分子生物计算
(*Perl* 语言编程)

天津医科大学
生物医学工程与技术学院

2016-2017 学年上学期（秋）
2014 级生信班

第一章　Markdown 标记语言

伊现富（Yi Xianfu）

天津医科大学（TIJMU）
生物医学工程与技术学院

2016 年 11 月

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

Markdown | 简介

Markdown 是一种轻量级标记语言，创始人为约翰・格鲁伯（John Gruber）。它允许人们“使用易读易写的纯文本格式编写文档，然后转换成有效的 XHTML（或者 HTML）文档”。这种语言吸收了很多在电子邮件中已有的纯文本标记的特性。

Markdown | 宗旨

Markdown 的目标是实现“易读易写”，成为一种适用于网络的书写语言。

一份使用 Markdown 格式撰写的文件应该可以直接以纯文本发布，并且看起来不会像是由许多标签或是格式指令所构成。Markdown 语法受到一些既有 text-to-HTML 格式的影响，包括 Setext、atx、Textile、reStructuredText、Grutatext 和 EtText，而最大灵感来源其实是纯文本电子邮件的格式。

总之，Markdown 的语法全由一些符号所组成，这些符号经过精挑细选，其作用一目了然。比如：在文字两旁加上星号，看起来就像**强调**。Markdown 的列表看起来，嗯，就是列表。Markdown 的区块引用看起来就真的像是引用一段文字，就像你曾在电子邮件中见过的那样。

Markdown | 兼容 HTML

Markdown 的构想**不是**要使得 HTML 文档更容易书写。在我看来，HTML 已经很容易写了。Markdown 的理念是，能让文档更容易读、写和随意改。HTML 是一种**发布**的格式，Markdown 是一种**书写**的格式。因此，Markdown 的格式语法只涵盖纯文本可以涵盖的范围。

不在 Markdown 涵盖范围之内的标签，都可以直接在文档里面用 HTML 撰写。不需要额外标注这是 HTML 或是 Markdown；只要直接加标签就可以了。

