



Scrivere in modo sostenibile usando il testo semplice (plain text) con Pandoc e Markdown

Dennis Tenen e Grant Wythoff

In questo tutorial imparerai le basi di Markdown, una sintassi di markup per testo semplice, facile da leggere e scrivere e conoscerai anche Pandoc, uno strumento da riga di comando che converte il testo semplice in diversi di tipi di file magnificamente formattati: PDF, docx, HTML, LaTeX, slide deck e altro.

Testo originale: <https://programminghistorian.org/en/lessons/sustainable-authorship-in-plain-text-using-pandoc-and-markdown>

Traduzione italiana: <https://framapiaf.org/@nilocram>

Distribuito con licenza [CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

Sommario

- [Obiettivi](#)
- [Filosofia](#)
- [Principi](#)
- [Requisiti del software](#)
- [Nozioni di base di Markdown](#)
- [Familiarizzare con il terminale del tuo computer](#)
- [Usare Pandoc per convertire Markdown in un documento MS Word](#)
- [Lavorare con le bibliografie](#)
- [Cambiare stili di citazione](#)
- [Sommario](#)
- [Risorse utili](#)



Obiettivi

In questo tutorial imparerai le basi di Markdown, una sintassi di markup per testo semplice (plain text) facile da leggere e scrivere e conoscerai anche [Pandoc](#), uno strumento da riga di comando che converte il testo semplice in diversi tipi di file magnificamente formattati: PDF, docx, HTML, LaTeX, slide deck e altro. [1](#) Con Pandoc come strumento di composizione digitale, è possibile utilizzare la sintassi di Markdown per aggiungere numeri, una bibliografia, la formattazione e modificare facilmente gli stili di citazione da Chicago a MLA (ad esempio), il tutto usando il testo semplice.

Il tutorial non presuppone alcuna conoscenza tecnica preliminare, ma progredisce con l'esperienza, poiché spesso suggeriamo tecniche più avanzate verso la fine di ogni sezione. Queste sono chiaramente contrassegnate e possono essere rivisitate in un secondo momento con un po' di pratica e sperimentazione.

Invece di seguire questo tutorial in modo meccanico, ti consigliamo di cercare di comprendere le soluzioni offerte qui come una *metodologia* che potrebbe dover essere adattata ulteriormente al tuo ambiente e al tuo flusso di lavoro. L'installazione degli strumenti necessari presenta forse il più

grande ostacolo alla partecipazione. Concediti abbastanza tempo e pazienza per installare tutto correttamente, oppure fallo con un collega che ha una configurazione (del computer) simile alla tua e aiutatevi reciprocamente. Se hai qualche difficoltà, consulta qui sotto la sezione [Risorse utili](#). [2](#)

Filosofia

Scrivere, archiviare e recuperare documenti sono attività centrali nel flusso di lavoro della ricerca umanistica. Eppure, molti autori basano la loro pratica su strumenti e formati proprietari che a volte non soddisfano nemmeno i requisiti di base della scrittura accademica. Forse puoi rimanere frustrato dalla fragilità di note a piè di pagina, bibliografie, immagini e bozze di libri scritti in Microsoft Word o Google Documenti. Ciò nonostante, la maggior parte delle riviste continua ad accettare testi in formato .docx.

Più che causare frustrazione personale, questa dipendenza da strumenti e formati proprietari ha implicazioni negative a lungo termine per la comunità accademica. In tale ambiente le riviste devono esternalizzare la composizione, allontanare gli autori dai contesti materiali della pubblicazione e aggiungere ulteriori ostacoli inutili alla libera circolazione della conoscenza. [3](#)

Quando utilizzi MS Word, Google Documenti o OpenOffice per scrivere documenti, ciò che vedi non è ciò che ottieni. Sotto lo strato visibile di parole, frasi e paragrafi si trova un livello complicato di codice comprensibile solo alle macchine. A causa di tale livello nascosto, per essere visualizzati correttamente i file .docx e .pdf dipendono da strumenti proprietari. Tali documenti sono difficili da cercare, stampare e convertire in file di altro formato.

Inoltre, il tempo impiegato per la formattazione del documento in MS Word o OpenOffice è sprecato, poiché tutta la formattazione viene rimossa dall'editore a cui viene inviato. Sia gli autori che gli editori trarrebbero vantaggio dallo scambio di file con una formattazione minima, lasciando la composizione alla fase finale del processo di pubblicazione.

Questo è il passaggio dove Markdown si mette in luce. Markdown è una sintassi per marcare gli elementi semantici all'interno di un documento in modo esplicito, non in un livello nascosto. L'idea è di identificare le unità che sono significative per gli esseri umani, come titoli, sezioni, sottosezioni, note a piè di pagina e illustrazioni. Per lo meno, i tuoi file rimarranno sempre comprensibili per te, anche se l'editor che stai attualmente utilizzando smette di funzionare o "chiude".

Scrivere in questo modo libera l'autore dallo strumento. Markdown può essere scritto in qualsiasi editor di testo e offre un ricco ecosistema di software in grado di trasformare quel testo in bellissimi documenti. Per questo motivo, Markdown sta attualmente godendo di un periodo di crescita, non solo come mezzo per scrivere articoli accademici, ma come convenzione per l'editing online in generale.

