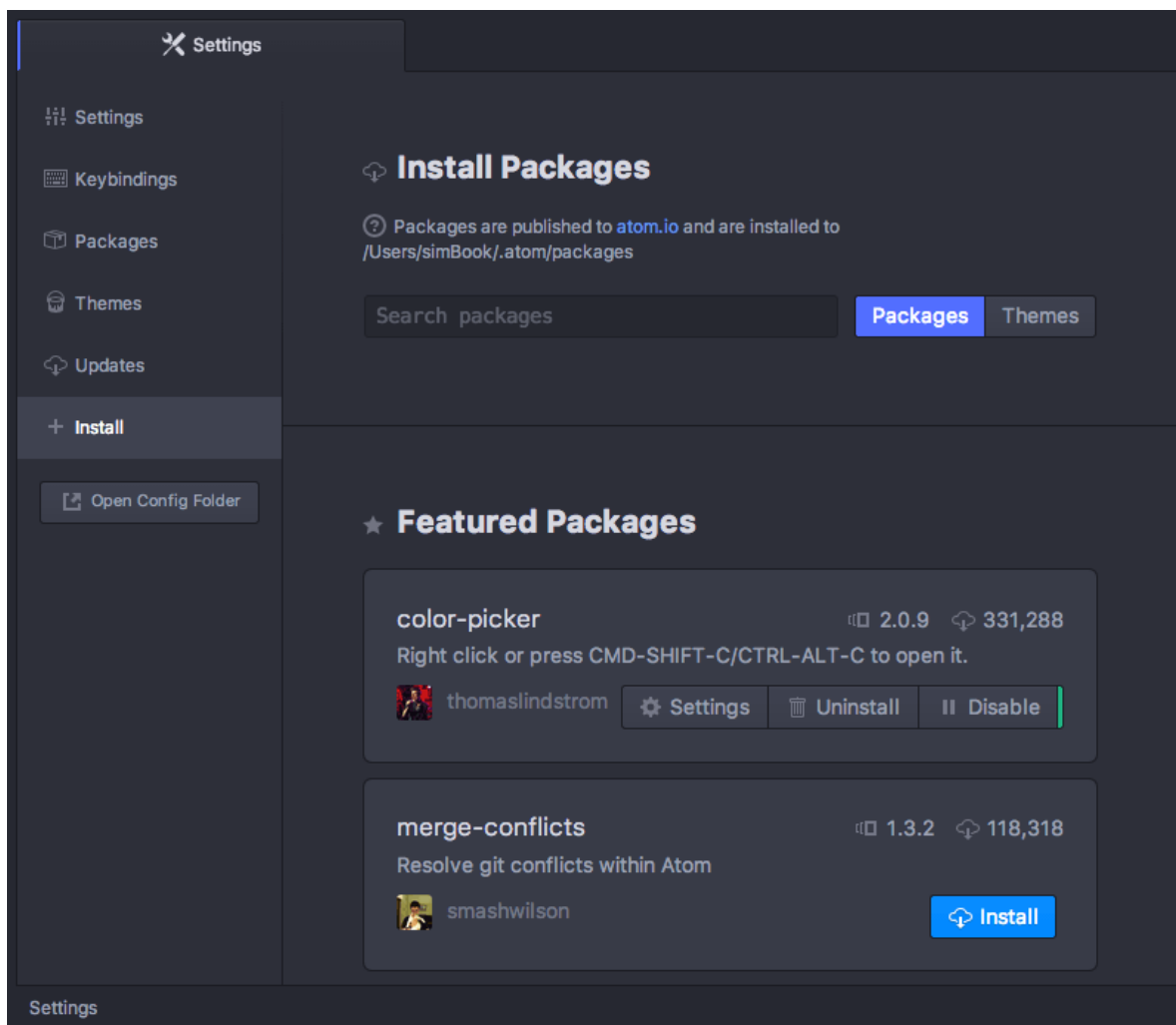
3.3 Installing packages

Atom's a little different than Word. Word comes with a whole bunch of features, most of which you'll never use. Atom comes with a few features but allows you to quickly install many more. You install those features via the **package manager**. Let's install most of the packages we'll need this semester.

1. Once you've launched Atom, you should see a screen with a Welcome Guide and other information. At the top of the screen, you should see a menu bar like you do with other applications (File, Edit, View, etc.). Open the Settings view by choosing **File => Settings (Win)** or **Atom => Preferences (Mac)**. Alternatively, if you want to be a baller, just hit **ctrl+comma (Win)** or **cmd+comma (Mac)**. You should see a screen like the one below.



2. First, click the Editor tab, scroll down to **Soft Wrap**, and check the corresponding box.
3. Next, click the Themes tab. Here you can choose a dark background or a light background. If you prefer a dark background, do nothing. If you prefer a light background, choose One Light. Be sure to change both the UI Theme and the Syntax Theme.
4. Finally, click the Install tab. You should see a screen like the one below.



5. In the Install Packages search bar, search for **atom-beautify**. When the package appears, click the **Install** button and wait for the installation to complete. Congrats—you’ve just installed a package!
6. Repeat step 5 for each of the packages below. Once you’ve installed the packages, you can view some of them in the Packages menu (in the same menu bar as File, Edit, View, etc.).
 - **atom-html-preview** – allows you to view changes to your website from within Atom
 - **emmet** – allows you to write your code more quickly
 - **linter** – helps identify potential errors in your code. When you install this one, Atom may ask you to install “dependencies.” Allow each of these
 - **markdown-writer** – allows you to make pretty documents with no fuss (we’ll use this one right away!)
 - **tool-bar** – with the next package, adds a toolbar with buttons for italics, etc.
 - **tool-bar-markdown-writer** – see directly above
 - **pandoc-convert** – converts Markdown files (see below) to Word docs, PDFs, or other formats

3.4 Optional packages

If you wish, you may also download these packages:

- **wordcount** – adds a word count to the bottom of Atom’s interface
- **linter-write-good** – tries to identify common writing issues (e.g., passive voice). Can be helpful, but when in doubt use your own judgment

Chapter 4

Markdown

Markdown is a lightweight markup language, which basically means that it allows you to easily create italics, boldface, links, images, and bulleted/numbered lists. Markdown is faster than Word for most kinds of writing, but the best part is that a Markdown file can become a Word doc, a PDF, an HTML file, a slide show, an MLA document—whatever!

