

[文件](#) [資源](#) [標準](#) [回報錯誤](#)

# Markdown文件



**NOTE:** This is Traditional Chinese Edition Document of Markdown Syntax. If you are seeking for English Edition Document. Please refer to [Markdown: Syntax](#).

---

## Markdown: Syntax

- [概述](#)
  - [哲學](#)
  - [行內 HTML](#)
  - [特殊字元自動轉換](#)
- [區塊元素](#)
  - [段落和換行](#)
  - [標題](#)
  - [區塊引言](#)
  - [清單](#)
  - [程式碼區塊](#)
  - [分隔線](#)
- [區段元素](#)
  - [連結](#)
  - [強調](#)
  - [程式碼](#)
  - [圖片](#)
- [其他](#)
  - [跳脫字元](#)
  - [自動連結](#)
- [感謝](#)

注意：這份文件是用Markdown寫的，你可以[看看它的原始檔](#)。

---

## 概述

### 哲學

Markdown的目標是實現「易讀易寫」。

不過最需要強調的便是它的可讀性。一份使用Markdown格式撰寫的文件應該可以直接以純文字發佈，並且看起來不會像是由許多標籤或是格式指令所構成。Markdown語法受到一些既有text-to-HTML格式的影響，包括Setext、atx、Textile、reStructuredText、Grutatext和EtText，然而最大靈感來源其實是純文字的電子郵件格式。

因此Markdown的語法全由標點符號所組成，並經過嚴謹慎選，是為了讓它們看起來就像所要表達的意思。像是在文字兩旁加上星號，看起來就像\*強調\*。Markdown的清單看起來，嗯，就是清單。假如你有使用過電子郵件，區塊引言看起來就真的像是引用一段文字。

## 行內HTML

Markdown的語法有個主要的目的：用來作為一種網路內容的寫作語言。

Markdown不是要來取代HTML，甚至也沒有要和它相似，它的語法種類不多，只和HTML的一部分有關係，重點不是要創造一種更容易寫作HTML文件的語法，我認為HTML已經很容易寫了，Markdow的重點在於，它能讓文件更容易閱讀、編寫。HTML是一種發佈的格式，Markdown是一種編寫的格式，因此，Markdown的格式語法只涵蓋純文字可以涵蓋的範圍。

不在Markdown涵蓋範圍之外的標籤，都可以直接在文件裡面用HTML撰寫。不需要額外標註這是HTML或是Markdown；只要直接加標籤就可以了。

只有區塊元素——比如<div>、<table>、<pre>、<p>等標籤，必需在前後加上空行，以利與內容區隔。而且這些（元素）的開始與結尾標籤，不可以用tab或是空白來縮排。Markdown的產生器有智慧型判斷，可以避免在區塊標籤前後加上沒有必要的<p>標籤。

舉例來說，在Markdown文件裡加上一段HTML表格：

```
This is a regular paragraph.

<table>
  <tr>
    <td>Foo</td>
  </tr>
</table>

This is another regular paragraph.
```

請注意，Markdown語法在HTML區塊標籤中將不會被進行處理。例如，你無法在HTML區塊內使用Markdown形式的\*強調\*。

HTML的區段標籤如<span>、<cite>、<del>則不受限制，可以在Markdown的段落、清單或是標題裡任意使用。依照個人習慣，甚至可以不用Markdown格式，而採用HTML標籤來格式化。舉例說明：如果比較喜歡HTML的<a>或<img>標籤，可以直接使用這些標籤，而不用Markdown提供的連結或是影像標示語法。

HTML區段標籤和區塊標籤不同，在區段標籤的範圍內，Markdown的語法是有效的。

## 特殊字元自動轉換

在HTML文件中，有兩個字元需要特殊處理：<和&。<符號用於起始標籤&符號則用於標記HTML實體，如果你只是想要使用這些符號，你必須要使用實體的形式，像是<lt;和<amp;。