Markdown | vs. HTML

Text using Markdown syntax	Corresponding HTML produced by a Markdown processor	Text viewed in a browser

```
# Heading

## Sub-heading

### Another deeper heading

Paragraphs are separated
by a blank line.

Two spaces at the end of a line
leave a
line break.

Text attributes _italic_,
*italic*, __bold__, **bold**,
`monospace`.

Horizontal rule:

---

Bullet list:

  * apples
  * oranges
  * pears

Numbered list:

  1. apples
  2. oranges
  3. pears

A [link](http://example.com).
```

```
<h1>Heading</h1>

<h2>Sub-heading</h2>

<h3>Another deeper heading</h3>

<p>Paragraphs are separated
by a blank line.</p>

<p>Two spaces at the end of a line
leave a<br />
line break.</p>

<p>Text attributes <em>italic</em>,
<em>italic</em>,
<strong>bold</strong>,
<strong>bold</strong>,
<code>monospace</code>.</p>

Horizontal rule:

<hr />

<p>Bullet list:</p>

<ul>
<li>apples</li>
<li>oranges</li>
<li>pears</li>
</ul>

<p>Numbered list:</p>

<ol>
<li>apples</li>
<li>oranges</li>
<li>pears</li>
</ol>

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

Heading

Sub-heading

Another deeper heading

Paragraphs are separated by a blank line.

Two spaces at the end of a line leave a line break.

Text attributes *italic*, *italic*, **bold**, **bold**, `monospace`.

Horizontal rule:

Bullet list:

- apples
- oranges
- pears

Numbered list:

1. apples
2. oranges
3. pears

A link.

Markdown | 用户

- Bitbucket 提供 Markdown 作为编写项目 README 文档的其中一种标记语言。
- GitHub 使用 Markdown 的一个分支版本来格式化评论、消息以及其它内容。
- Reddit 的编辑器使用了 Markdown 语法。
- Stack Overflow 以及其他 Stack Exchange Network 网站使用一种 Markdown 的分支作为它的文章格式化系统。
- 图灵社区使用 Markdown 语法供用户写作电子书。
- 简书写作网站支持 Markdown。
- 为知笔记是一种类似印象笔记的笔记软件，支持使用 Markdown 语法编辑笔记。
- ……

Markdown | 编辑器

作为一种小型标记语言，Markdown 很容易阅读，也很容易用普通的文本编辑器编辑。另外也有一些编辑器专为 Markdown 设计，可以直接预览文档的样式。

- Cmd Markdown 编辑阅读器：支持实时同步预览，区分写作和阅读模式，支持在线存储，分享文稿网址。
- Dillinger.io：一个在线 Markdown 编辑器，提供实时预览以及到 GitHub 和 Dropbox 的拓展链接。
- 简书：一个在线 Markdown 编辑器与阅读社区，支持实时预览，提供分享网址。
- Mou：一个 Mac OS X 上的 Markdown 编辑器。
- MarkdownPad：Windows 上的全功能 Markdown 编辑器。
- PageDown：一个 Javascript 写的 “WYSIWYM”（所见即所得）Markdown 编辑器 (来自 StackOverflow)
- IPython Notebook：以 IPython 为后台，利用浏览器做 IDE，支持 Markdown 与 LaTeX 公式。
- ……

Markdown | 编辑器 | 免费

- Windows 平台
  - MarkdownPad
  - MarkPad
- Linux 平台
  - Typora
  - Haroopad
  - Remarkable
  - ReText
  - MdCharm
- Mac 平台：Mou
- 在线编辑器
  - Markable.in
  - Dillinger.io
- 浏览器插件：MaDe (Chrome)
- 高级应用：Sublime Text 2 + MarkdownEditing

Vim 插件管理器

Vundle: the plug-in manager for Vim

Vim 的 Markdown 插件

- vim-markdown: Syntax highlighting, matching rules and mappings for the original Markdown and extensions.
- vim-instant-markdown: Instant Markdown previews from Vim!

Vim 插件管理器

Vundle: the plug-in manager for Vim

Vim 的 Markdown 插件

- vim-markdown: Syntax highlighting, matching rules and mappings for the original Markdown and extensions.
- vim-instant-markdown: Instant Markdown previews from Vim!

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

一个 Markdown 段落是由一个或多个连续的文本行组成，它的前后要有一个以上的空行（空行的定义是显示上看起来像是空的，便会被视为空行。比方说，若某一行只包含空格和制表符，则该行也会被视为空行）。普通段落不该用空格或制表符来缩进。

“由一个或多个连续的文本行组成”这句话其实暗示了 Markdown 允许段落内的强迫换行（插入换行符），这个特性和其他大部分的 text-to-HTML 格式不一样。

在文本中输入的换行会从最终生成的结果中删除，浏览器会根据可用空间自动换行。如果想强迫换行，可以在行尾插入至少两个空格。

Markdown 支持两种标题的语法，类 Setext 和类 atx 形式。

- 类 Setext 形式是用底线的形式，利用 = （最高阶标题）和 - （第二阶标题）。任何数量的 = 和 - 都可以有效果。
- 类 atx 形式则是在行首插入 1 到 6 个 #，对应到标题 1 到 6 阶。你可以选择性地“闭合”类 atx 样式的标题，这纯粹只是美观用的，若是觉得这样看起来比较舒适，你就可以在行尾加上 #，而行尾的 # 数量也不用和开头一样（行首的井字符数量决定标题的阶数）。

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

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

预览

This is an H1

This is an H2

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

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

预览

This is an H1
This is an H2

```
# This is an H1
## This is an H2
### This is an H3
#### This is an H4
##### This is an H5
###### This is an H6
```

预览

This is an H1
This is an H2
This is an H3
This is an H4
This is an H5
This is an H6

```
# This is an H1
## This is an H2
### This is an H3
#### This is an H4
##### This is an H5
###### This is an H6
```

预览

This is an H1
This is an H2
This is an H3
This is an H4
This is an H5
This is an H6

Markdown 标记区块引用是使用类似 email 中用 > 的引用方式，看起来像是你自己先断好行，然后在每行的最前面加上 >。

