许可证

软件项目中最重要的文件就是许可证。许可证说明了软件的拥有者、哪些人在什么情况下可以使用该软件，以及拥有者和用户各自有哪些权利和保障。如果缺乏许可证文件，上述这些信息就无从得知。

许可证文件应该放在显眼的位置。一般会位于项目的顶级目录中，名称大多为 LICENSE、license.txt 或其他类似的形式。需要注意的是，有时同一个项目的不同部分会使用不同的许可证。有些项目针对不同的用途提供了不同的许可证。

常见的两类许可证：

1. 专有许可证：

通常由销售公司的法律团队编写，用于声明版权归公司所有，保护公司的权益，减少对公司的威胁

2. 免费/开源许可证：

能够让用户和开发人员自由查看和修改代码，帮助建立用户和开发人员社区，开发者们可以聚集起来完成一些重要任务，让软件项目成功进行下去。

需要注意的是：**最好使用现有的开源许可证**。通常自己编写的许可证是不具有法律效应的。

版权

深入讨论许可证之前，先了解一下版权。版权是法律术语，指的是对作品的所有权。在西方的判例中，**世界上大部分版权法都认为思想和观念不具有版权，但思想的表达方式有版权。**

哪些内容可以授予版权

1. 版权不适用于抽象的思想、概念或事实，如物理规律和数学常数（比如圆周率 Pi）。但一旦这些思想或概念以特定的方式表达出来（如通过文字、图片等），它们就可以获得版权保护。
2. 数字 Pi 本身没有版权，但如果你用 Pi 和其数值装饰一个馅饼，并拍摄图片，那张图片是你的创作，受版权保护。
3. 类似地，游戏规则本身不受版权保护，但发布的游戏版本和规则说明书的具体表达则会有版权。软件中的版权：软件的接口（如 API）是概念性的，它们本身没有版权。但接口的具体实现（即代码的写法）是受版权保护的。例如，计算标准偏差的算法实现是有版权的，但其概念（例如函数名 std() 和参数 vals）并没有版权。

首次发布权

当知道哪些内容是受版权保护的，那么就应该了解版权适用的时间。大多数版权系统具有所谓的首次出版权。这是因为该创意产品的第一个发布者自动拥有版权。无论当时发布商是否指定了此权利。这些法律保护出版者只要是他们首次实现了这些工作，那么这些工作就不会被盗。创意产品的第一个发布者自动拥有版权。对于**第一个没有许可证**的作品，版权属于作者，他人无法合法地使用修改该作品。软件许可证很重要，可用来保留版权，向他人说明代码的使用和修改软件的条款，也可以用来完全放弃所有的权利。

公共领域

公共领域是一个概念，社会作为一个整体来“拥有”一项工作，因此是免费的，任何人都可以使用和修改。因此作品属于大家，而不是某个个人。在大多数情况下，现有作品的版权将在特定年数（25、50、90 等）后过期，那时作品将进入公共领域。但版权会自然过期并不意味着一定要等待那么长时间。作者可以更快地将自己的作品添加到公共领域。

如果原作者不想涉及许可证或不想保留版权，那么会发生什么情况？如果只是为科学本身做贡献，想让大家能够无条件使用代码，那么可以将该工作放到公共领域（PD）中，然后用一句话描述一下：“这项工作已经属于公共领域”。

如果不想放在公共领域，但仍希望软件是免费和开放的，那么必须选择一个许可证。

选择一个软件许可证

多年来，针对不同的情况出现了许多许可证。最常见的是**专有许可证和免费/开源许可证:**

- 专有许可证通常由销售软件的公司编写。这些文档通常宣布版权归公司所有、拒绝赔偿因滥用软件导致的损失、保留对盗版软件诉讼的权力。专有许可证通常由一个律师团队手工编写，以尽量减少对公司的威胁。
- 自由和开源软件许可证（有时缩写为 FOSS、FLOSS 或 OSS）与计算物理软件关联度更高，特别是软件主要用于研究的情况下。研究型软件通常具有以下属性：1. 没有立即和直接的商业利益。2. 源代码必须由同行审阅。3. 改动频繁以适应研究者的需要。