`&` 符號其實很容易讓寫作網路文件的人感到困擾，如果你要打「AT&T」，你必須要寫成「AT&amp;T」，還得轉換網址內的 `&` 符號，如果你要連結到：

```
http://images.google.com/images?num=30&q=larry+bird
```

你必須要把網址轉成：

```
http://images.google.com/images?num=30&amp;q=larry+bird
```

才能放到連結標籤的 `href` 屬性裡。不用說也知道這很容易忘記，這也可能是HTML標準檢查所檢查到的錯誤中，數量最多的。

Markdown允許你直接使用這些符號，但是你要小心跳脫字元的使用，如果你是在HTML實體中使用`&`符號的話，它不會被轉換，而在其他情形下，它則會被轉換成`&amp;`。所以你如果要在文件中插入一個著作權的符號，你可以這樣寫：

```
&copy;
```

Markdown將不會對這段文字做修改，但是如果你這樣寫：

```
AT&T
```

Markdown就會將它轉為：

```
AT&amp;T
```

類似的狀況也會發生在`<`符號上，因為Markdown支援行內 HTML，如果你是使用`<`符號作為HTML標籤使用，那Markdown也不會對它做任何轉換，但是如果你是寫：

```
4 < 5
```

Markdown將會把它轉換為：

```
4 &lt; 5
```

不過需要注意的是，code範圍內，不論是行內還是區塊，`<`和`&`兩個符號都一定會被轉換成HTML實體，這項特性讓你可以很容易地用Markdown寫HTML code（和HTML相對而言。在HTML語法中，你要把所有的`<`和`&`都轉換為 HTML 實體，才能在HTML文件裡面寫出HTML code。）

---

## 區塊元素

### 段落和換行

一個段落是由一個以上相連接的行句組成，而一個以上的空行則會切分出不同的段落（空行的定義是顯示上看起來像是空行，便會被視為空行。比方說，若某一行只包含空白和tab，則該行也會被視為空行），一般的段落不需要用空白或斷行縮排。

「一個以上相連接的行句組成」這句話其實暗示了Markdown允許段落內的強迫斷行，這個特性和其他大部分的text-to-HTML格式不一樣（包括 MovableType的「Convert Line Breaks」選項），其他的格式會把每個斷行

都轉成 `<br />` 標籤。

如果你真的想要插入 `<br />` 標籤的話，在行尾加上兩個以上的空白，然後按enter。

是的，這確實需要花比較多功夫來插入 `<br />`，但是「每個換行都轉換為 `<br />`」的方法在Markdown中並不適合，Markdown中email式的**區塊引言**和多段落的**清單**在使用換行來排版的時候，不但更好用，還更好閱讀。

## 標題

Markdown支援兩種標題的語法，**Setext**和**atx**形式。

Setext形式是用底線的形式，利用 `=`（最高階標題）和 `-`（第二階標題），例如：

```
This is an H1
=====

This is an H2
-----
```

任何數量的 `=` 和 `-` 都可以有效果。

Atx形式則是在行首插入1到6個 `#`，各對應到標題1到6階，例如：

```
# This is an H1

## This is an H2

##### This is an H6
```

你可以選擇性地「關閉」atx樣式的標題，這純粹只是美觀用的，若是覺得這樣看起來比較舒適，你就可以在行尾加上 `#`，而行尾的 `#` 數量也不用和開頭一樣（行首的井字數量決定標題的階數）：

```
# This is an H1 #

## This is an H2 ##

### This is an H3 #####
```

## 區塊引言

Markdown使用email形式的區塊引言，如果你很熟悉如何在email信件中引言，你就知道怎麼在Markdown文件中建立一個區塊引言，那會看起來像是你強迫斷行，然後在每行的最前面加上 `>`：

```
> This is a blockquote with two paragraphs. Lorem ipsum dolor sit amet,
> consectetur adipiscing elit. Aliquam hendrerit mi posuere lectus.
> Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus.
>
> Donec sit amet nisl. Aliquam semper ipsum sit amet velit. Suspendisse
> id sem consectetur libero luctus adipiscing.
```