```
>半亩方塘一鉴开，天光云影共徘徊。
>问渠那得清如许？为有源头活水来。
```

Markdown 也允许你偷懒只在整个段落的第一行最前面加上 >。

```
1 >古之学者必有师。师者，所以传道受业解惑也。人非生而知
  之者，孰能无惑？惑而不从师，其为惑也，终不解矣。生乎
  吾前，其闻道也固先乎吾，吾从而师之；生乎吾後，其闻道
  也亦先乎吾，吾从而师之。吾师道也，夫庸知其年之先後生
  於吾乎！是故无贵无贱无长无少，道之所存，师之所存也。
2 >圣人无常师。孔子师郯子、苌子、师襄、老聃。郯子之徒，
  其贤不及孔子。孔子曰：“三人行，必有我师。”是故弟子
  不必不如师，师不必贤於弟子。闻道有先後，术业有专攻，
  如是而已。
```

区块引用可以嵌套（例如：引用内的引用），只要根据层次加上不同数量的 > 即可。

```
>圣人无常师。孔子师郯子、苌子、师襄、老聃。郯子之徒，其贤不及孔子。孔子曰：
>
>>三人行，必有我师。
>
>是故弟子不必不如师，师不必贤於弟子。闻道有先後，术业有专攻，如是而已。
```

引用的区块内也可以使用其他的 Markdown 语法，包括标题、列表、代码区块等。

```
>## 观书有感
>#### **朱熹**（*南宋*）
>半亩方塘一鉴开，天光云影共徘徊。
>问渠那得清如许？为有源头活水来。
```

列表

Markdown 支持无序列表和有序列表。

列表项目标记通常是放在最左边，但是其实也可以缩进，最多 3 个空格，项目标记后面则一定要接着至少一个空格或制表符。

无序列表

无序列表使用星号（*）、加号（+）或是减号（-）作为列表标记。

有序列表

有序列表则使用数字接着一个英文句点。（在列表标记上使用的数字并不会影响输出的 HTML 结果）

列表

Markdown 支持无序列表和有序列表。

列表项目标记通常是放在最左边，但是其实也可以缩进，最多 3 个空格，项目标记后面则一定要接着至少一个空格或制表符。

无序列表

无序列表使用星号（*）、加号（+）或是减号（-）作为列表标记。

有序列表

有序列表则使用数字接着一个英文句点。（在列表标记上使用的数字并不会影响输出的 HTML 结果）

列表

Markdown 支持无序列表和有序列表。

列表项目标记通常是放在最左边，但是其实也可以缩进，最多 3 个空格，项目标记后面则一定要接着至少一个空格或制表符。

无序列表

无序列表使用星号（*）、加号（+）或是减号（-）作为列表标记。

有序列表

有序列表则使用数字接着一个英文句点。（在列表标记上使用的数字并不会影响输出的 HTML 结果）

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

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

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

预览

- Red
- Green
- Blue

```
1. Red
2. Green
3. Blue
```

```
1. Red
1. Green
1. Blue
```

```
3. Red
1. Green
5. Blue
```

预览

1. Red
2. Green
3. Blue

列表项目可以包含多个段落，每个项目下的段落都必须缩进 4 个空格或是 1 个制表符。

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

    This is the second paragraph.

2.  This is another...
```

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

This is the second paragraph.

2.  This is another...
```

列表项目可以包含多个段落，每个项目下的段落都必须缩进 4 个空格或是 1 个制表符。

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

    This is the second paragraph.

2.  This is another...
```

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

This is the second paragraph.

