

Usando Pandoc

Sergio Alvariño salvari@gmail.com

abril-2016

Resumen

Una chuleta para usar Pandoc, copiado descaradamente de varios sitios de internet.
Solo para referencia rápida y personal.

Índice general

1	¿Cómo usar esto?	2
1.1	Muy rápido	2
1.2	Más detalles	2
2	¿Qué es Pandoc?	3
3	¿Qué necesitas tener instalado?	3
4	Chuleterio de Pandoc	3
4.1	Backslash Escapes	3
4.2	Bloque de título	3
4.3	Incrustar TeX y HTML	4
4.4	Párrafos y retornos de línea	4
4.5	Itálica, negrita, superescrito, subescrito, tachado	4
4.6	TeX matemático o código incrustado en línea	5
4.7	Enlaces e imágenes	5
4.8	Notas al pie de página	5
4.9	Citas	6
4.10	Encabezados	6
4.11	Listas	6
4.12	Blockquotes	7
4.13	Tablas	7
4.14	Bloques de código	8
4.15	Lineas horizontales	8
4.16	Bloques verbatim	8

1 ¿Cómo usar esto?

1.1 Muy rápido

Clona el repo en un directorio :

```
git clone https://bitbucket.org/salvari/pandoc_basico
```

Renombra el directorio:

```
mv pandoc_basico miProyecto
```

Elimina la info de git

```
rm -rf miProyecto/.git
```

Edita el fichero miProyecto/src/documento.md con tu editor de texto favorito.

Ejecuta:

make Para generar todos los ficheros de salida y el fichero README.md (equivale a *make all*)

make clean Para borrar todos los ficheros de salida

make reset Equivale a *make clean all*

1.2 Más detalles

El makefile está preparado para procesar **todos** los ficheros con extensión *.md* que haya en el directorio *src*. Esto permite escribir documentos largos y dividirlos en secciones, por ejemplo podríamos tener los siguientes documentos en el directorio *src*

```
00_Comienzo.md
10_Capitulo_01.md
20_Capitulo_02.md
30_Conclusion.md
40_apendices.md
```

Al ejecutar **make** nos crearía **un solo documento de salida** concatenando todos los ficheros. El orden en que los concatena es el orden en el que aparecen al hacer un *ls* por eso se nombran con una numeración al principio que permita ordenarlos a gusto del autor.

Si quieres cambiar el nombre del fichero de salida (*documento*) tendrás que editar el makefile y cambiar la línea:

```
target := documento
```

Otras líneas que puedes tocar en el makefile son las que especifican el idioma y los tipos de letra usados.

2 ¿Qué es Pandoc?

Como explican en [<http://pandoc.org>], Pandoc es una librería en Haskell para hacer conversión de documentos de un formato markup a otro. Y también es una herramienta de terminal de comandos que usa esa librería.

Lo que nos permite Pandoc a la hora de documentar un proyecto es mantener la documentación en un formato abierto y sencillo (markdown) y generar salidas en distintos formatos (pdf, mediawiki, epub, html, etc) con un simple comando.

3 ¿Qué necesitas tener instalado?

- Pandoc
- make
- git (no es imprescindible pero muy recomendable)
- Las plantillas de Pandoc (o *templates*)
- Un buen editor de texto

4 Chuletario de Pandoc

4.1 Backslash Escapes

Salvo que estemos dentro de un bloque de código o de "código en línea", **cualquier carácter de puntuación o espacio** precedido de contrabarra se tratará de forma literal, incluso si ese carácter normalmente indique algún formato.

4.2 Bloque de título

Es una forma rápida de indicar el título el autor o autores y la fecha. Tiene que ir al principio del documento

```
% título
% autor(es) (separados por :)
% fecha
```

Alternativamente se puede usar otro estilo para el bloque de título, mucho más completo, en formato [YAML](#), especificando variables. No puede usarse simultáneamente con el anterior, hay que escoger entre los dos estilos.

Se pueden especificar todo tipo de variables ¹.

```
---
title: Título
author:
- Autor Uno <autor.uno@correo.com>
- Otro autor <otroautor@correo.com>
tags: [nothing, nothingness]
date: enero-2016
lang: es-ES
abstract: |
  Este es el resumen.

  Con dos párrafos.
...
---
```

4.3 Incrustar TeX y HTML

- Los comandos TeX se pasan de forma transparente al Markdown, y afectan solo a la salida de LaTeX y ConTeXt; en el resto de casos se borran
- El código HTML pasará a la salida sin cambios, pero el Markdown dentro de los bloques HTML se procesa como Markdown

4.4 Párrafos y retornos de línea

- Un párrafo es una o más líneas de texto separadas por una línea en blanco del resto
- Una línea que termina con dos espacios, o una línea que termina con un fin de línea escapado (contrabarra seguida de retorno de línea) indica un cambio de línea manual

4.5 Itálica, negrita, superescrito, subescrito, tachado

Itálica and ****negrita**** se indican con asteriscos.

