Markdown

Miguel Ortuño
Escuela de Ingeniería de Fuenlabrada
Universidad Rey Juan Carlos

Septiembre de 2023

Introducción

Markdown es un lenguaje de marcado ligero

- Creado por John Gruber y Aaron Swartz en 2004
- Sintaxis muy sencilla, fácil de escribir a mano con un editor de texto plano
- No intenta reemplazar a lenguajes de marcado más potentes como HTML, Latex, PostScript, DocBook, etc
- Muy popular en la actualidad para entornos donde se desea una cierta *maquetación* del texto, no muy exigente: foros, blogs, mensajería instantánea, documentación de código fuente, ficheros *readme*, documentos internos, manuales de usuario *online*, etc

Herramientas para Markdown

- El texto en Markdown se puede leer *tal cual* o se puede usar para generar otros formatos como html, pdf, rtf (para Microsoft Word), odt (para LibreOffice), entre otros. Empleando para ello
  - Grip, una herramienta muy sencilla para generar html desde markdown
    https://github.com/joeyespo/grip
  - Pandoc, una herramienta muy potente que convierte documentos desde y hasta múltiples formatos (Markdown, Html, epub, docbook, LaTex, RTF, ODT, ...)
    https://pandoc.org/
- También hay herramientas que facilitan su edición
  - Aplicaciones completas de escritorio como Abricotine, Remarkable o ReText
  - Plugins para editores como Atom, Vim o Visual Studio Code
  - Editores *online* como GitBook

Variantes de Markdown

- La definición inicial del lenguaje era bastante informal, con muchas imprecisiones. Herramientas muy variadas lo fueron incorporando, cada una con interpretaciones y extensiones ligeramente distintas en los aspectos no básicos
- Hay por tanto muchas variantes de Markdown. Tal vez el intento de normalización más extendido es GFM (*GitHub Flavored Markdown*, año 2017)

Párrafos

- Los saltos de línea *ordinarios* se ignoran. En el momento del *rendering*, la línea se rompe por donde donde corresponda
- Para hacer un párrafo (punto y aparte) es necesario escribir al menos una línea en blanco
- Para forzar un salto de línea, se pueden escribir dos espacios antes del salto de línea
  - Esto es problemático porque es invisible. Muchas herramientas admiten escribirlo en html: `<br>`

Secciones en el texto

```
# Encabezado nivel 1  (Sección)
## Encabezado nivel 2 (Subsección)
(...)
###### Encabezado nivel 6 (Sub-sub-sub-sub-sub-sección)
```

Es necesario dejar un espacio entre la almohadilla y el título

Cursiva, negrita, enlaces

Cursiva

- Una barra baja al comienzo y final del texto a resaltar
  `_ejemplo cursiva_`
- O bien un asterisco al comienzo y al final del texto a resaltar
  `*ejemplo cursiva*`

Negrita

- Dos barras bajas al comienzo y final del texto a resaltar
  `__ejemplo negrita__`
- O bien dos astericos al comienzo y al final del texto a resaltar
  `**ejemplo negrita**`

Enlaces

- Descripción del enlace, entre corchetes, seguido de la URL, entre paréntesis

  ```
  [Universidad Rey Juan Carlos](https://urjc.es)
  ```

Imágenes (1)

Se pueden añadir imágenes con la misma sintaxis que los enlaces, pero añadiendo una admiración. Esta es una extensión GFM que no soportan todas las herramientas

- Fichero en una dirección web

  ```
  ![Logo de la URJC](https://gsyc.urjc.es/~mortuno/urjc.gif)
  ```

- Fichero en un trayecto absoluto de mi ordenador

  ```
  ![Pantallazo 1](/home/jperez/fotos/pantallazo01.png)
  ```

  Cuidado: incluir el trayecto absoluto de una cuenta de usuario casi siempre es un error: dejará de funcionar cuando el fichero lo abra otro usuario

Imágenes (2)

- Fichero en el mismo directorio que el documento actual

  ```
  ![Pantallazo 2](pantallazo02.png)
  ```

- Fichero en un subdirectorio del directorio actual

  ```
  ![Pantallazo 3](images/pantallazo03.png)
  ```

  Posiblemente esto es lo más recomendable: escribir el fichero en un subdirectorio llamado *images* dentro del directorio donde está el documento. Observa que NO es

  ```
  ![Pantallazo 3](/images/pantallazo01.png)
  ```

  La barra antes del directorio indicaría un trayecto absoluto

Listas sin ordenar

Se crean con un guión seguido de un espacio

```
- Sota
   - Sota de espadas
   - Sota de oros
- Caballo
  La figura del caballero, generalmente llamado
*caballo* es una peculiaridad de la baraja española
que sustituye a la figura de la reina que aparece
en la mayoría de las restantes barajas. Son cuatro:
   - Caballo de oros
   - ...
```

- Para hacer sublistas, añadimos indentación. Basta con hacerlo en la primera línea del párrafo. También usamos indentación para que el párrafo pertenezca a un elemento de la lista y no sea un párrafo distinto. Es necesario escribir el mismo número de espacios que en la línea que abrió ese nivel
- En vez de guiones se pueden usar asteriscos

Listas ordenadas

Se crean como las listas no ordenadas, pero con un número, un punto y un espacio

```
1. Análisis
1. Diseño
1. Codificación
1. Prueba
```

El número que escribamos es irrelevante. Puede ser una secuencia correcta como 1,2,3, todo unos, números fuera de secuencia...

