

# 少年科技人

*Young Maker* 雜誌



讀書做善事、寫書做公益 – 歡迎程式人認養專欄或捐出您的網誌  
參考價：NT 50 元，如果您喜歡本雜誌，請將書款捐贈公益團體  
羅慧夫顱顏基金會 彰化銀行 (009) 帳號：5234-01-41778-800



愛心條碼

# 少年科技人雜誌

**2015 年 2 月**

本期焦點：開放原始碼的世界

# 少年科技人雜誌

- 前言
  - 編輯小語
  - 授權聲明
- 本期焦點
  - 開放原始碼授權
  - 進入開放原始碼領域的方法
- 科技人小傳
  - 開放原始碼之父-Richard Stallman
  - Linux 創建者-Linus Torvalds
  - Markdown 與 RSS 的創造者-Aaron Swartz
- 程式人文集
  - 我的開放原始碼之路

- 雜誌訊息
  - 讀者訂閱
  - 投稿須知
  - 參與編輯
  - 公益資訊

# 前言

## 編輯小語

在資訊工業的領域，『開放原始碼』已經是極度重要的工業基礎建設，其影響力遠遠超過 蘋果與微軟等廠商，但對還沒進入『資訊工業』的少年科技人而言，通常很難一窺『開放原始碼』領域的奧秘。

因此我們將在本期當中全面性的介紹有關『開放原始碼』的『歷史、授權、軟體、硬體、人物』，還有『開放原始碼與工業體系』之間的關係，希望能透過這樣的介紹讓少年科技人對此一領域有更完整深入的瞭解。

----（「少年科技人雜誌」編輯 - 陳鍾誠）

## 授權聲明

本雜誌許多資料修改自維基百科，採用 創作共用：[姓名標示、相同方式分享](#) 授權，若您想要修改本書產生衍生著作時，至少應該遵守下列授權條件：

1. 標示原作者姓名 (包含該文章作者，若有來自維基百科的部份也請一併標示)。
2. 採用 創作共用：[姓名標示、相同方式分享](#) 的方式公開衍生著作。

另外、當本雜誌中有文章或素材並非採用 [姓名標示、相同方式分享](#) 時，將會在該文章或素材後面標示其授權，此時該文章將以該標示的方式授權釋出，請修改者注意這些授權標示，以避免產生侵權糾紛。

例如有些文章可能不希望被作為「商業性使用」，此時就可能會採用創作共用：[\[姓名標示、非商業性、相同方式分享\]](#) 的授權，此時您就不應當將該文章用於商業用途上。

最後、懇請勿移除公益捐贈的相關描述，以便讓愛心得以持續散播！

# 本期焦點

## 開放原始碼授權

雖然說、『開放原始碼』的表面意義是將原始碼開放，但背後其實是有很多種不同的法律授權可供選擇，深入了解『開放原始碼授權』之後，才能理解何時該用什麼方式授權，以及如何使用這些『開放原始碼』作品才不會侵權等問題。

**BSD** 與 **GPL** 是兩種具有代表性的開放原始碼授權，其中 **BSD** 屬於非常寬鬆的類型，而 **GPL** 則屬於非常嚴格的類型，就讓我們從這兩種授權開始看起。

### **BSD** 授權

早在有開放原始碼之父 **Richard Stallman** 推動開放原始碼運動之前，就有 **BSD** 這個授權了，像是柏克萊大學的網路 **socket** 函式庫就是採用 **BSD** 授權的，而

經典的作業系統 UNIX 的關鍵版本 V6 也是採用 BSD 授權釋出的，以下網址是目前開源組織所公佈的BSD-2授權。

- <http://opensource.org/licenses/BSD-2-Clause>

但是、由於 BSD 授權只規定了程式修改者必須標示原作者姓名，而且原作者並不為該程式提供保證，但是卻沒有強程式修改後要開放原始碼，於是後來 UNIX 就被很多大公司拿去修改而且不再開放原始碼了，導致 UNIX 後續的版本成為各家大公司私有財產。