I più popolari editor di testo semplice includono [TextWrangler](#) e [Sublime](#) per Mac, [Notepad ++](#) per Windows, oltre a Gedit e Kate per Linux. Tuttavia, ci sono anche editor specializzati nella visualizzazione e modifica di Markdown.

È importante capire che Markdown è semplicemente una convenzione. I file Markdown vengono archiviati come testo semplice, aggiungendo ulteriore flessibilità al formato. I file di testo semplice sono in circolazione dall'epoca della macchina da scrivere elettronica. La longevità di questo standard rende intrinsecamente il testo semplice più sostenibile e stabile rispetto ai formati proprietari. Mentre i file prodotti anche dieci anni fa in Microsoft Word e le pagine di Apple possono causare problemi significativi quando vengono aperti con la versione più recente, è ancora possibile aprire un file scritto da qualsiasi editor di testo semplice "morto" magari da alcuni decenni: AlphaPlus, Perfect Writer, Text Wizard, Spellbinder, WordStar, o lo SCRIPSIT 2.0 preferito da Isaac Asimov, prodotto da Radio Shack. La scrittura in testo semplice garantisce che i tuoi file rimarranno leggibili tra dieci, quindici, venti anni. In questo tutorial, delineiamo un flusso di lavoro che libera il ricercatore dal programma di elaborazione testi proprietario e dai formati di file fragili.

Ora è possibile scrivere una vasta gamma di documenti in un unico formato: articoli, post di blog, wiki, programmi e lettere di raccomandazione, utilizzando lo stesso insieme di strumenti e tecniche per cercare, scoprire, eseguire il backup e distribuire i nostri materiali. Le note, i post di blog, la documentazione del codice e i wiki possono essere tutti creati in Markdown. Sempre più piattaforme come WordPress, Reddit e GitHub supportano in modo nativo la scrittura in Markdown. A lungo termine, la tua ricerca trarrà vantaggio da tali flussi di lavoro unificati, rendendo più semplice il salvataggio, la ricerca, la condivisione e l'organizzazione dei tuoi materiali.

I principi

Ispirati dalle migliori pratiche in una varietà di discipline, siamo stati guidati dai seguenti principi:

1. *Sostenibilità.* Il testo semplice garantisce la trasparenza e risponde agli standard di conservazione a lungo termine. MS Word potrebbe in futuro fare la fine di Word Perfect, ma il testo semplice rimarrà sempre facile da leggere, catalogare, estrarre e trasformare. Inoltre, il testo semplice consente il controllo delle versioni del documento, utile per la collaborazione e nell'organizzazione delle bozze. I tuoi file di testo in chiaro saranno accessibili su telefoni cellulari, tablet o, forse, su un terminale di bassa potenza in alcune biblioteche remote. Il testo semplice è retrocompatibile e a prova di futuro. Qualunque sia, il software o l'hardware successivo sarà in grado di capire i tuoi file di testo.
2. *Preferenza per i formati leggibili dall'uomo.* Quando scrivi in Word o Google Documenti, ciò che vedi non è ciò che ottieni. Il file .doc contiene caratteri di formattazione nascosti generati automaticamente, creando un livello di composizione offuscato che rende difficile all'utente la risoluzione di eventuali problemi. Qualcosa di semplice come incollare un'immagine o un testo dal browser può avere effetti imprevedibili sulla formattazione del documento.
3. *Separazione di forma e contenuto.* Scrivere e formattare allo stesso tempo è fonte di distrazione. L'idea è di scrivere prima e formattare in seguito, il più vicino possibile al momento della pubblicazione. Un compito come il passaggio dal formato Chicago a quello MLA dovrebbe essere indolore. I redattori di riviste che desiderano risparmiare tempo su

formattazione e copia inutili dovrebbero essere in grado di fornire agli autori un modello di formattazione che si occupi della minuzie della composizione.

4. *Supporto per l'apparato accademico.* Il flusso di lavoro deve gestire con eleganza note a piè di pagina, immagini, caratteri internazionali e bibliografie.
5. *Indipendenza dalla piattaforma* A mano a mano che i vettori di pubblicazione si moltiplicano, dobbiamo essere in grado di generare una molteplicità di formati tra cui proiezione di diapositive, stampa, web e mobile. Idealmente, vorremmo essere in grado di generare i formati più comuni senza rompere i riferimenti bibliografici. Anche il nostro flusso di lavoro deve essere portatile: sarebbe bello poter copiare una cartella su una chiavetta e sapere che contiene tutto il necessario per la pubblicazione. Scrivere in testo semplice significa che puoi condividere, modificare e archiviare facilmente i tuoi documenti praticamente in qualsiasi ambiente. Ad esempio, un programma scritto in Markdown può essere salvato come PDF, stampato come volantino e convertito in HTML per il web, tutto dallo stesso file. Entrambi i documenti web e di stampa devono essere pubblicati dalla stessa fonte e avere un aspetto simile, preservando il layout logico del materiale.

