

# Acordeón de Markdown

## Acordeón de Ejemplos de Markdown

Este es un archivo markdown de ejemplo para ayudarte a escribir Markdown rápidamente :)

Si usas el fabuloso [Sublime Text 2/3 editor](#) junto con el [Markdown Preview plugin](#), abre tu Paleta ST2 con `CMD+ +P` luego elige `Markdown Preview in browser` para ver el resultado en tu navegador.

### Texto básico

esto es *italic* y esto es **bold** . otro *italic* y otro **bold**

esto es texto **importante** y signos de porcentaje : % y %

Esto es un párrafo con una nota a pie de página (sólo analizador sintáctico incorporado). <sup>1</sup>

Inserte [ TOC ] sin espacios para generar una tabla de contenidos (sólo analizadores incorporados).

### Sangría

Aquí hay algo de texto sangrado > aún más sangrado

---

<sup>1</sup>Este es el texto de la nota.

## Títulos

**Título grande (h1)**

**Título medio (h2)**

**Título más pequeño (h3)**

**y así sucesivamente (hX)**

**y así sucesivamente (hX)**

y así sucesivamente (hX)

## Listas de ejemplo (1)

- las viñetas pueden ser -, + o \*.
- lista de viñetas 1
- lista de viñetas 2
  - subelemento 1
  - subelemento 2con texto sangrado en el interior
- lista 3
- lista 4 ## bullet list 5

## Enlaces

Este es un [ejemplo de enlace en línea](#) y [otro con título](#).

Los enlaces también pueden basarse en referencias: [referencia 1](#) o [referencia 2 con título](#).

Las referencias suelen colocarse al final del documento

## Imágenes

Una imagen de ejemplo :

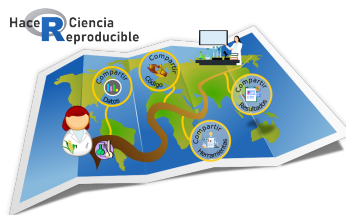


Figure 1: por una ciencia reproducible

Como enlaces, las imágenes también pueden utilizar referencias en lugar de enlaces en línea :

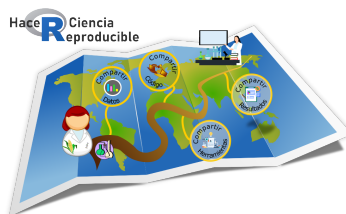


Figure 2: logo revolunet

## Código

Es bastante fácil mostrar código en archivos markdown.

Las marcas pueden utilizarse para **resaltar** algunas palabras.

Además, cualquier bloque con sangría se considera un bloque de código. If `enable_highlight` is `true`, syntax highlighting will be included (for the builtin parser - the github parser does this automatically).

```
<script>
  document.location = 'http://lmgtyf.com/?q=markdown+cheat+sheet';
</script>
```

## Fórmulas Matemáticas

Las matemáticas pueden mostrarse en el navegador usando MathJax o Katex. La función puede activarse configurando correctamente los campos de configuración "js", "css" y "markdown\_extensions". Esto permite incluir matemáticas en línea  $\frac{\pi}{2}$  o sólo  $\pi$ .

Alternativamente, las matemáticas se pueden escribir en su propia línea:

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

$$\int_0^1 f(t) dt$$

$$\sum_j \gamma_j^2 / d_j$$

## GitHub Flavored Markdown

If you use the Github parser, you can use some of [Github Flavored Markdown](#) syntax :

- User/Project@SHA: [revolunet/sublimetext-markdown-preview@7da61bade468b5019869d11000307e07e0](#)
- Usuario/Proyecto#Edición: [revolunet/sublimetext-markdown-preview#1](#)
- Usuario : @revolunet

Algo de código Python :

```
importar random

class JuegoDeCartas(object):
    """ una clase python de ejemplo """
    NB_CARDAS = 32
    def __init__(self, cards=5):
        self.cards = random.sample(range(self.NB_CARDAS), 5)
        print 'listo para jugar'
```

Algo de código Javascript :

```

var config = {
    duración 5,
    comentario: 'WTF'
}
// callbacks beauty un action
async_call('/ruta/a/api', function(json) {
    otra_llamada(json, function(resultado2) {
        otra_otra_llamada(resultado2, function(resultado3) {
            another_another_another_call(result3, function(result4) {
                alert('Y si todo ha ido bien, ya tengo mi resultado :)');
            });
        });
    });
});
})

```

El Github Markdown también trae algunos [bonito soporte Emoji](#) : +1: :heart: :beer:

## Parsers y Extensiones

Markdown Preview viene con **Python-Markdown** precargado.