這種結果讓很多程式人員都無法在閱讀並修改新版的UNIX原始碼了，於是這個經典的作業系統就被私有化了。

這種現象讓某些程式人員感到很不滿，特別是 Richard Stallman 。

## GPL 授權

為了對抗這種原始碼私有化的情形， Richard Stallman 除了努力寫出一堆好程



式之外，還提出了一種新的授權方式，稱為 **GPL**，以下是**GPL**第三版的授權網址。

- <http://opensource.org/licenses/GPL-3.0>

**GPL** 强制規定任何用到 **GPL** 授權的開放原始碼程式，以及這些程式的修改版都必需要開放原始碼，透過這種授權可以確保這些軟體不會被大公司私有化，否則軟體創作者或開放原始碼組織就可以提出法律訴訟並且請求賠償，然後再强制對方要開放原始碼。

於是、**GPL** 標誌了開放原始碼運動的正式啓動，有了 **GPL** 之後，程式碼就比較不容易被私人公司拿去修改並封閉起來了，而這也正是 **Richard Stallman** 被稱為開放原始碼之父的重要原因之一。

事實上、除了 **BSD** 和 **GPL** 授權之外，還有很多『開放原始碼授權』，像是『[Apache License](#)、[MIT License](#)』等等，每一個授權的條款都有些不同，因此在修改開源程式之前，務必要先了解所使用的程式之授權範圍如何，如此才

不會侵犯開放原始碼授權的規定。

## 創作共用授權

以上所說的授權都是針對程式碼而言的，但是自從 1990 年 web 網路被發展出來之後，網路開始進入人們的日常生活，於是很多人開始將自己的文章或圖片創作放上網路。

但是、我們到底能不能合法的修改並使用這些作品呢？

原則上、如果對方沒有聲明授權方式，那麼根據『著作權法』的規定，這些著作就受到『著作權法』的保護，因此就都是『版權所有、翻印必究』的，所以我們幾乎完全不能修改這些作品，也不能轉貼這些文章的。

為了解決這個問題，哈佛大學的憲法學教授 Lawrence Lessig 創造了一系列新的授權方式與標章，稱為『創作共用』（Creative Commons, 簡稱為 CC）授權。







Lawrence Lessig 事實上也是從開放原始碼領域的 GPL 等授權所得到的靈感，以下是『創作共用』授權的四種基本屬性與圖示。



[圖、創作共用授權的四種圖示]

CC 授權中的「姓名標示」為必選，透過這四種屬性，可以組合出六種合理的基本授權組合，這六種授權組合如下表所示，您可以透過 CC 授權釋出作品，也可以尋找網路上的 CC 授權文章、圖片、影片等進行衍生創作，修改後再度創作，本雜誌就是採用CC的 [姓名標示](#)、[相同方式分享](#) 的授權釋出的。



授權圖示	說明
	姓名標示
	姓名標示、非商業性
	姓名標示、相同方式分享
	姓名標示、禁止改作
	姓名標示、非商業性、相同方式分享
	姓名標示、非商業性、禁止改作

結語

現在、『開放原始碼授權』與『創作共用授權』在網路上被使用的情況也越來越普遍了，於是我們有了不少合法授權的程式碼、文章與圖片可以修改使用了。

但必須注意的是，這些授權並非毫無限制的，您必須先認識這些授權的範圍，才能有效並合法的運用這些開放原始碼資源。

## 進入開放原始碼領域的方法

對於還沒有進入『電腦軟硬體工業』的人們而言，很多人並不清楚開放原始碼對目前資訊工業的影響力到底如何？因為很多學校都還是採用商業公司，特別是微軟的解決方案在進行教學。大部分學校的電腦教室也都還是安裝著 Windows，而非開放原始碼的 Linux。