¹Ojo por que en el makefile propuesto se especifica el lenguaje, así que la variable del bloque de título no va a tener efecto en este caso.

Para `~tachar~` texto usa tildes dobles.

Superscrito se indica así: 2^{ndo} .

Subescrito con tildes simples, así: H_{2-0} .

Los espacios en el superscrito y el subescrito tienen que ir escapados, p.ej., $H_{\text{esto}} \text{ es } \backslash \text{un} \backslash \text{subescrito}$.

4.6 TeX matemático o código incrustado en línea

El TeX matemático va entre signos\$: $\$2 + 2\$$.

El código en línea va entre comillas invertidas: ``echo 'hello'``

4.7 Enlaces e imágenes

```
<http://example.com>
<foo@bar.com>
[inline link](http://example.com "Title")
![inline image](/path/to/image, "alt text")
```

```
[reference link][id]
[implicit reference link][]
![reference image][id2]
```

```
[id]: http://example.com "Title"
[implicit reference link]: http://example.com
[id2]: /path/to/image "alt text"
```

4.8 Notas al pie de página

Las notas en línea son como
esta.^[Nótese que las notas en línea no pueden tener más de un párrafo.]
Las notas de referencia son como esta.^[^id]

[^id]: Las notas de referencia pueden contener varios párrafos.

Los parámetros a continuación deben estar identados.

4.9 Citas

Blah blah [see @doe99, pp. 33-35; also @smith04, ch. 1].

Blah blah [@doe99, pp. 33-35, 38-39 and *passim*].

Blah blah [@smith04; @doe99].

Smith says blah [-@smith04].

@smith04 says blah.

@smith04 [p. 33] says blah.

4.10 Encabezados

Encabezado 1

=====

Encabezado 2

Encabezado 1

Encabezado 2

Las almohadillas de cierre # son opcionales. Es necesario añadir una línea en blanco antes y después de cada cabecera.

4.11 Listas

4.11.0.1 Listas Ordenadas

1. example
2. example

- A) example
- B) example

4.11.0.2 Listas desordenadas

Los items de la lista deben ir marcados con '*', '+', or '-'.

- + example
- example
- * example

Las listas se pueden anidar de la forma usual:

- + example
 - + example
- + example

4.11.0.3 Listas de definición

Term 1

: Definition 1

Term 2

: Definition 2
Second paragraph of definition 2.

4.12 Blockquotes

- > blockquote
- >> nested blockquote

Es necesario añadir líneas en blanco antes y después de los bloques-cita.

4.13 Tablas

Right	Left	Center	Default
12	12	12	12
123	123	123	123
1	1	1	1

Table: Demonstration of simple table syntax.

(Para tablas más complejas consulta la [documentación de Pandoc.](#))

4.14 Bloques de código

Los bloques de código empiezan con tres o más tildes; y acaban por lo menos con el mismo número de tildes:

```
~~~~~  
{code here}  
~~~~~
```

Opcionalmente, se puede especificar el lenguaje que corresponde al bloque de código:

```
~~~~~ {haskell .numberLines}  
qsort [] = []  
qsort (x:xs) = qsort (filter (< x) xs) ++ [x] ++  
               qsort (filter (>= x) xs)  
~~~~~
```

4.15 Líneas horizontales

3 o más guiones o asteriscos en una línea (se permiten espacios intercalados)

```
---  
* * *  
- - -
```

4.16 Bloques verbatim

Todo el texto indentado cuatro espacios

Ejemplo Esto es un bloque verbatim y por ejemplo **esto** aparece tal cual y no en itálica.

5 En que me he basado (o copiado si lo prefieres)

- En la [guía de usuario de Pandoc](#) Importante leerla para sacarle todo el jugo a esta herramienta
- En la [chuleta de Pandoc](#) de [David Sanson](#), perfecta para referencia rápida
- Para hacer el makefile me he leído varios tutoriales y copiado descaradamente de varios sitios que olvidé apuntar (lo siento)