Markdown e LaTeX rispondono a tutti questi requisiti. Abbiamo scelto Markdown (e non LaTeX) perché offre la sintassi più leggera e disordinata (quindi, *mark down*) e perché se abbinato a Pandoc consente la massima flessibilità di output (inclusi i file .docx e .tex). [4](#)

Requisiti del software

Omettiamo intenzionalmente alcuni dettagli granulari, relativi all'installazione del software elencato di seguito nelle diverse piattaforme o sistemi operativi. Ad esempio, non ha senso fornire istruzioni di installazione per LaTeX, quando le istruzioni online canoniche per il proprio sistema operativo saranno sempre più attuali e più complete. Allo stesso modo, la meccanica dell'installazione di Pandoc può essere esplorata meglio cercando "installare Pandoc" su Google, dove probabilmente il primo risultato sarà la homepage di Pandoc.

- **Editor di testo semplice.** Entrare nel mondo dell'editing del testo semplice espande in modo spettacolare la tua scelta di strumenti di authoring innovativi. Cerca online "editor di testo markdown" e sperimenta le tue opzioni. Non importa quello che usi purché sia esplicitamente un editor di testo semplice. Notepad ++ su Windows o TextWrangler su Mac sono scelte semplici e gratuite. Ricorda che dal momento che non siamo legati allo strumento, puoi cambiare editor in qualsiasi momento.
- **Terminale della riga di comando.** Lavorare "nella riga di comando" equivale a digitare comandi nel terminale. Su un Mac devi semplicemente utilizzare il Finder per cercare "Terminale". Su Windows, usa PowerShell. È probabile che gli utenti Linux abbiano già familiarità con i loro terminali. Tratteremo qui sotto le basi su come trovare e utilizzare la riga di comando.
- **Pandoc.** Informazioni dettagliate sull'installazione, specifiche per ogni piattaforma, sono disponibili sul [sito Web di Pandoc](#). *L'installazione di Pandoc sulla tua macchina è*

fondamentale per questo tutorial, quindi assicurati di prendere il tuo tempo e fare clic sulle istruzioni. Pandoc è stato creato ed è gestito da John MacFarlane, professore di filosofia presso l'Università della California, a Berkeley. Questo è il meglio dell'informatica per le scienze umane e servirà da motore del nostro flusso di lavoro. Con Pandoc, sarai in grado di compilare testo e bibliografia in documenti flessibili e magnificamente formattati. Una volta seguite le istruzioni di installazione, verifica che Pandoc sia installato inserendo `pandoc --version` nella riga di comando. presumiamo che tu abbia almeno la versione 1.12.3, rilasciata a gennaio 2014.

I seguenti due programmi sono consigliati, ma non sono necessari per completare questo tutorial.

- **Zotero o Endnote.** Software per i riferimenti bibliografici come Zotero e Endnote sono strumenti indispensabili per l'organizzazione e la formattazione di citazioni in un documento di ricerca. Questi programmi possono esportare le tue biblioteche come file BibTeX (di cui parleremo più avanti nel caso 2). Questo file, un documento formattato in testo semplice che raccoglie tutte le tue citazioni, ti consentirà di citare rapidamente e facilmente i riferimenti usando `@tags`. Va notato che è anche possibile digitare tutti i riferimenti bibliografici a mano, usando la [nostra bibliografia](#) come modello.
- **LaTeX.** Istruzioni di installazione dettagliate e specifiche per le diverse piattaforme sono disponibili sul [sito Web di Pandoc](#). Sebbene LaTeX non sia trattato in questo tutorial, è utilizzato da Pandoc per la creazione di .pdf. Gli utenti esperti spesso convertiranno direttamente in LaTeX per avere un controllo più granulare sulla composizione del file .pdf. I principianti potrebbero considerare di saltare questo passaggio. Altrimenti, digitare `latex -v` per vedere se LaTeX è stato installato correttamente (si otterrà un errore se non lo fosse e alcune informazioni sulla versione se lo fosse).

Nozioni di base su Markdown

Markdown è una convenzione per strutturare semanticamente i tuoi documenti in testo semplice. L'idea è di identificare strutture logiche nel tuo documento (titolo, sezioni, sottosezioni, note a piè di pagina, ecc.), contrassegnarle con alcuni caratteri non invadenti e quindi "compilare" il testo risultante con un interprete di composizione che formatterà il documento in modo coerente, secondo uno stile specificato.

Le convenzioni di Markdown sono disponibili in diverse "varianti" progettate per l'utilizzo in determinati contesti, come blog, wiki o repository di codice. La variante di Markdown utilizzata da Pandoc è orientata all'uso accademico. Le sue convenzioni sono descritte nella pagina [Markdown di Pandoc](#). Le sue convenzioni includono il [blocco "YAML"](#), che contiene alcuni metadati utili.