但是、一但進入資訊工業領域，特別是偏向系統程式的領域，開放原始碼的影響力就會變得非常強大，從底層的 Linux 作業系統、Gcc 編譯器工具鏈、到應用層的 webkit 瀏覽器、Apache, Nginx 伺服器、MySQL, PostgreSQL 資料

庫、Python, Ruby, PHP, Node.js 等開發環境，甚至到整個雲端架構的 OpenStack, QEMU, Docker, Hadoop 等等，還有 google 推出的 android 手機系統等等，都是開放原始碼的天下。

這種情況對於一位還沒進入工業體系的少年科技人而言，可以說是好事也是壞事。好的一面是我們可以透過開放原始碼理解整個資訊工業體系的各種技術，並且從中學習並參與資訊工業的運作。但是壞的一面是，開放原始碼的技術太多，對於剛入門的人而言，很容易會迷失方向而找不到頭緒，相對的比較難以入門。

舉例而言，現在要學習 web 網站的程式設計，我們可以選擇採用 PHP , ASP, Python, Ruby 或 Node.js 等方案，而當你進入這些方案之後，又會發現有更多紛亂的技術並陳在你的面前，於是『歧路之中、又有歧焉』的情況就會讓你難以抉擇，這時很多人就會無所適從了。

以 Node.js 為例，當您想學習此一技術時，就會發現需要學習前端的 HTML/CSS/JavaScript/jQuery/Bootstrap/React.js/Angular.js/Backbone.js 等技

術，還有前後兩端的 Express.js/Connect.js/Socket.io 等套件。還有前後端都可以用的 underscore.js/Async.js/Assert/Vows/Mocha 等等，以及用來產生網頁的樣板引擎 jade, ejs，以及 Redis, MongoDB, CouchDB, Mongoose 等相關軟體，然後還必須了解開放原始碼交換平台 github，google code, sourceforge 等網站，以及最後要上線到雲端的宿主網站 Heroku, Cloud Foundry, Nodester, Amazon EC2 等等服務，這可以說是相當複雜的一個世界。

而且、這些還不是全部！

由於每個人都可以自行創見開放原始碼專案，所以您可以用在 node.js 上建制網站的開源專案恐怕有超過一萬個，我想沒有任何人可以對所有專案都能有深入的了解。

還好、我們並不需要先理解所有的這些才能開始寫程式或建構自己的專案，通常我們會採用『一邊學一邊做』的方式，需要什麼就學什麼，需要哪類專案就去尋找，然後用『吸星大法』納為己用，而不是一開始就學會所有的技術，才開始去製作專案的。

所以、如果您想透過開放原始碼，成為一位專業的程式人，筆者會建議您在閱讀一兩本入門書籍之後，就開始構思自己想參與哪種專案，或者想建構出什麼樣的程式。然後採用邊學邊做的方式，完成您想完成的專案。

當您透過這種方式完成了一些專案的時候，您就已經從『素人』開始變成『專家』了，這時候您就已經身在開放原始碼所建構出來的資訊工業體系裡，成為其中的一位『程式人』了。

以下是筆者非常喜歡的一部影片，這是 `pcman` 的創建者『洪任諭醫師』講述自己如何透過建構 `pcman` 這個連接 BBS 的專案，從『程式素人』變成『程式專家』的故事，在此分享給讀者參考。

- [YouTube: 那些年，我們一起上的 BBS: 洪任諭 PCMan at TEDxNCTU 2013](#)



# 科技人小傳

## 開放原始碼之父–**Richard Stallman**

Richard Stallman 是開放原始碼領域最知名的人物之一，也是創造整個開源文化的始祖，他於 1983 年創建了 GNU 計畫，想建立一個類似 UNIX 的作業系統，雖然這個作業系統卻遲遲未能完成，但是 GNU 卻創造出了完成這個作業系統的大部分工具，像是 gcc, gdb, glibc, Emacs 等等，這讓後來的 (Linus Torvalds) 能夠在此基礎上創造出 Linux。