许可证能够让用户和其他开发人员自由查看和修改代码，**对科学论文、教育、代码比较、代码质量保证、项目参与程度都有帮助。** 由于大多数研究人员没有足够的资金聘请数千名开发人员来完成所有这些活动，因此开源许可证可以帮助建立用户和开发人员社区。科学家们可以聚集在一起完成一些重要任务，让软件项目成功进行下去。

强烈建议读者使用现成的开源许可证。不要试图编写自己的许可证，一部分是因为读者不是律师，自己撰写是浪费时间。另一方面，只有获得法庭认可的许可证才是可信的。

现在来看看一些重要的许可证及其对开源的影响。

伯克利软件发布许可证 (BSD)

Berkeley 软件分发 (或 BSD 许可证) 实际上是三个许可证的集合, 分别被称为 BSD 4条款、3条款、2条款许可证。 在历史上, 最先出现的是 4 条款许可证, 而 2 条款是最新的。现在不再推荐使用 4 条款, 通常使用 3 和 2 条款。在下面讨论的所有许可证中, 建议在软件项目中使用 3 或 2 条款许可证。这两个是最适合科学和研究的许可证。诸如NumPy、SciPy、IPython和其他科学Python 生态系统的主要项目都使用这些许可证。

许可证详细内容请查看:https://opensource.org/license/BSD-3-clause

BSD 许可证称为授权许可证。这是因为该许可证允许在任何许可证下重新发布代码。只需显示原始版权声明即可。此外，该许可证对代码将来采用什么许可证没有要求。代码修改者可以自行选择其他许可证发布。授权许可为原始作者以外的用户和开发人员提供了很多自由，同时保护原作者免于承担责任。例如，假设 B 复制了 A 的代码。而 A 的代码最初是以 BSD 许可证发布。B 想修改代码并重新授权整个新的代码库。B 这么做完全可以，甚至不需要包括原始 BSD 许可证的副本。唯一的要求是需要包括版权声明 “Copyright (c), Person A.”，确保 A 位列其中，但这并不是个问题。

推荐 BSD 作为科学计算的许可证，主要就是为了修改和重新授权的自由。这种许可证为未来的科学家保留了最多的选择空间。MIT 许可证等同于 BSD 2 条款，可作为 BSD 的替代品。而下一节介绍的许可证是 BSD 的主要竞争对手之一。

GNU General Public License (GPL)

GNU 通用公共许可证 (GPL) 也是三个不同许可证的集合: GPLv1、GPLv2、GPLv3。 此外, 还有 v2.1 和 v3 GNU 轻量通用公共许可证 (Lesser General Public License, LGPL)。后者与相同主版本的 GPL 兼容, 但在精神上更接近 BSD 和 MIT 许可证。GPLv1 已过期, 不应使用。而 v3 和 v2 之间仍有争议。**Linux 几乎肯定是最大的使用 GPLv2 的项目, 且这种情况会一直持续下去。GNU 编译器集合 (GCC) 则可能是最大的使用 GPLv3 的项目。**

许可证详细内容请查看：GPLv2,GPLv3

需要注意的是，使用GPLv3许可证需要在v3许可证顶部添加如下内容：

one line to give the program's name and an idea of what it does. Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, see https://www.gnu.org/licenses.

GNU General Public License (GPL)

与 BSD 不同, GPL 许可证不是授权类型的。这是一种 copyleft 许可证。在 copyleft 许可证中, 对代码的修改必须以与原始代码相同 (或相似) 的方式获得许可。对于开源代码则意味着任何第三方 fork 的代码也必须开源 fork 的代码。原则上这是一个好主意。随着时间的推移, 这种方式能建立了一个开源程序的生态系统, 彼此之间进行良好的协作。但在实践中, 有时这种要求限制了开发者的自由度, 以致于甚至不能提交改进、修改或 fork 代码。

一般来说, GPL 许可证是软件项目的一个良好和合理的选择。但对于中小型项目来说, 可能会阻碍潜在的贡献者。因为即使是非常流行的物理程序, 其规模也不大, 因此默认情况下不推荐使用 GPL。可能仍然有一些情况。请读者务必仔细考虑。