Markdown也允許你只在整個段落的第一行最前面加上 `>`：

```
> This is a blockquote with two paragraphs. Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Aliquam hendrerit mi posuere lectus.
Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus.
```

```
> Donec sit amet nisl. Aliquam semper ipsum sit amet velit. Suspendisse  
id sem consectetur libero luctus adipiscing.
```

區塊引言可以有階層（例如：引言內的引言），只要根據層數加上不同數量的 `>`：

```
> This is the first level of quoting.  
>  
> > This is nested blockquote.  
>  
> Back to the first level.
```

引言的區塊內也可以使用其他的Markdown語法，包括標題、清單、程式碼區塊等：

```
> ## This is a header.  
>  
> 1. This is the first list item.  
> 2. This is the second list item.  
>  
> Here's some example code:  
>  
>     return shell_exec("echo $input | $markdown_script");
```

任何標準的文字編輯器都能簡單地建立email樣式的引言，例如BBEdit，你可以選取文字後然後從選單中選擇增加引言階層。

## 清單

Markdown支援有序清單和無序清單。

無序清單使用星號、加號或是減號作為清單標記：

```
* Red  
* Green  
* Blue
```

等同於：

```
+ Red  
+ Green  
+ Blue
```

也等同於：

```
- Red  
- Green  
- Blue
```

有序清單則使用數字接著一個英文句點：

```
1. Bird  
2. McHale  
3. Parish
```

很重要的一點是，你在清單標記上使用的數字並不會影響輸出的HTML結果，上面的清單所產生的HTML標記為：

```
<ol>
<li>Bird</li>
<li>McHale</li>
<li>Parish</li>
</ol>
```

如果你的清單標記寫成：

```
1. Bird
1. McHale
1. Parish
```

或甚至是：

```
3. Bird
1. McHale
8. Parish
```

你都會得到完全相同的HTML輸出。重點在於，你可以讓Markdown文件的清單數字和輸出的結果相同，或是你懶一點，你可以完全不用在意數字的正確性。

如果你使用懶惰的寫法，建議第一個項目最好還是從「1.」開始，因為Markdown未來可能會支援有序清單的start屬性。

清單項目標記通常是放在最左邊，但是其實也可以縮排，最多三個空白，項目標記後面則一定要接著至少一個空白或tab。

要讓清單看起來更漂亮，你可以把內容用固定的縮排整理好：

```
*   Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Aliquam hendrerit mi posuere lectus. Vestibulum enim wisi,
    viverra nec, fringilla in, laoreet vitae, risus.
*   Donec sit amet nisl. Aliquam semper ipsum sit amet velit.
    Suspendisse id sem consectetur libero luctus adipiscing.
```

但是如果你很懶，那也不一定需要：

```
*   Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Aliquam hendrerit mi posuere lectus. Vestibulum enim wisi,
viverra nec, fringilla in, laoreet vitae, risus.
*   Donec sit amet nisl. Aliquam semper ipsum sit amet velit.
Suspendisse id sem consectetur libero luctus adipiscing.
```

如果清單項目間用空行分開，Markdown會把項目的內容在輸出時用<p>標籤包起來，舉例來說：

```
*   Bird
*   Magic
```

會被轉換為：

```
<ul>
<li>Bird</li>
<li>Magic</li>
</ul>
```

但是這個：

```
* Bird

* Magic
```

會被轉換為：

```
<ul>
<li><p>Bird</p></li>
<li><p>Magic</p></li>
</ul>
```

清單項目可以包含多個段落，每個項目下的段落都必須縮排4個空白或是一個tab：

1. This is a list item with two paragraphs. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam hendrerit mi posuere lectus.  
  
Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus. Donec sit amet nisl. Aliquam semper ipsum sit amet velit.
2. Suspendisse id sem consectetur libero luctus adipiscing.