[圖、Richard Stallman 2010 年的照片]

另外、Stallman 也創造出 GPL (GNU General Public License) 這個重要的開放原始碼授權，並且找了律師一起制定相關的法律條款，為開放原始碼的發展奠定了法律基礎。2001 年時，Lawrence Lessig 也是受到這類授權的啓發而創造出了 Creative Commons 創作共用授權，讓開放原始碼文化擴散到一般領域。

根據 Stallman 的傳記，他是因為想要改良一台 Xerox 9700 的雷射印表機的功能，槓上了全錄公司，由於該印表機沒有附上驅動程式的原始碼，並且拒絕 Stallman 等人的索取程式碼，所以才引發了他推展開放原始碼運動的想法。

這個故事告訴我們，有時候毫不起眼的小事，會改變你的一生，然後讓你變成偉人！

【本文由陳鍾誠取材並修改自維基百科】

## Linux 創建者-Linus Torvalds

Linus Torvalds 這位仁兄我想不用介紹，大家都知道他做了甚麼事，因為有一個作業系統幾乎就是用他名字命名的，Linu-X，簡稱 Linux。



## [圖、開放原始碼大師 -- Linus Torvalds]

Linus Benedict Torvalds 於 1969年12月28日出生於芬蘭赫爾辛基市，擁有美國國籍。他除了創造並主導 Linux 核心的開發之外，還發展出 Git 這個廣為世人使用的版本管理系統。如果諾貝爾獎設立程式領域的項目，那他肯定是得獎者之一。

Torvalds 畢業於赫爾辛基大學計算機科學系，1997年至2003年在美國加州矽谷任職於全美達公司（Transmeta Corporation）參與該公司晶片的code morph技術研發。後受聘於開源碼發展實驗室（OSDL : Open Source Development Labs, Inc），全力開發 Linux 核心，並任職於 Linux 基金會。

托瓦茲與妻子托芙（Tove，一位六次芬蘭前女子空手道冠軍）育有三名孩子。

Torvalds 在大學時代，由於想在 x86 個人電腦上使用類似 UNIX 的系統，但是卻找不到這樣的作業系統。只看到一個由 Andrew Stuart Tanenbaum 這位「荷

蘭阿姆斯特丹自由大學的計算機科學教授」所寫的小型 MINIX 作業系統可用，於是他在研究了 MINIX 之後，於 1991 年開始就逐步改造了 MINIX，最後全部換成了自己的程式碼，並將相關訊息放到 Usenet 新聞群組 comp.os.minix 上，標題如下所示：

Linus Benedict Torvalds (5 October 1991). "Free minix-like kernel sources for 386-AT".

從那時候開始，Linux 就迅速發展，Linux 站在 GNU gcc 這樣的開發工具上，逐步壯大起來，並且吸引了越來越多的程式人員 為此平台開發程式。

Torvalds 開發的是 Linux Kernel，也就式作業系統的核心部分，然後許多人將此核心連同一大群軟體，包裝成許多的分支，像是 RedHat, Ubuntu, Fedora, CentOS, Debian 等等。

後來、連商業公司也開始進來了，基於 Linux 的 Android 作業系統在手機上取得了重大的成功，這讓 Google 這樣的公司 進入了作業系統的領域，並結合整

個電子產業的硬體公司，像是 HTC、三星、Motorola、ASUS 等，占領了智慧型手機的大部分市場。

Linux 改造了整個電腦工業，其創造者 Torvalds 扮演著重要且關鍵的角色。

Torvalds 的另一個重要貢獻是創造並主導了 Git 這個版本管理系統的開發，現在 Git 已經凌駕 CVS, SVN 與 Mercurial，成為程式人員最常使用的版本管理系統，著名的 github 網站就提供了這樣一個平台，讓全世界的人可以將原始碼上傳並且透過 git 進行開發與合作，連這本「程式人雜誌」也是透過 github 發行並提供給讀者下載的。