Creiamo ora un documento semplice in Markdown. Apri un editor di testo semplice di tua scelta e inizia a digitare. Dovrebbe sembrare come questo:

```
---
title: Plain Text Workflow
author: Dennis Tenen, Grant Wythoff
date: January 20, 2014
---
```

Il Markdown di Pandoc memorizza ciascuno dei valori sopra riportati e li "stampa" nella posizione appropriata del documento prodotto una volta che si è pronti per la composizione. In seguito impareremo ad aggiungere altri campi più potenti al blocco YAML. Per ora, facciamo finta di scrivere un documento che contiene tre sezioni, ciascuna suddivisa in due sottosezioni. Lascia una riga vuota dopo gli ultimi tre trattini nel blocco YAML e incolla quanto segue:

```
# Section 1
```

```
## Subsection 1.1
```

```
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
```

Il prossimo paragrafo dovrebbe iniziare così. Non indentare.

```
## Subsection 1.2
```

```
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium
doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore
veritatis et quasi architecto beatae vitae dicta sunt explicabo.
```

```
# Section 2
```

```
## Subsection 2.1
```

Vai avanti e inserisci anche un testo fittizio. Lo spazio vuoto è significativo in Markdown: non indentare i paragrafi. Invece, separa i paragrafi usando una riga vuota. Le righe vuote devono anche precedere le intestazioni di sezione.

È possibile utilizzare gli asterischi per aggiungere un'enfasi in grassetto o in corsivo alle parole, come segue: *italics* e **bold**. Dovremmo anche aggiungere un link e una nota a piè di pagina al nostro testo per completare i componenti di base di un documento medio. Scrivi:

```
Una frase che ha bisogno di una nota. [^1]
```

```
[^1]: la mia prima nota a piè di pagina! E un [link] (https://www.eff.org/).
```

Quando il testo del link e l'indirizzo sono gli stessi, è più veloce scrivere `<www.eff.org>` anziché `www.eff.org`.

Salviamo il nostro file prima di procedere ulteriormente. Crea una nuova cartella che ospiterà questo progetto. È probabile che tu abbia un sistema per organizzare i tuoi documenti, progetti, illustrazioni e bibliografie. Ma spesso il tuo documento, le sue illustrazioni e la sua bibliografia si trovano in cartelle diverse, il che rende difficile tenerne traccia. Il nostro obiettivo è creare una singola cartella per ogni progetto, con tutti i materiali pertinenti inclusi. La regola generale è un progetto, un testo, una cartella. Chiama il tuo file `main.md` o qualcosa del genere, dove "md" sta per markdown.

Una volta salvato il file, aggiungiamo un'illustrazione. Copia un'immagine (qualsiasi immagine piccola) nella tua cartella e aggiungi il seguente testo nel corpo del testo `![image caption](your_image.jpg)`.

A questo punto, il tuo `main.md` dovrebbe apparire come il file seguente. Puoi scaricare questo file `.md` di esempio [qui](#).

```
---
title: Plain Text Workflow
author: Dennis Tenen, Grant Wythoff
date: January 20, 2014
---

# Section 1

## Subsection 1.1
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat.

## Subsection 1.2
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium
doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore
veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Il prossimo paragrafo dovrebbe iniziare così. Non indentare.

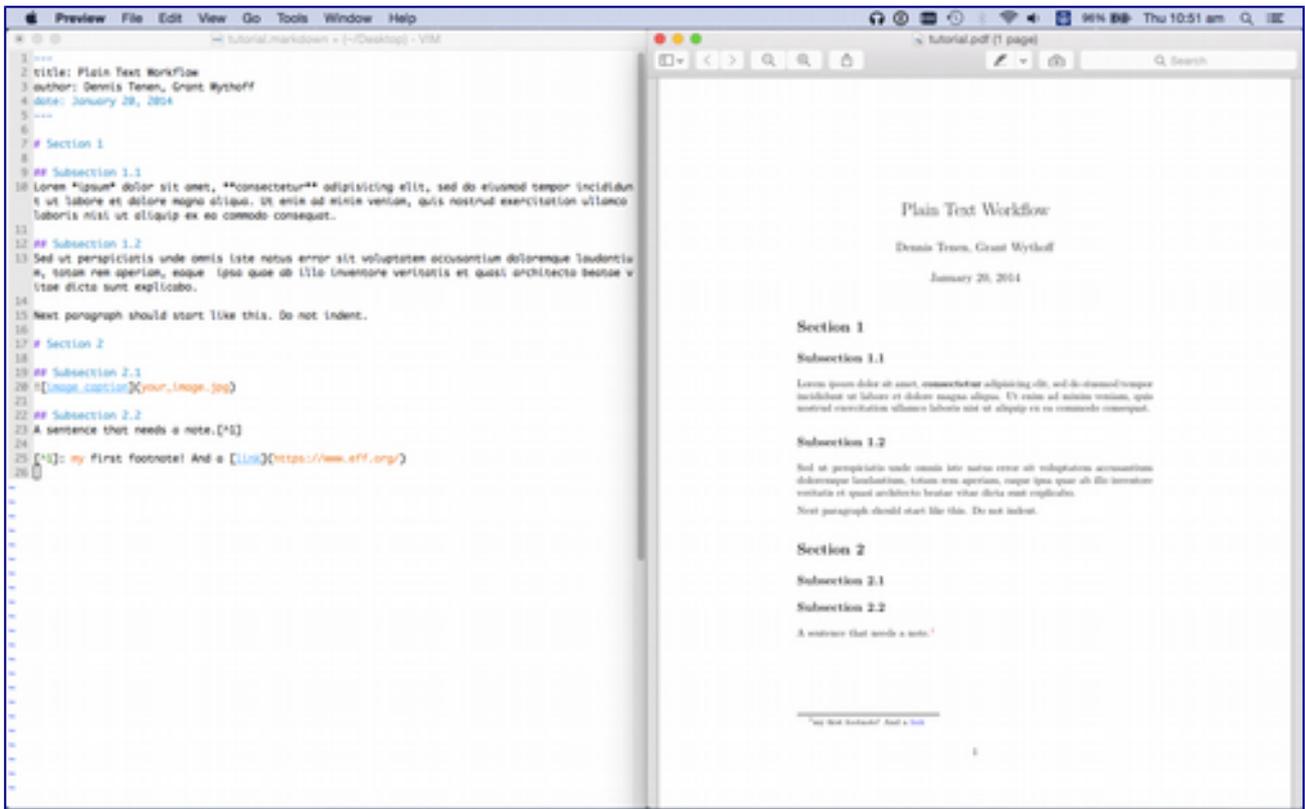
# Section 2

## Subsection 2.1
![image caption](your_image.jpg)

## Subsection 2.2
Una frase che ha bisogno di una nota.[^1]

[^1]: la mia prima nota a piè di pagina! E un [link](https://www.eff.org/)
```