Código

- Código dentro de una línea
  Se escribe entre comillas invertidas. (la comilla a la derecha de la letra P). Ejemplo:
  ```
  Ejecuta la orden `ls -l`
  ```

- Bloques de código
  - Apertura:Tres comillas invertidas. Se puede añadir el nombre del lenguaje
  - Cierre: Tres comillas invertidas

  Propio de Github Markdown, no siempre disponible

  ````
  ```python
  #!/usr/bin/env python3

  def main():
      return

  if __name__ == "__main__":
      main()
  ```
  ````

Tablas

En GFM también se pueden crear tablas

```
Hora   | Lunes | Martes | Miércoles
---|-------|--------|------
9:00 | Tal | Tal | Cual
11:00| Esto | Lo otro | Más allá
```

Generación de HTML

Fichero Markdown de ejemplo

- https://gsyc.urjc.es/~mortuno/md/ejemplo.md

Conversión a HTML con pandoc, sin plantilla de estilo

```
pandoc -s  ejemplo.md -o ejemplo_no_css.html
```

Resultado:

- https://gsyc.urjc.es/~mortuno/md/ejemplo_no_css.html

Es usable pero bastante *feo*. Mejora mucho con la hoja de estilos de pandoc

```
pandoc -s -c pandoc.css ejemplo.md -o ejemplo_css_01.html
```

Resultado:

- https://gsyc.urjc.es/~mortuno/md/ejemplo_css_01.html

En Internet encontraremos muchas otras hojas de estilo css. Por ejemplo esta:
https://gist.github.com/killercup/5917178
Resultado:

- https://gsyc.urjc.es/~mortuno/md/ejemplo_css_02.html

Lo más sencillo es dejar los ficheros css en el mismo directorio que el fichero md. Puedes obtenerlos desde stos enlaces:

- css de pandoc
- css de killercup

Otras opciones de pandoc

- Añadiendo a pandoc la opción `--toc` se crea una tabla de contenidos al principio el documento (un índice)
- Pandoc dará un *warning* porque nuestro documento no tiene título. En Markdown no hay forma estándar de incluir metainformación al documento, pero muchas herramientas soportan el formato de YAML para metadatos, justo al pricipio del documento

```
---
title: Introducción al formato Markdown
---
```

Ejemplo:
fpi_practica_08.md
fpi_practica_08_css_01.html
fpi_practica_08_css_02.html

Este script facilita el uso de pandoc

```
#!/bin/bash

# Leave uncomented the style sheet that you prefer
#TEMPLATE=md.css  # https://gist.github.com/killercup/5917178
#TEMPLATE=pandoc.css
TEMPLATE=https://gsyc.urjc.es/~mortuno/pandoc.css

if test $# -eq 0 || test $# -gt 1
then
    echo "Es necesario indicar un argumento (y solo uno)" >&2
    exit
fi

filename=$(basename "$1")
extension="${filename##*.}"
filename="${filename%.*}"

pandoc -s  --toc -c ${TEMPLATE} $1 -o ${filename}.html
```

- Recibe como argumento un fichero *.md* (o *.tex*, entre otros formatos)
- Genera una versión del documento en HTML
- Puedes descargarlo aquí: http://gsyc.urjc.es/~mortuno/mypandoc

Este script usa pandoc para convertir un fichero desde markdown hasta markdown. El formato final no cambia, pero resulta muy útil, porque *limpia* el fichero: líneas homogéneas, estilo homogéneo, etc

```
#!/bin/bash

# -s  standalone document
# -S typographically smart


if test $# -eq 0 || test $# -gt 1
then
    echo "Es necesario indicar un argumento (y solo uno)" >&2
    exit
fi

filename=$(basename "$1")
extension="${filename##*.}"
filename="${filename%.*}"

if test $extension != md
then
    echo "Es necesario que la extensión sea .md" >&2
    exit
fi

tmp_name=/tmp/${filename}.$$.md
pandoc -s  $1 -o ${tmp_name}
mv ${tmp_name} $1
```

Puedes descargarlo aquí: http://gsyc.urjc.es/~mortuno/clean_md

grip

Una herramienta alternativa a pandoc para convertir ficheros markdown en ficheros HTML es grip

- En ocasiones genera HTML de mayor calidad
- Actualmente no permite insertar css, pero podemos añadirlo nosotros fácilmente, basta añadir un elemento `<style>` o mejor un `<link>`

Para instalar grip, ejecutamos (con privilegios de administrador)

```
sudo apt install grip
```

Para usarlo:

```
grip FICHERO_ENTRADA.md  --export FICHERO_SALIDA.html
```

Referencias

https://guides.github.com/features/mastering-markdown/
https://www.markdownguide.org/basic-syntax/