Torvalds 的驚人貢獻，來自學生時代的一個想法，並將這個想法付諸行動，成就了不凡的一生。

## 參考文獻

- [Wikipedia:Linus\\_Torvalds](#)
- [維基百科:林納斯·托瓦茲](#)

- [Mac OS X 背后的故事（二）——Linus Torvalds的短视](#)

【本文由陳鍾誠取材並修改自維基百科】

## Markdown 與 RSS 的創造者-Aaron Swartz





[圖、Aaron Swartz 2008 年 12 月 13 日在 Creative Commons event 的照片]

Aaron Swartz 是個程式領域的天才，但是他的人生，卻以悲劇收場。

Aaron Swartz 於 1986 年 11 月 8 日出生美國芝加哥的伊利諾伊州一個猶太家庭。他的父親本身就是程式領域的創業者，這讓他從很小就開始接觸程式與網路，並於 13 歲時就贏的了 ArsDigita Prize 比賽的冠軍。

14 歲時，斯沃茨加入了 RSS-DEV 組織，與夥伴們一同創造出了 RSS 規格，後來他積極參與各種網路規格制定組織，像是為 W3C 撰寫 RFC 以定義 RDF 及 XML 等規格，以及參與創作共用 Creative Commons (CC) 組織，並成為了 CC 創辦人 Lawrence Lessig 指導的研究員。

如果您仔細觀察 Aaron Swartz 的程式、文章與經歷，您會發現他的生命幾乎完全投入在數位出版領域，而且有非常大個貢獻。

舉例而言、除了參與 RSS, RDF, XML 規格的制定，並參與 CC 組織之

外，Aaron Swartz 還與 John Gruber 共同創造了 Markdown 這個影響深遠的文件格式，這個格式已成為 github, stackoverflow 等網站的主要書寫格式，並且有衆多工具 pandoc, kitabu 被創造出來將 markdown 用在出版領域，而且也有像 leanpub 這樣的出版網站使用此種格式進行出版。

另外、他還創造出 web.py 這個以 Python 語言寫成的網站框架，用來作為出版平台。並且改寫了 Reddit.com 這個訊息分享網站的程式，用程式讓訊息的分享更為容易。

正是因為對數位出版的熱愛，讓 Aaron Swartz 更感覺到當今商業公司把持了大量的出版利益，並且透過法律體系抑制版權開放的運動，因此他寫下了「游擊隊開放存取宣言」(Guerrilla Open Access Manifesto)，並於 2010 年成立了 DemandProgress.org 網站，反對 SOPA/PIPA 這些網際網路審查法案。

接著、他更在 2010 年 9 月，在 MIT 校園內透過網路自動大量下載了 480 萬份 JSTOR 的期刊論文與相關文件，期望讓這些學術論文公開自由的被存取。接著在 2011 年 1 月 6 日時，他被聯邦調查局逮捕了，並控以「電信詐欺 (wire

fraud)、電腦詐欺 (computer fraud)、從受保護電腦非法取得資訊、故意毀壞受保護電腦」等四項罪名。

後來這些罪名陸續增加，從 4 項變為 13 項，並且檢察官威脅他要以刑事罪起訴，這讓他非常沮喪。

2013 年 1 月 11 日時，他在住處自殺身亡。

Aaron 死了，年僅 26 歲！

## 參考文獻

- Aaron Swartz 的個人網站 -- <http://www.aaronsw.com/>
- 紀念 Aaron Swartz 的網站 -- <http://www.rememberaaronsw.com/>
- Inside 硬塞的網路趨勢觀察:早逝的天才，網路神童與資訊自由鬥士:Aaron Swartz
- PanSci 泛科學:美國司法部如何追殺資訊自由化推手 Aaron Swartz
- OpenFoundary:網路運動者 Aaron Swartz 自殺身亡，各方紛紛表達哀悼紀