这套许可证的另一个重要概念是 GPL 兼容性。FSF 定义的兼容性是指能否在不同许可证下发布代码库的不同部分, 且其中一个许可证是 GPL。为了与 GPL 兼容, 许可证不需要是 copyleft。值得注意的是, BSD 3 和 2 条款许可证都是 GPL兼容的。

即使 GPL 可能不适合科学工作，但仍然是一个非常成功的系列的许可证，已经得到法庭的认可。一般来说，如果你想要 GPL 式的 copyleft，但也希望在再分布方面的限制较少，LGPL 可以作为中间选择。

授权型的许可和 copyleft 型的许可证是开源软件的主要参与者。但是，有些许可证也可用于通用创意作品，并不限于软件。

知识共享 (CC)

知识共享 (CC) 许可证套件是一种替代方案，不仅适用于软件，还能更广泛地适用于所有创意作品。包括诗歌、散文、电影、音频等。CC 套件现在处于第四版，有时称为 v4。CC 许可证的目标是使开源软件中的共享和协作的想法更广泛地适用于其他工作。例如，维基百科上的所有内容都按照 CC 许可。这是一个非常成功的许可证。

知识共享许可证都相当冗长，因此无法在这里显示出来。这么长是为了让许可证在世界上的每个国家都保持同样的作用，因此文档相当冗长。而这是法律工程的巨大壮举。许可证详细内容请查看：
https://creativecommons.org/licenses/by/4.0/legalcode.zh-hans

CC 许可证有下面 4 种不同的形式：

- BY (Attribution)。其他人使用作品时，必须列出原来的版权所有人。这一条适用于所有 CC 许可证。
- SA (ShareAlike)。其他人若使用或修改作品，必须以相同的许可共享。
- ND (NoDerivatives)。其他人可以分享，但不能以任何方式修改作品。
- NC (NonCommercial)。只要不做商业用途，其他人可以随意使用或分享作品。

上面各种形式能够组成 6 种 CC 许可证，分别是 CC BY、CC BY-SA、CC BY-ND、CC BY-NC、CC BY-NC-SA、CC BYNC-ND。当然，ND 和 NC 不能放在一起。

知识共享 (CC)

CC BY 许可证与 BSD 许可证形式相同，也有允许修改的条款，但需要版权到期。CC BY-SA 许可证与 GPL 或 LGPL 形式相同，都是将 copyleft 条款添加到归属条款中。科学计算项目都可以使用这两个许可证，具体取决于是否需要 copyleft。

另一方面, ND 和 NC 对科学软件极为有害。这两款一般用于其他领域, 如艺术 (如网络漫画 xkcd 使用的是 CC BY-NC 许可证, 因此本书无法使用其中的火柴人形象)。但对于物理程序, 无法修改软件就让开源失去了作用。即使只是不让用 于商业用途应用程序。这样的许可条款阻碍了充满活力的科学和软件社区的进一步发展。使用 ND 或 NC 都是带有极端偏见。幸运的是, 在计算科学中很少见到 使用这些许可证。

最后, 知识共享提供了一个名为 CC0 的公共域替代（发音为 see-see-zero)。在许多 方面, 将代码作为 CC0 比放在公共领域更好。因为不同国家的公共领域定义有出 人, 而 CC0 适用于所有地方, 包括在没有定义公共领域类似法律的国家。此外, 在公有领域定义很庞大的国家, CC0 许可证又可以有效地减少所涵盖的领域。在这 里介绍的所有许可证中, CC0 是最自由的。虽然目前没有类似 Linux 这样的大型项 目使用这个许可证, 但其也正在慢慢获得注意。

总之, CC BY、CC BY-SA、CC0 都是科学软件项目的合理选择。而应该避免其他 许可证, 因为这些许可证会阻碍科学界的进步。

其他许可证

还有许多其他许可证，但本书无法深入讨论。虽然无论是宽松类型还是 copyleft 类型，其中甚至有自己独特的条款，但大部分都属于宽泛的类别。有些许可证已经在法庭上获得支持。与之前讨论的许可证不同，这里介绍的大多数许可证最初都是用于单个代码项目。