### *Python-Markdown*

El [Python-Markdown Parser][] proporciona soporte para varias extensiones.

Parser [Python-Markdown]: <https://github.com/Python-Markdown/markdown>

### Extensiones extra

- abbr – [Abreviaturas](#)
- attr\_list – [Listas de Atributos](#)
- def\_list – [Listas de definiciones](#)
- fenced\_code – [Bloques de código delimitados](#)
- footnotes – [Notas a pie de página](#)
- tables – [Tablas](#)
- smart\_strong – [Smart Strong](#)

Puede activarlas todas a la vez utilizando la palabra clave **extra**.

```
extensiones: [ 'extra' ]
```

Si quieres todos los extras más la extensión `toc`, su configuración se vería así:

```
{
    ...
    analizador sintáctico: 'markdown',
    extensiones: ['extra', 'toc'],
    ...
}
```

## Otras extensiones

También hay algunas extensiones que no están incluidas en Markdown Extra pero vienen en la biblioteca estándar Python-Markdown.

- `code-hilite` – [CodeHilite](#)
- `header-id` – [HeaderId](#)
- `meta_data` – `[Meta-Data][]`
- `nl2br` – [Nueva línea a romper](#)
- `sane_lists` – [Listas Sanas](#)
- `smarty` – [Smarty](#)
- `toc` – `[Tabla de contenidos][]`
- `wikilinks` – [Enlaces Wiki](#)

## Extensiones de terceros

*Python-Markdown* está diseñado para ser extendido.

Algunas incluidas son:

- `delete` – soporte de borrado estilo github mediante `~~word~~`.
- `githubemoji` – soporte para emoji de github
- `tasklist` – listas de tareas estilo github
- `magiclink` – conversión automática de enlaces `http|ftp` al estilo github
- `headeranchor` – enlaces de anclaje de cabecera al estilo github
- `github` – Añade las extensiones anteriores de una sola vez
- `b64` – convierte e incrusta imágenes locales a base64. Se configura añadiendo `b64(base_path=${BASE_PATH})`.

También hay una serie de otros disponibles:

Simplemente bifurca este repositorio y añade tus extensiones dentro de la carpeta `.../Packages/Markdown Preview/markdown/extensions/`.

Echa un vistazo a la lista de [3rd Party extensions](#).

## Extensiones por defecto

Las extensiones por defecto son:

- `footnotes` – [Notas a pie de página](#)
- `toc` – [Tabla de contenidos]
- `fenced_code` – [Bloques de código delimitados](#)
- `tables` – [Tablas](#)

Utilice la palabra clave `default`, para seleccionarlos todos. Si quiere todos los predeterminados más la extensión `definition_lists`, su configuración se vería así:

```
{  
  ...  
  analizador sintáctico: 'markdown',  
  extensiones: ['default', 'definition_lists'],  
  ...  
}
```

## Ejemplos

### Tablas

La extensión `tables` del analizador sintáctico *Python-Markdown* está activada por defecto, pero actualmente **no** está disponible en *Markdown2*.

La sintaxis fue adoptada del [php markdown project](#), y también se usa en markdown con sabor a github.

Año	Temperatura (baja)	Temperatura (alta)
1900	-10	25
1910	-15	30
1920	-10	32

### Tablas Wiki

Si estás usando *Markdown2* con el extra `wiki-tables` activado deberías ver la siguiente tabla:

*Año* || *Temperatura (baja)* | *Temperatura (alta)* ||

```
|| 1900 || -10 || 25 ||  
|| 1910 || -15 || 30 ||  
|| 1920 || -10 || 32 ||
```

## Listas de definiciones

Este ejemplo requiere la extensión `def_list` de *Python Markdown*.

**Apple** Fruto pomáceo de plantas del género *Malus* de la familia Rosaceae.

**Naranja** Fruto de un árbol perenne del género *Citrus*.

## Acerca de

Este plugin y este archivo de ejemplo es orgullosamente traído a usted por el [equipo revolunet](#).  
La traducción la hizo [Miguel Equihua](#)