## 後記

按理說，那些學者寫的論文，應該是全人類共有的文明資產才對，但是為何會變成幾家出版商所控制的私有資產呢？非學術界的人通常無法理解此種情況，而學術界的人，則是理解之後仍然將論文交給出版商，繼續為他們創造私有資產。

目前的情況是、學者們所寫的論文投稿到期刊之後，商業權將移轉到期刊公司，於是那些未付費就下載這些論文的行為 就都成了侵害私有財產權的行為。

更糟糕的是，連那篇論文的作者，也不能再出版自己的論文，有時連將自己的論文放上網路都成了非法行為。

對這個問題有興趣的朋友可以參看以下的幾篇文章。

- 科技橘報: 「封鎖知識」VS.「釋放知識」——致 Aaron Swartz
- 洪朝貴, 學者的研究成果淪為出版商的生財工具—談學術作品的開放近用
- 洪朝貴, 學者的智慧, 期刊的財產, 圖書館的業績...或是負擔?
- 陳鍾誠, 為台灣教育界投下一顆震撼彈! (續篇)

【本文由陳鍾誠取材並修改自維基百科】

# 程式人文集

## 我的開放原始碼之路

在 2000 年之前，我幾乎都只會用微軟等公司所推出的商用軟體，但是自從在松下 PTL 兼差研發程式之後，慢慢的接觸到了開放原始碼工具，特別是像 `make` 與 `gcc` 等以 C 語言為核心的開發工具，當時只覺得這些東西好原始，為何不像微軟的 `Visual Studio` 那樣有完整的開發環境，不過還好用了一陣子也就習慣了。

後來我進入金門大學任教之後，有一次在台東舉辦的研討會上，聽到了洪朝貴老師講述開放原始碼軟體，當時對這些東西也沒什麼特別的好感，但卻開始逐漸理解隱藏在開源軟體背後的理念。

後來為了要在學校上課，又不想使用破解的盜版軟體，於是開始試用了一些

開放原始碼工具，像是用 **blender** 來教動畫設計、用 **R** 來教機率統計、用 **Icarus Verilog** 來教計算機結構等等。

經過這些年的『邊學邊教』的過程之後，我逐漸的認識了這個紛亂而多元的領域，也慢慢的在這個領域裡可以創建一些專案了。像是最近創建了 **wikidown.js** 維基網誌, **jslab** 機率統計平台, 與 **elearn** 英文學習系統等等。

雖然身在金門這個比較偏遠的環境，但是在寒暑假時我會回去看看有沒有什麼黑客松或程式相關研討會可以參加，目前曾經參加過兩次的 **COSCUP** 開源人年會與一次的 **SITCON** 學生計算機年會，這些活動對我進一步學習開放原始碼，以及認識程式領域的朋友都很有幫助，也建議剛入門的朋友可以多參加這些活動，相信也會收穫良多的。

最近我開始透過 **Kubuntu** 系統學習 **Linux**，因為 **Linux** 可以說是存取開放原始碼資源的一扇窗口，就像英文是吸取全世界知識的一扇窗口一樣，我相信透過 **Linux** 可以帶領我進一步深入開放原始碼的世界。

另外我開始嘗試用 `node.js`+`JavaScript` 來教授『網路程式設計』課程與實作「人工智慧、系統程式」等課程的相關程式，慢慢的開始移向 `Javascript` 這個語言與相關的開發環境，並打算改用 `JavaScript` 來教授大一新生第一學期的程式設計課程，然後在第二學期再改用 `C/C++` 以便讓同學銜接嵌入式系統的領域。希望透過 `JavaScript` 的引導讓我的學生們更容易的進入程式設計的領域，而不會被生硬的 `C` 語言卡住，造成學習上的瓶頸。

身為一個大學老師，我不喜歡作高深的學術研究，相反的、我喜歡透過分享一些自己寫的程式，讓學生們找到自己的開源之路，成為一名專業的程式設計師，讓他們有能力透過程式改變這個世界！