如果你每行都有縮排，看起來會好看很多，當然，再次地，如果你很懶惰，Markdown也允許：

```
* This is a list item with two paragraphs.

    This is the second paragraph in the list item. You're
    only required to indent the first line. Lorem ipsum dolor
    sit amet, consectetur adipiscing elit.

* Another item in the same list.
```

如果要在清單項目內放進引言，那>就需要縮排：

```
* A list item with a blockquote:

    > This is a blockquote
    > inside a list item.
```

如果要放程式碼區塊的話，該區塊就需要縮排兩次，也就是8個空白或是兩個tab：

```
* A list item with a code block:

    <code goes here>
```

當然，項目清單很可能會不小心產生，像是下面這樣的寫法：

```
1986. What a great season.
```

換句話說，也就是在行首出現數字一句點一空白，要避免這樣的狀況，你可以在句點前面加上反斜線。

```
1986\. What a great season.
```

## 程式碼區塊

和程式相關的寫作或是標籤語言原始碼通常會有已經排版好的程式碼區塊，通常這些區塊我們並不希望它以一般段落文件的方式去排版，而是照原來的樣子顯示，Markdown會用 `<pre>` 和 `<code>` 標籤來把程式碼區塊包起來。

要在Markdown中建立程式碼區塊很簡單，只要簡單地縮排4個空白或是1個tab就可以，例如，下面的輸入：

```
This is a normal paragraph:
```

```
    This is a code block.
```

Markdown會轉換成：

```
<p>This is a normal paragraph:</p>
```

```
<pre><code>This is a code block.</code></pre>
```

這個每行一階的縮排（4個空白或是1個tab），都會被移除，例如：

```
Here is an example of AppleScript:
```

```
    tell application "Foo"
        beep
    end tell
```

會被轉換為：

```
<p>Here is an example of AppleScript:</p>
```

```
<pre><code>tell application "Foo"
    beep
end tell</code></pre>
```

一個程式碼區塊會一直持續到沒有縮排的那一行（或是文件結尾）。

在程式碼區塊裡面，`&`、`<`和`>`會自動轉成HTML實體，這樣的方式讓你非常容易使用Markdown插入範例用的HTML原始碼，只需要複製貼上，再加上縮排就可以了，剩下的Markdown都會幫你處理，例如：

```
    <div class="footer">
        &copy; 2004 Foo Corporation
    </div>
```

會被轉換為：

```
<pre><code>&lt;div class="footer"&gt;
    &amp;copy; 2004 Foo Corporation
&lt;/div&gt;
</code></pre>
```

程式碼區塊中，一般的Markdown語法不會被轉換，像是星號便只是星號，這表示你可以很容易地以Markdown語法撰寫Markdown語法相關的文件。

## 分隔線



你可以在一行中用三個或以上的星號、減號、底線來建立一個分隔線，行內不能有其他東西。你也可以在星號中間插入空白。下面每種寫法都可以建立分隔線：

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

## 區段元素

### 連結

Markdown支援兩種形式的連結語法：行內和參考兩種形式。

不管是哪一種，連結的文字都是用 [方括號] 來標記。

要建立一個行內形式的連結，只要在方塊括號後面馬上接著括號並插入網址連結即可，如果你還想要加上連結的title文字，只要在網址後面，用雙引號把title文字包起來即可，例如：

```
This is [an example](http://example.com/ "Title") inline link.  
  
[This link](http://example.net/) has no title attribute.
```

會產生：

```
<p>This is <a href="http://example.com/" title="Title">  
an example</a> inline link.</p>  
  
<p><a href="http://example.net/">This link</a> has no  
title attribute.</p>
```

如果你是要連結到同樣主機的資源，你可以使用相對路徑：

```
See my [About](/about/) page for details.
```

參考形式的連結使用另外一個方括號接在連結文字的括號後面，而在第二個方括號裡面要填入用以辨識連結的標籤：

```
This is [an example][id] reference-style link.
```

你也可以選擇性地在兩個方括號中間加上空白：

```
This is [an example] [id] reference-style link.
```

接著，在文件的任意處，你可以把這個標籤的連結內容定義出來：

```
[id]: http://example.com/ "Optional Title Here"
```