2.  This is another...
```

如果要在列表项目内放进引用，那么 > 就需要缩进：

```
*   A list item with a blockquote:

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

如果要在列表中放代码区块的话，该区块就需要缩进两次，也就是 8 个空格或是 2 个制表符：

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

        #!/usr/bin/perl
        print "Hello, world!\n";
```

项目列表很可能会不小心产生，也就是在行首出现数字-句点-空白的时候：

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

要避免这样的状况，可以在句点前面加上反斜线：

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

项目列表很可能会不小心产生，也就是在行首出现数字-句点-空白的时候：

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

要避免这样的状况，可以在句点前面加上反斜线：

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

和程序相关的写作或是标签语言源代码通常会有已经排版好的代码区块，对于这些区块，通常我们并不希望它以一般段落文件的方式去排版，而是照原来的样子显示。

要在 Markdown 中建立代码区块很简单，只要简单地缩进 4 个空格或是 1 个制表符就可以：

```
A normal paragraph followed by a Perl script:

    #!/usr/bin/perl
    print "Hello, world!\n";
```

基本语法 | 区块元素 | 代码区块

- 每行一阶的缩进，即 4 个空格或是 1 个制表符，都会被移除。
- 一个代码区块会一直持续到没有缩进的那一行（或是文件结尾）。
- 在代码区块里面，&、< 和 > 会自动转成 HTML 实体，这样的方式让你非常容易使用 Markdown 插入范例用的 HTML 源代码——只需要复制贴上，再加上缩进就可以了——剩下的 Markdown 都会帮你处理。
- 代码区块中，一般的 Markdown 语法不会被转换，像是星号便只是星号，这表示你可以很容易地以 Markdown 语法撰写 Markdown 语法相关的文件。

基本语法 | 区块元素 | 引用 vs. 代码

```
>May the Force be with you!
```

```
#!/usr/bin/perl
print "Hello, world!\n";
```

```
>May the Force be with you!
```

```
    #!/usr/bin/perl
    print "Hello, world!\n";
```

你可以在一行中用三个以上的星号、减号、下划线来建立一个分隔线，行内不能有其他东西。你也可以在星号、减号或者下划线中间插入空格。

```
* * *

***

*****

- - -

---------------------------------------

___

_ _ _ _ _
```

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

Markdown 支持两种形式的链接语法：行内式和参考式两种形式。

不管是哪一种，链接文字都是用 [ ] （方括号）来标记。

要建立一个行内式的链接，只要在方块括号后面紧接着圆括号并插入网址链接即可，如果你还想要加上链接的 title 文字，只要在网址后面，用双引号把 title 文字包起来即可：

```
This is [an example](http://example.com/ "Title") inline link.

[This link](http://example.net/) has no title attribute.
```

如果你是要链接到同样主机的资源，可以使用相对路径：

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

要建立一个行内式的链接，只要在方块括号后面紧接着圆括号并插入网址链接即可，如果你还想要加上链接的 title 文字，只要在网址后面，用双引号把 title 文字包起来即可：

```
This is [an example](http://example.com/ "Title") inline link.

[This link](http://example.net/) has no title attribute.
```

如果你是要链接到同样主机的资源，可以使用相对路径：

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

参考式的链接是在链接文字的括号后面再接上另一个方括号，而在第二个方括号里面要填入用以辨识链接的标记。你也可以选择性地在两个方括号中间加上一个空格。

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

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

接着，在文件的任意处，你可以把这个标记的链接内容定义出来：

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

网址定义只有在产生链接的时候用到，并不会直接出现在文件之中。

参考式的链接是在链接文字的括号后面再接上另一个方括号，而在第二个方括号里面要填入用以辨识链接的标记。你也可以选择性地在两个方括号中间加上一个空格。

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

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

接着，在文件的任意处，你可以把这个标记的链接内容定义出来：

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

网址定义只有在产生链接的时候用到，并不会直接出现在文件之中。

- 方括号（前面可以选择性地加上至多三个空格来缩进），里面输入链接文字
- 接着一个冒号
- 接着一个以上的空格或制表符
- 接着链接的网址
- 选择性地接着 title 内容，可以用单引号、双引号或是小括号包着

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

链接网址也可以用尖括号包起来：

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

也可以把 title 属性放到下一行，也可以加一些缩进，若网址太长的话，这样会比较好看：

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

链接辨别标签可以有字母、数字、空白和标点符号，但是并不区分大小写，因此下面两个链接是一样的：

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

链接网址也可以用尖括号包起来：

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

也可以把 title 属性放到下一行，也可以加一些缩进，若网址太长的话，这样会比较好看：

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

链接辨别标签可以有字母、数字、空白和标点符号，但是并不区分大小写，因此下面两个链接是一样的：

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

链接网址也可以用尖括号包起来：

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

也可以把 title 属性放到下一行，也可以加一些缩进，若网址太长的话，这样会比较好看：

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

链接辨别标签可以有字母、数字、空白和标点符号，但是并不区分大小写，因此下面两个链接是一样的：

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

隐式链接标记功能让你可以省略指定链接标记，这种情形下，链接标记会视为等同于链接文字。要用隐式链接标记只要在链接文字后面加上一个空的方括号。比如要让“Google”链接到 google.com：