Come faremo tra poco, questo semplice file di testo può essere trasformato in un bel PDF:



Schermata di un PDF prodotto da Pandoc

Se vuoi avere un'idea di come questo tipo di markup verrà interpretato come testo formattato in HTML, prova [questa sandbox online](#) e gioca con vari tipi di sintassi. Ricorda che alcuni elementi del Markdown in versione *Pandoc* (come il blocco del titolo e le note a piè di pagina) non funzioneranno in questo modulo web, che accetta solo le funzioni di base.

A questo punto, dovresti passare un po' di tempo ad esplorare alcune delle altre funzionalità di Markdown come le citazioni (rappresentate con il simbolo >), elenchi puntati che iniziano con * o - , interruzioni di riga che iniziano con | (utile per la poesia), tabelle e alcune delle altre funzioni elencate sulla pagina del markdown di Pandoc.

Presta particolare attenzione allo spazio vuoto e al flusso dei paragrafi. La documentazione lo afferma in modo sintetico quando definisce un paragrafo come "una o più righe di testo seguite da una o più righe vuote". Nota che "gli a capo sono trattati come spazi" e che "se hai bisogno di un'interruzione di riga devi inserire due o più spazi alla fine di una riga". Il modo migliore per capire cosa significa è sperimentare liberamente. Usa la modalità di anteprima del tuo editor o esegui semplicemente Pandoc per vedere i risultati dei tuoi esperimenti.

Soprattutto, ti evita la necessità di formattare. Ricorda che stai identificando unità *semantiche* : sezioni, sottosezioni, enfasi, note a piè di pagina e illustrazioni. Anche **corsivo** e ***grassetto*** in Markdown non sono realmente segni di formattazione, ma indicano un diverso livello di *enfasi*. La formattazione avverrà più tardi, una volta che tu conosca la collocazione e i requisiti della pubblicazione.

Ci sono programmi che ti permettono di vedere un'anteprima dell'output di Markdown mentre modifichi il tuo file di testo normale, li indichiamo di seguito nella sezione Risorse utili. Pochi tuttavia supportano note a piè di pagina, illustrazioni e bibliografie. Per trarre il massimo vantaggio da Pandoc, si consiglia di conservare sul proprio computer i file di testo semplici salvati localmente.

Familiarizza con il tuo terminale

Prima di poter iniziare a pubblicare il nostro file `main.md` in altri formati, dobbiamo saperci orientare nell'uso della riga di comando usando il programma terminale del computer, che è l'unico (e migliore) modo di usare Pandoc.

La riga di comando è un posto amichevole, una volta che ci si abitua. Se hai già familiarità con l'utilizzo della riga di comando, salta pure questa sezione. Per gli altri, è importante capire che essere in grado di utilizzare direttamente il proprio programma terminale consente a tutti di utilizzare una vasta gamma di potenti strumenti di ricerca che non è possibile utilizzare diversamente e può servire come base per un lavoro più avanzato. Ai fini di questo tutorial, è necessario imparare solo alcuni comandi molto semplici.

Innanzitutto, apri una finestra della riga di comando. Se utilizzi un Mac, apri Terminale nella directory "Applicazioni / Utility". Su Windows, utilizzerai PowerShell. In Windows 7 o versioni successive, fai clic su Start, digita "powershell" in "Cerca programmi e file" e premi invio. Per un'introduzione dettagliata all'uso della riga di comando, vedi l'eccellente [Command Line Crash Course](#) di Zed A. Shaw .

Una volta aperto, dovresti vedere una finestra di testo e un prompt che assomiglia a questo: `nome-compute:~username$` . La tilde indica la tua directory "home" e in effetti puoi digitare `$ cd ~` in qualsiasi momento per tornare alla tua home directory. Non digitare il simbolo del dollaro, simboleggia semplicemente il prompt dei comandi del terminale, ti permette di digitare qualcosa nel terminale (anziché digitarlo nel documento); ricorda di premere Invio dopo ogni comando.