連結定義的形式為：

- 方括號，裡面輸入連結的辨識用標籤
- 接著一個冒號
- 接著一個以上的空白或tab
- 接著連結的網址
- 選擇性地接著title內容，可以用單引號、雙引號或是括弧包著

下面這三種連結的定義都是相同：

```
[foo]: http://example.com/  "Optional Title Here"  
[foo]: http://example.com/  'Optional Title Here'  
[foo]: http://example.com/  (Optional Title Here)
```

請注意：有一個已知的問題是Markdown.pl 1.0.1會忽略單引號包起來的連結title。

連結網址也可以用角括號包起來：

```
[id]: <http://example.com/>  "Optional Title Here"
```

你也可以把title屬性放到下一行，也可以加一些縮排，網址太長的話，這樣會比較好看：

```
[id]: http://example.com/longish/path/to/resource/here  
    "Optional Title Here"
```

網址定義只有在產生連結的時候用到，並不會直接出現在文件之中。

連結辨識標籤可以有字母、數字、空白和標點符號，但是並不區分大小寫，因此下面兩個連結是一樣的：

```
[link text][a]  
[link text][A]
```

預設的連結標籤功能讓你可以省略指定連結標籤，這種情形下，連結標籤和連結文字會視為相同，要用預設連結標籤只要在連結文字後面加上一個空的方括號，如果你要讓“Google”連結到google.com，你可以簡化成：

```
[Google][]
```

然後定義連結內容：

```
[Google]: http://google.com/
```

由於連結文字可能包含空白，所以這種簡化的標籤內也可以包含多個文字：

```
Visit [Daring Fireball][] for more information.
```

然後接著定義連結：

```
[Daring Fireball]: http://daringfireball.net/
```

連結的定義可以放在文件中的任何一個地方，我比較偏好直接放在連結出現段落的後面，你也可以把它放在文件最後面，就像是註解一樣。

下面是一個參考式連結的範例：

```
I get 10 times more traffic from [Google] [1] than from
[Yahoo] [2] or [MSN] [3].
```

```
[1]: http://google.com/      "Google"
[2]: http://search.yahoo.com/ "Yahoo Search"
[3]: http://search.msn.com/   "MSN Search"
```

如果改成用連結名稱的方式寫：

```
I get 10 times more traffic from [Google][] than from
[Yahoo][] or [MSN][].
```

```
[google]: http://google.com/      "Google"
[yahoo]:   http://search.yahoo.com/ "Yahoo Search"
[msn]:     http://search.msn.com/   "MSN Search"
```

上面兩種寫法都會產生下面的HTML。

```
<p>I get 10 times more traffic from <a href="http://google.com/"
title="Google">Google</a> than from
<a href="http://search.yahoo.com/" title="Yahoo Search">Yahoo</a>
or <a href="http://search.msn.com/" title="MSN Search">MSN</a>.</p>
```

下面是用行內形式寫的同樣一段內容的Markdown文件，提供作為比較之用：

```
I get 10 times more traffic from [Google](http://google.com/ "Google")
than from [Yahoo](http://search.yahoo.com/ "Yahoo Search") or
[MSN](http://search.msn.com/ "MSN Search").
```

參考式的連結其實重點不在於它比較好寫，而是它比較好讀，比較一下上面的範例，使用參考式的文章本身只有81個字元，但是用行內形式的連結卻會增加到176個字元，如果是用純HTML格式來寫，會有234個字元，在HTML格式中，標籤比文字還要多。

使用Markdown的參考式連結，可以讓文件更像是瀏覽器最後產生的結果，讓你可以把一些標記相關的資訊移到段落文字之外，你就可以增加連結而不讓文章的閱讀感覺被打斷。