# 雜誌訊息

## 讀者訂閱

程式人雜誌是一個結合「開放原始碼與公益捐款活動」的雜誌，簡稱「開放公益雜誌」。開放公益雜誌本著「讀書做善事、寫書做公益」的精神，我們非常歡迎程式人認養專欄、或者捐出您的網誌，如果您願意成為本雜誌的專欄作家，請加入 [程式人雜誌社團](#) 一同共襄盛舉。

我們透過發行這本雜誌，希望讓大家可以讀到想讀的書，學到想學的技術，同時也讓寫作的朋友的作品能產生良好價值 – 那就是讓讀者根據雜誌的價值捐款給慈善團體。讀雜誌做公益也不需要壓力，您不需要每讀一本就急著去捐款，您可以讀了十本再捐，或者使用固定的月捐款方式，當成是雜誌訂閱費，或者是季捐款、一年捐一次等都 OK！甚至是單純當個讀者我們也都很歡迎！

本雜誌每期參考價：NT 50 元，如果您喜歡本雜誌，請將書款捐贈公益團體。例如可捐贈給「羅慧夫顱顏基金會 彰化銀行(009) 帳號：5234-01-41778-800」。(若匯款要加註可用「程式人雜誌」五個字)

## 投稿須知

**給專欄寫作者：** 做公益不需要有壓力。如果您願意撰寫專欄，您可以輕鬆的寫，如果當月的稿件出不來，我們會安排其他稿件上場。

**給網誌捐贈者：** 如果您沒時間寫專欄或投稿，沒關係，只要將您的網誌以 [創作共用的「姓名標示、非商業性、相同方式分享」授權] 並通知我們，我們會自動從中選取需要的文章進行編輯，放入適當的雜誌當中出刊。

**給文章投稿者：** 程式人雜誌非常歡迎您加入作者的行列，如果您想撰寫任何文章或投稿，請用 markdown 或 LibreOffice 編輯好您的稿件，並於每個月 25 日前投稿到[程式人雜誌社團](#) 的檔案區，我們會盡可能將稿件編入隔月1號出版程式人雜誌當中，也歡迎您到社團中與我們一同討論。

如果您要投稿給程式人雜誌，我們最希望的格式是採用 `markdown` 的格式撰寫，然後將所有檔按壓縮為 `zip` 上傳到社團檔案區給我們，如您想學習 `markdown` 的撰寫出版方式，可以參考 [看影片學 `markdown` 編輯出版流程] 一文。

如果您無法採用 `markdown` 的方式撰寫，也可以直接給我們您的稿件，像是 MS. Word 的 `doc` 檔或 LibreOffice 的 `odt` 檔都可以，我們 會將這些稿件改寫為 `markdown` 之後編入雜誌當中。

## 參與編輯

您也可以擔任程式人雜誌的編輯，甚至創造一個全新的公益雜誌，我們誠摯的邀請您加入「開放公益出版」的行列，如果您想擔任編輯或創造新雜誌，也歡迎到 [程式人雜誌社團](#) 來與我們討論相關事宜。

## 公益資訊

公益團體	聯絡資訊	服務對象	捐款帳號
財團法人羅慧夫 顱顏基金會	<a href="http://www.nncf.org/">http://www.nncf.org/</a> <a href="mailto:lynn@nncf.org">lynn@nncf.org</a> 02-27190408分機 232	顱顏患者 (如唇顎 裂、小耳症或其他 罕見顱顏缺陷)	銀行: 009 彰化銀行民 生分行 帳號: 5234-01- 41778-800
社團法人台灣省 兒童少年成長協 會	<a href="http://www.cyga.org/">http://www.cyga.org/</a> <a href="mailto:cyga99@gmail.com">cyga99@gmail.com</a> 04-23058005	單親、隔代教養.弱 勢及一般家庭之兒 童青少年	銀行: 新光 銀行 戶名: 台灣 省兒童少年 成長協會 帳號: 103- 0912-10-

			000212-0
--	--	--	----------