È molto probabile che la cartella "Documenti" si trovi qui. Digita `$ pwd` (= stampa la directory di lavoro) e premi invio per visualizzare il nome della directory corrente). Usa `$ pwd` ogni volta che ti senti perso.

Il comando `$ ls` (= lista) elenca semplicemente i file nella directory corrente. Infine, puoi usare `$ cd` (= cambia directory) come `$ cd DIRECTORY_NAME` (dove `DIRECTORY_NAME` è il nome della directory in cui desideri navigare). Puoi usare `$ cd ..` per spostarti automaticamente di un livello nella struttura della directory (la directory principale della directory in cui ti trovi attualmente). Dopo aver iniziato a digitare il nome della directory, utilizzare il tasto Tab per completare automaticamente il testo, è particolarmente utile per i nomi lunghi di directory o per i nomi di directory che contengono spazi. [5](#)

Questi tre comandi del terminale: `pwd` , `ls` e `cd` sono tutto ciò che serve per questo tutorial. Usali per qualche minuto per navigare nella cartella dei documenti e riflettere sul modo in cui hai organizzato i file. Se lo desideri, segui il e file manager grafico per mantenere il tuo orientamento.

Utilizzo di Pandoc per convertire Markdown in un documento MS Word

Ora siamo pronti a comporre! Apri la finestra del terminale, usa `$ pwd` e `$ cd` per navigare nella cartella corretta per il tuo progetto. Una volta che sei lì, digita `$ ls` nel terminale per vedere l'elenco dei file. Se vedi il tuo file `.md` e le tue immagini, sei nel posto giusto. Per convertire `.md` in `.docx` digita:

```
$ pandoc -o main.docx main.md
```

Apri il file con MS Word per verificare i risultati. In alternativa, se usi Open o LibreOffice puoi eseguire:

```
$ pandoc -o project.odt main.md
```

Se sei nuovo nella riga di comando, immagina di leggere il comando precedente come qualcosa del tipo: "Pandoc, crea un file MS Word dal mio file Markdown". La parte `-o` è un "flag", che in questo caso dice qualcosa come "invece di farmi indicare esplicitamente i formati dei file sorgente e destinazione, ricavali guardando l'estensione dei file". Sono disponibili molte opzioni attraverso questi flag in Pandoc. È possibile visualizzare l'elenco completo sul [sito Web di Pandoc](#) o digitando

```
$ man pandoc
```

nel terminale.

Prova a eseguire il comando

```
$ pandoc -o project.html main.md
```

Ora torna alla directory del tuo progetto. Puoi dire cos'è successo?

Gli utenti più esperti che hanno installato LaTeX potrebbero voler sperimentare convertendo Markdown in `.tex` o in file `.pdf` appositamente formattati. Una volta installato LaTeX, è possibile creare un file PDF ben formattato utilizzando la stessa struttura di comando:

```
$ pandoc -o main.pdf main.md
```

Con il tempo, sarai in grado di mettere a punto la formattazione dei documenti PDF specificando un file in stile LaTeX (salvato nella stessa directory) e eseguendo qualcosa come:

```
$ pandoc -H format.sty -o project.pdf --number-sections --toc project.tex
```

Lavorare con le bibliografie

In questa sezione, aggiungeremo una bibliografia al nostro documento e quindi convertirlo dal formato Chicago a quello MLA.

Se non stai usando un gestore di riferimenti bibliografici come Endnote o Zotero, dovresti farlo. Preferiamo Zotero, perché, come Pandoc, è stato creato dalla comunità accademica e come altri progetti open-source è rilasciato sotto la GNU General Public License. Molto importante per noi, il

tuo gestore di riferimenti deve avere la capacità di generare bibliografie in formato di testo normale, per rimanere in linea con il nostro principio "tutto in testo semplice". Vai avanti e apri un gestore di riferimenti a tua scelta e aggiungi alcune voci di esempio. Quando sei pronto, trova l'opzione per esportare la tua bibliografia in formato BibTeX (.bib). Salva il tuo file .bib nella directory del tuo progetto e assegnagli un titolo ragionevole come "project.bib".

L'idea generale è di mantenere le tue fonti organizzate in un unico database bibliografico centralizzato, generando file .bib specifici e molto più piccoli che risiederanno nella stessa directory del progetto. Vai avanti e apri il tuo file .bib con l'editor di testo semplice di tua scelta. [6](#)

Il tuo file .bib dovrebbe contenere più voci che assomigliano a questo:

```
@article{fyfe_digital_2011,  
  title = {Digital Pedagogy Unplugged},  
  volume = {5},  
  url = {http://digitalhumanities.org/dhq/vol/5/3/000106/000106.html},  
  number = {3},  
  urldate = {2013-09-28},  
  author = {Fyfe, Paul},  
  year = {2011},  
  file = {fyfe_digital_pedagogy_unplugged_2011.pdf}  
}
```