## 強調

Markdown使用星號（\*）和底線（\_）作為標記強調字詞的符號，被\*或\_包圍的字詞會被轉成用<em>標籤包圍，用兩個\*或\_包起來的話，則會被轉成<strong>，例如：

```
*single asterisks*

_single underscores_

**double asterisks**

__double underscores__
```

會轉成：

```
<em>single asterisks</em>

<em>single underscores</em>

<strong>double asterisks</strong>

<strong>double underscores</strong>
```

你可以隨使用你喜歡的樣式，唯一的限制是，你用什麼符號開啟標籤，就要用什麼符號結束。

強調也可以直接插在文字中間：

```
un*frigging*believable
```

但是如果你的 `*` 和  兩邊都有空白的話，它們就只會被當成普通的符號。

如果要在文字前後直接插入普通的星號或底線，你可以用反斜線：

```
\*this text is surrounded by literal asterisks\*
```

## 程式碼

如果要標記一小段行內程式碼，你可以用反引號把它包起來（```），例如：

```
Use the printf() function.
```

會產生：

```
<p>Use the printf() function.</p>
```

如果要在程式碼區段內插入反引號，你可以用多個反引號來開啟和結束程式碼區段：

```
``There is a literal backtick (`) here.``
```

這段語法會產生：

```
<p><code>There is a literal backtick (`) here.</code></p>
```

程式碼區段的起始和結束端都可以放入一個空白，起始端後面一個，結束端前面一個，這樣你就可以在區段的一開始就插入反引號：

```
A single backtick in a code span: `` ` ``
```

```
A backtick-delimited string in a code span: `` `foo` ``
```

會產生：

```
<p>A single backtick in a code span: `</p>
```

```
<p>A backtick-delimited string in a code span: `foo`</p>
```

在程式碼區段內，`&`和角括號都會被轉成HTML實體，這樣會比較容易插入HTML原始碼，Markdown會把下面這段：

```
Please don't use any <blink> tags.
```

轉為：

```
<p>Please don't use any &lt;blink> tags.</p>
```

你也可以這樣寫：

```
`&#8212;` is the decimal-encoded equivalent of `&mdash;`.
```

以產生：

```
<p><code>&#8212;</code> is the decimal-encoded  
equivalent of <code>&mdash;</code>.</p>
```

## 圖片

很明顯地，要在純文字應用中設計一個「自然」的語法來插入圖片是有一定難度的。

Markdown使用一種和連結很相似的語法來標記圖片，同樣也允許兩種樣式：行內和參考。

行內圖片的語法看起來像是：

```
![Alt text](/path/to/img.jpg)  
  
![Alt text](/path/to/img.jpg "Optional title")
```

詳細敘述如下：

- 一個驚嘆號！
- 接著一個方括號，裡面放上圖片的替代文字
- 接著一個普通括號，裡面放上圖片的網址，最後還可以用引號包住並加上 選擇性的'title'文字。

參考式的圖片語法則長得像這樣：

```
![Alt text][id]
```

「id」是圖片參考的名稱，圖片參考的定義方式則和連結參考一樣：

```
[id]: url/to/image "Optional title attribute"
```

到目前為止，Markdown還沒有辦法指定圖片的寬高，如果你需要的話，你可以使用普通的標籤。

---

## 其他

### 自動連結

Markdown支援比較簡短的自動連結形式來處理網址和電子郵件信箱，只要是用角括號包起來，Markdown就會自動把它轉成連結，連結的文字就和連結位置一樣，例如：

```
<http://example.com/>
```

Markdown會轉為：

```
<a href="http://example.com/">http://example.com</a>
```

自動的郵件連結也很類似，只是Markdown會先做一個編碼轉換的過程，把文字字元轉成16進位碼的HTML實體，這樣的格式可以混淆一些不好的信箱地址收集機器人，例如：

Markdown會轉成：

在瀏覽器裡面，這段字串會變成一個可以點擊的「[address@example.com](mailto:address@example.com)」連結。

## 跳脫字元

Markdown支援在下面這些符號前面加上反斜線來幫助插入普通的符號：

感謝

Made with <3 by othree.