例如，Apache 许可证最初是为 Apache Web 服务器编写的，并且由 Apache Software Foundation 支持。这款许可证在 Apache 本身之外使用面也很广。与其他许可证不同，Apache 许可证包含与专利有关的条款。

Python Software Foundation 许可证适用于 CPython 解释器和一些其他 Python Software Foundation 代码的许可证。Python 生态系统的其他部分偶尔会用到，但外 界很少使用。这款许可证兼容 GPL。

而 JSON 许可证是一款有问题的许可证。这是一款在世界上很容易被滥用的许可证 之一。许可证的全文如下：该许可证中对好坏的定义有主观性，无法产生有效的约束。虽然一般能明白条款中好坏行为的粗略定义，但在严格功利的情况下，就会出现许多灰色地带。软件不能使用这么模糊的许可证。JSON 许可证只是假定开源软件不会做一些“更邪恶”的事情而已。

即使大多数开源软件开发人员和科学家反对做坏事，但明确地禁止这些行为也是不合理的。许可证中的条款是非强制的。假设有人真的想做坏事，想要使用 JSON 来做一些恶意的任务。JSON 许可证是不可能知道他们具体会做哪些事情的。许可证是规定性的，而不是预防性的。简单来说，在法庭上是无法指责说别人违反了“不许做坏事”这个模糊的条款。这个许可证也可作为一个警告，解释了为什么不应该编写自己的许可证，因为许可证的结果可能与预想的不同。

其他许可证

最后，芝加哥大学的 FLASH 代码有一个特别有趣的许可证。FLASH 是一种等离子体和高能量密度物理程序。FLASH 是开源的，但用户不能直接重新发布。此外，在用户访问代码之前，须要签署一个文档向 Flash Center 注册，来让 Flash Center 授予代码的访问权。FLASH 许可证的第 7 项在学术性质上是独一无二的：

1. Use Feedback. The Center requests that all users of the FLASH Code notify the Center about all publications that incorporate results based on the use of the code, or modified versions of the code or its components. All such information can be sent to info@flash.uchicago.edu.

这里尝试收集该许可证对科学软件影响的统计数据。考虑到 FLASH 网站上有 850多个出版物，因此这个策略非常成功。FLASH 许可证的其他非标准规定是针对特定情况，但不建议读者使用诸如 FLASH 之类的许可证（实际上，笔者都找不到另一种类似的许可证）。读者的代码应该不会遇到像 FLASH 代码的情况。而前面章节中介绍的许多许可证会更利于科学计划。搞特殊化对软件许可证并没有什么帮助。现在已经介绍了许多许可证类型，包括授权型、copyleft 型，以及一些更加奇特的许可证类型。下一步可以考虑如何将一个项目从一个许可证转移到另一个许可证上了。

更换许可证

有时开发人员会决定更改代码分发的许可证。这可能是因为在许可证中发现的法律问题，因为许可证已经不适应软件的生态系统。或者是认为其他许可证能更好地维持项目。在计算物理项目中，yt 就成功地更换了许可证。读者可以阅读项目博客上的说明。

即使在最好的条件下，更改许可证也是一个漫长而痛苦的过程。即使当前所有开发人员都同意更换，实际过程中仍有可能失败。

更换开源项目许可证需要获得以前所有贡献者的同意。贡献者以主动（签名放弃、发送电子邮件等）或被动（如果我们在三个月内没有收到您的回复，我们将视您表示赞同）的形式同意。如果之前或现在的开发者不同意，则开发人员工作的代码必须保留其原始许可证，或必须重写该人员的所有代码以符合新许可证。这两个问题都令人头疼。

如果有相当一部分人不同意更换许可证，那就很麻烦了。在这种情况下，从头开始编写代码比强制使用现有代码库的问题更简单。实际上，这种焦土策略非常频繁。大部分情况是因为试图在 copyleft 和授权型许可证之间切换。