Raramente dovrai modificarli a mano (anche se puoi farlo). Nella maggior parte dei casi, semplicemente "esporterai" il file .bib da Zotero o da un gestore di riferimenti simile. Prenditi un momento per orientarti qui. Ogni voce è composta da un tipo di documento, "articolo" nel nostro caso, un identificatore univoco (fyfe_digital_2011) e i relativi metadati su titolo, volume, autore e così via. La cosa che ci interessa di più è l'ID univoco che segue immediatamente la parentesi graffa nella prima riga di ogni voce. L'ID univoco è ciò che ci consente di collegare la bibliografia con il documento principale. Lascia questo file aperto per ora e torna al tuo file `main.md`

Modifica la nota a piè di pagina nella prima riga del tuo file `main.md` in modo che assomigli ai seguenti esempi, dove `@name_title_date` può essere sostituito con uno degli ID univoci dal file `project.bib`.

- A reference formatted like this will render properly as inline- or footnote- style citation [`@name_title_date`, 67]. [7](#)
- "For citations within quotes, put the comma outside the quotation mark" [`@name_title_2011`, 67].

Una volta eseguito il markdown tramite Pandoc, "`@ fyfe_digital_2011`" verrà espanso in una citazione completa nello stile di tua scelta. Puoi utilizzare la sintassi `@citation` in qualsiasi modo tu `@citation` ritenga opportuno: in linea con il tuo testo o nelle note a piè di pagina. Per generare una bibliografia includi semplicemente una sezione chiamata `# Bibliography` alla fine del documento.

Ora, torna all'intestazione dei metadati nella parte superiore del documento .md e specifica il file bibliografico da utilizzare, in questo modo:

```
---
title: Plain Text Workflow
author: Dennis Tenen, Grant Wythoff
date: January 20, 2014
bibliography: project.bib
---
```

Questo dice a Pandoc di cercare la tua bibliografia nel file `project.bib`, nella stessa directory di `main.md`. Vediamo se funziona. Salva il tuo file, passa alla finestra del terminale ed esegui:

```
$ pandoc -S -o main.docx --filter pandoc-citeproc main.md
```

Il flag maiuscolo `S` indica "smart", una modalità che produce "output tipograficamente corretto, conversione di virgolette semplici in virgolette ricurve, - in trattini elettronici, - in trattini e ... in ellissi." Il filtro "pandoc-citeproc" analizza tutti i tag delle citazioni. Il risultato dovrebbe essere un file MS Word formattato in modo decente. Se hai installato LaTeX, converti in `.pdf` usando la stessa sintassi per risultati più precisi. Non preoccuparti se le cose non sono esattamente come ti piacciono, ricorda, preciserai la formattazione tutto in una volta e in un secondo momento, il più vicino possibile al momento della pubblicazione. Per ora stiamo solo creando bozze basate su valori di default ragionevoli.

Cambiare gli stili di citazione

Lo stile di citazione predefinito in Pandoc è Chicago autore-data. Possiamo specificare uno stile diverso usando il foglio di stile, scritto nel "Citation Style Language" (un'altra convenzione in testo semplice, in questo caso per la descrizione degli stili di citazione) e denotata dall'estensione del file `.csl`. Fortunatamente, il progetto CSL mantiene un repository di stili di citazione comuni, alcuni persino adattati a riviste specifiche. Visita <http://editor.citationstyles.org/about/> per trovare il file `.csl` per Modern Language Association, scarica `modern-language-association.csl` e salva nella tua directory del progetto come `mla.csl`. Ora dobbiamo dire a Pandoc di usare il foglio di stile MLA invece del Chicago predefinito. Facciamo questo aggiornando l'intestazione YAML:

```
---
title: Plain Text Workflow
author: Dennis Tenen, Grant Wythoff
date: January 20, 2014
bibliography: project.bib
csl: mla.csl
---
```

Quindi usa semplicemente lo stesso comando:

```
$ pandoc -S -o main.docx --filter pandoc-citeproc main.md
```

Analizza il comando in inglese mentre stai scrivendo. Nella mia testa, traduco quanto sopra in qualcosa del tipo: "Pandoc, sii intelligente con la formattazione e crea in output un documento di Word usando il filtro di citazione sul mio file Markdown (come puoi intuire dall'estensione)." Man mano che acquisisci familiarità con i fogli di stile delle citazioni, considera la possibilità di

aggiungere all'archivio dei file .csl personalizzati per riviste del tuo campo come servizio alla comunità.

Sommario

Ora dovresti essere in grado di scrivere documenti in Markdown, creare bozze in più formati, aggiungere bibliografie e modificare facilmente gli stili di citazione. Un'ultima occhiata alla directory del progetto rivelerà un certo numero di file "sorgente": il tuo file `main.md` , il file `project.bib` , il tuo file `mla.csl` e alcune immagini. Oltre ai file di origine dovresti vedere alcuni file "destinazione" che abbiamo creato durante il tutorial: `main.docx` o `main.pdf` . La tua cartella dovrebbe apparire in questo modo:

```
Pandoc-tutorial/  
  main.md  
  project.bib  
  mla.csl  
  image.jpg  
  main.docx
```

Tratta i tuoi file sorgente come una versione autorizzata del tuo testo e i tuoi file destinazione come file "stampabili" usa e getta che puoi facilmente generare al volo con Pandoc. Tutte le revisioni dovrebbero andare in `main.md` Il file `main.docx` è lì per la pulizia e la formattazione della fase finale. Ad esempio, se la rivista richiede manoscritti a spaziatura doppia, è possibile duplicare rapidamente lo spazio in OpenOffice o Microsoft Word. Ma non dedicare troppo tempo alla formattazione. Ricorda che viene eliminata quando il tuo manoscritto va in stampa. Il tempo speso per una formattazione inutile può essere meglio utilizzato per migliorare la prosa della bozza.