To learn Markdown, complete the tutorial at [CommonMark](#). When you've completed the tutorial, make your first Markdown file in Atom by following the steps below.

1. In Atom, choose File => New file.
2. Save the file as a Markdown file by choosing File => Save. Name the file `markdown-test.md`. Be sure to use the `.md` suffix.
3. Paste the text below into the file, then save the file.

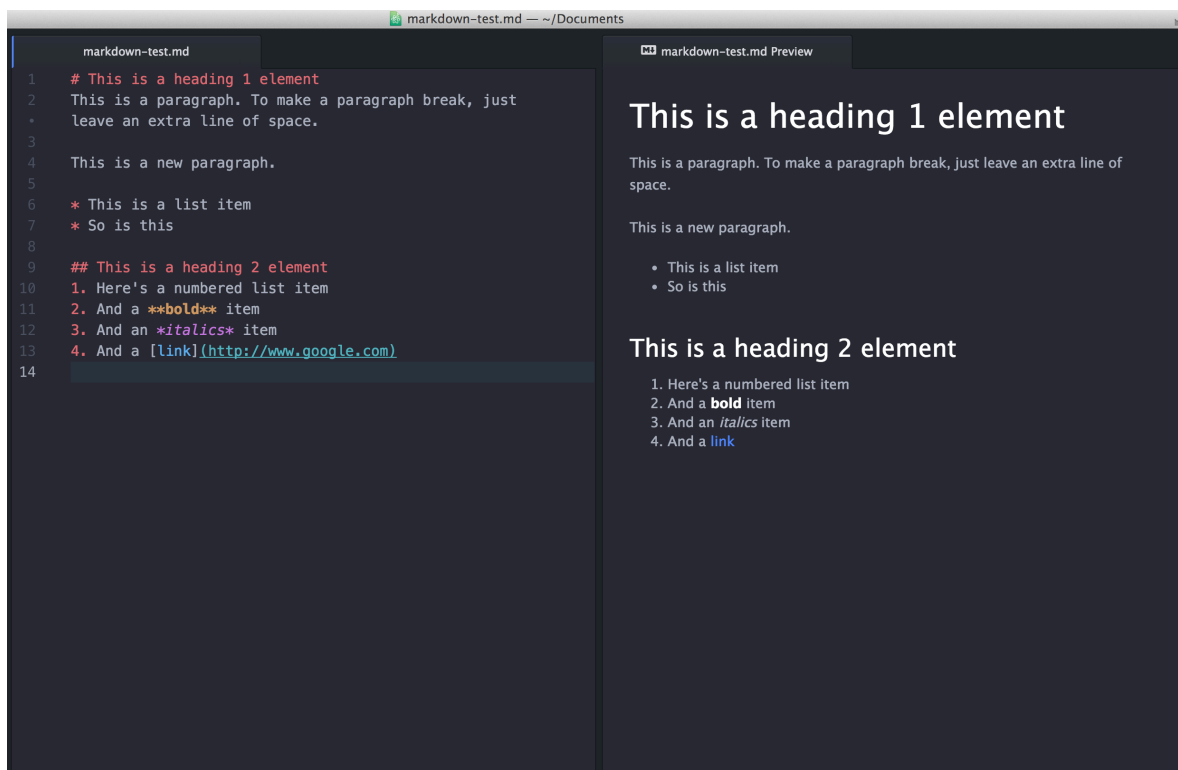
```
\# This is a heading 1 element
This is a paragraph. To make a paragraph break, just leave an extra line of space.

This is a new paragraph.

\* This is a list item
\* So is this

\## This is a heading 2 element
1\. Here's a numbered list item
2\. And a \*\*bold\*\* item
3\. And an \*italics\* item
4\. And a \[link\](http://www.google.com)
```

4. Now, to see the output, look for the Packages menu in the menu bar (File, Edit, View, etc.) at the top of the screen. Click Packages, then choose Markdown Preview. You should see a preview window like the one below.



5. That's it! Markdown is that simple. If you want to learn more, use the Markdown cheat sheet. Soon, we'll learn how to convert Markdown to a nice pdf or html document—and how to paste perfectly formatted writing into emails, newsletters, Google or Word docs, and more.

Super special bonus

Want to achieve perfect MLA formatting every time? Try markdowntomla.com!

You can also use Markdown to make slide decks *way* faster than you could with PowerPoint. Deckdown is a good place to start, but once we learn HTML and GitHub, we can add images, videos, etc.

4.1 Cheat sheet

This cheat sheet comes from Matt Cone at markdownguide.org. You can also find a cheat sheet within Atom by clicking **Packages => Markdown Writer => Open Cheat Sheet**.

4.1.1 Basic Syntax

These are the elements outlined in John Gruber's original design document. All Markdown applications support these elements.

Element

Markdown Syntax

Heading

H1
 ## H2
 ### H3

Bold

****bold text****

Italic

`*italicized text*`

Blockquote

`> blockquote`

Ordered List

1. First item 2. Second item 3. Third item

Unordered List

- First item - Second item - Third item

Code

`code`

Horizontal Rule

`---`

Link

`[title](https://www.example.com)`

Image

`![alt text](image.jpg)`