更换许可证

也就是说更换许可证并非不可能，只是很困难。至少需要三四个月才能完成。更换许可证的步骤如下：

1. 在私下，向当前核心开发人员提出更换新许可证的想法。讨论新许可证的适用性，并获得一定程度的支持。然后征求开发小组的反馈意见。如果当前核心团队不感兴趣则停止。
2. 基于先前所有核心开发人员的反馈，公开讨论更改许可证。征求开发者的意见并衡量利益。如果核心团队的所有成员都同意改变，且大多数当前开发人员支持或不在乎，那么就继续。否则停止。
3. 公开向用户征求更改许可证的意见。如果由于变更导致项目将失去大部分用户，那么就停止。用户是最宝贵的资源，不值得在这方面受到损失。
4. 承诺仅在发布后续版本时更改许可证。不要将许可证追溯于先前版本。
5. 获得所有当前和过去的开发人员的书面同意，让他们接受更换许可证。可以使用电子邮件，并为提交同意书设置一个正常的时间窗口，以便他人提出反对意见。通常一个月就足够了。此时修改仍有可能失败。
6. 等待适当的时间后，在下一个版本上重新授权代码。

版权不是一切

版权和许可证并不都是所谓的知识产权。专利涉及对思想和过程的临时垄断。专利背后的原始想法是允许创作者、发明者、创新者有机会将他们的想法推向市场。专利软件最近受到极大的关注，因为“专利巨魔”囤积专利只是为了发起诉讼而不做其他事情。商标也属于知识产权领域，是在特定域中对企业或组织的唯一可识别符号。商标必须持续使用，否则会失效，此时另一个组织就可能采用这个商标。有些人认为只使用概括性术语“知识产权”就可以是不正确的，因为版权、专利、商标在法律上属于不同的领域，每一方面都需要一个专门从事这方面的律师。但对我们普通人来说，这个术语就够了。

知识产权不是控制软件的唯一工具。事实上，大多数知识产权并不真正适用于计算物理。在许多物理软件中还有其他更有效的机制。

出口管制是一个特别强大的制度。在未经政府明确许可的情况下，禁止将源代码转移到任何其他国家。通常，即使是有权访问的公民也必须签署一份文件，承诺不违反相关出口限制的条款。出口管制特别强大，不仅适用于软件，对数据和想法同样有效。来看一个极端且不合理的例子，政府可以在出口管制的基础上限制你告诉其他人关于 $ \pi $ 的价值。当程序出口受管制时，就不能放在互联网上。此时分享的规则就变得更加复杂。违反出口管制通常面临长期监禁，要对此有清楚的认识。

版权不是一切

物理程序的出口管制要比其他软件要为常见。典型原因是因为物理软件能用来帮助构建各种高度受管制的武器。此外，还有一些软件会包含或产生敏感或机密数据，
因此不能共享。数学和计算机科学程序有时由于能用于加密，所以也会受到出口管制。读者应该了解并遵守所在国政府有关出口管制的法律。

最后，美国在 1996 年颁布了《健康保险流通与责任法案》 (Health Insurance Portability and Accountability Act of 1996, 简称 HIPAA)。该法案针对医学，只是偶尔会涉及物理软件。其中要求软件必须能够合理地匿名处理患者信息。美国卫生和人类服务部负责监督确保软件符合 HIPAA 的规定。其他国家也有类似的法律。

这些替代方式能够非常有效地限制软件的工作形式。软件并不总是位于作者的控制之下。虽然一般不会涉及专利和商标等知识产权方面的问题，但 HIPAA 规则和隐私问题是绝对不能违反和侵犯的。每一个计算物理学家都需要随时了解出口管制的规则。在出口管制方面不能有丝毫问题。

本章小结

现在已经了解了围绕软件开发的一系列法律问题，以及这些问题如何应用于计算物理。应该熟悉以下知识：

- 许可证文档是项目中最重要的文件。
- 科学软件应该是免费和开源的。
- 对于大多数数计算科学项目，建议使用授权型的 BSD 许可证。
- Copyleft GPL 许可证也是合理的选择。
- 公共域是代码授权的一个很好的替代方案，但并不适用于所有地方。CC0 可作为公共域的替代。
- 创作共享许可证不仅适用于软件。
- 不要自行编写许可证。
- 更换许可证可能非常困难。
- 了解所在国家的出口管制法律很重要。