Risorse utili

In caso di problemi, il posto migliore per iniziare a cercare supporto è il [sito Pandoc](#) di John MacFarlane e la [mailing list](#) collegata. Almeno due siti di tipo "Domande e risposte" possono contenere domande su Pandoc: [Stack Overflow](#) e [Digital Humanities Q & A](#). Le domande possono anche essere poste in diretta, su Freenode IRC, canale #Pandoc, frequentato da un gruppo amichevole di utenti abituali. Man mano che si impara di più su Pandoc, è anche possibile esplorare una delle sue funzioni più potenti: i [filtri](#) .

Anche se noi suggeriamo di iniziare con un semplice editor, ce ne sono molti altri (70+, secondo [questo post del blog](#)) specifici per Markdown e alternativi a MS Word, sono disponibili online e spesso gratuiti. Tra quelli stand-alone, ci piacevano [Mou](#) , [Write Monkey](#) e [Sublime Text](#). Di recente sono emerse diverse piattaforme web che forniscono ottime interfacce grafiche per la scrittura collaborativa e il tracciamento della versione tramite Markdown. Queste includono: [prose.io](#) , [Authorea](#) , [Penflip](#) , [Draft](#) e [StackEdit](#) .

Ma l'ecosistema non è limitato agli editor. [Gitit](#) e [Ikiwiki](#) supportano l'authoring in Markdown con Pandoc come parser. In questo elenco possiamo trovare una serie di strumenti che generano pagine web statiche e [veloci](#) , [Yst](#) , [Jekyll](#) , [Hakyll](#) e [script di shell bash](#) dello storico Caleb McDaniel.

Infine, intere piattaforme di pubblicazione si stanno formando attorno all'uso di Markdown. La piattaforma commerciale di Markdown [Leanpub](#) potrebbe essere un'alternativa interessante al modello editoriale tradizionale. E noi stessi stiamo sperimentando la progettazione di una rivista accademica basata su GitHub e [readthedocs.org](#) (strumenti solitamente usati per la documentazione tecnica).

1. Non ti preoccupare se non capisci ancora parte di questa terminologia! ↩
2. I file sorgente per questo documento possono essere [scaricati da GitHub](#). Usa l'opzione "raw" quando visualizzi in GitHub per vedere il Markdown di origine. Gli autori desiderano ringraziare Alex Gil e i suoi colleghi del Digital Humanities Center della Columbia e i partecipanti di openLab allo Studio nella biblioteca Butler per aver testato il codice contenuto in questo tutorial su una varietà di piattaforme. ↩
3. Guarda l'eccellente discussione di Charlie Stross su questo argomento in [Why Microsoft Word Must Die](#) . ↩
4. Non ci sono buone soluzioni per arrivare direttamente a MS Word da LaTeX. ↩
5. È una buona idea prendere l'abitudine di non usare spazi nei nomi di cartelle o file. I trattini o i caratteri di sottolineatura anziché gli spazi nei nomi dei file garantiscono una duratura compatibilità multiplatforma. ↩
6. Nota che l'estensione .bib potrebbe essere "collegata" a Zotero nel tuo sistema operativo. Ciò significa che quando fai clic su un file .bib è probabile che Zotero venga chiamato per aprirlo, mentre vogliamo aprirlo all'interno di un editor di testo. Eventualmente, potresti voler associare l'estensione .bib al tuo editor di testo. ↩
7. Grazie a [@njbart](#) per la correzione. In risposta al nostro suggerimento originale, `Some sentence that needs citation.^[@fyfe_digital_2011 argues that too.]` [Scrive](#) : "Questo non è raccomandato in quanto ti impedisce di passare facilmente dalla nota a piè di pagina agli stili autore-data. È meglio usare [corretto] (senza accento circonflesso, senza punto finale all'interno delle parentesi quadre e la punteggiatura finale della frase di testo dopo le parentesi quadre; con gli stili delle note a piè di pagina, pandoc regola automaticamente la posizione della punteggiatura finale). " ↩

Informazioni sugli autori

Dennis Tenen è un assistente di letteratura inglese e comparata presso la Columbia University.

Grant Wythoff è docente presso il Dipartimento di letteratura inglese e comparata della Columbia University.

Citazione suggerita

Dennis Tenen and Grant Wythoff, "Sustainable Authorship in Plain Text using Pandoc and Markdown," *The Programming Historian* 3 (2014),

<https://programminghistorian.org/en/lessons/sustainable-authorship-in-plain-text-using-pandoc-and-markdown>.