```
[Google][]
```

然后定义链接内容：

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

由于链接文字可能包含空白，所以这种简化型的标记内也许包含多个单词：

```
Visit [Daring Fireball][].
```

然后接着定义链接：

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

隐式链接标记功能让你可以省略指定链接标记，这种情形下，链接标记会视为等同于链接文字。要用隐式链接标记只要在链接文字后面加上一个空的方括号。比如要让“Google”链接到 google.com：

```
[Google][]
```

然后定义链接内容：

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

由于链接文字可能包含空白，所以这种简化型的标记内也许包含多个单词：

```
Visit [Daring Fireball][].
```

然后接着定义链接：

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

链接的定义可以放在文件中的任何一个地方，我比较偏好直接放在链接出现段落的后面，你也可以把它放在文件最后面，就像是注解一样。

参考式的链接其实重点不在于它比较好写，而是它比较好读。使用 Markdown 的参考式链接，可以让文件更像是浏览器最后产生的结果。通过把一些标记相关的元数据移到段落文字之外，就可以增加链接而不让文章的阅读感觉被打断。

```
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").
```

```
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"
```

```
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").
```

```
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"
```

```
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"
```

```
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"
```

```
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"
```

```
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"
```

Markdown 使用星号（*）和下划线（_）作为标记强调字词的符号。分为强调（斜体）、加重强调（粗体）和特别强调（粗斜体）。

```
*single asterisks*

_single underscores_

**double asterisks**

__double underscores__

***three asterisks***

___three underscores___
```

预览

single asterisks
single underscores
double asterisks
double underscores
three asterisks
three underscores

你可以随便用你喜欢的样式，唯一的限制是，你用什么符号开启标签，就要用什么符号结束。

Markdown 使用星号（*）和下划线（_）作为标记强调字词的符号。分为强调（斜体）、加重强调（粗体）和特别强调（粗斜体）。

```
*single asterisks*

_single underscores_

**double asterisks**

__double underscores__

***three asterisks***

___three underscores___
```

预览

single asterisks
single underscores
double asterisks
double underscores
three asterisks
three underscores

你可以随便用你喜欢的样式，唯一的限制是，你用什么符号开启标签，就要用什么符号结束。

强调也可以直接插在文字中间：

```
un*frigging*believable
```

但是如果你的* 和_ 两边都有空白的话，它们就只会被当成普通的符号。

如果要在文字前后直接插入普通的星号或下划线，你可以用反斜线：

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

强调也可以直接插在文字中间：

```
un*frigging*believable
```

但是如果你的* 和_ 两边都有空白的话，它们就只会被当成普通的符号。

如果要在文字前后直接插入普通的星号或下划线，你可以用反斜线：

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

强调也可以直接插在文字中间：

```
un*frigging*believable
```

但是如果你的* 和_ 两边都有空白的话，它们就只会被当成普通的符号。

如果要在文字前后直接插入普通的星号或下划线，你可以用反斜线：

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

如果要标记一小段行内代码，你可以用反引号把它包起来（`）：

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

如果要在代码区段内插入反引号，你可以用多个反引号来开启和结束代码区段：

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

代码区段的起始和结束端都可以放入一个空白，起始端后面一个，结束端前面一个，这样你就可以在区段的一开始就插入反引号：

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

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

如果要标记一小段行内代码，你可以用反引号把它包起来（`）：

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

如果要在代码区段内插入反引号，你可以用多个反引号来开启和结束代码区段：

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

代码区段的起始和结束端都可以放入一个空白，起始端后面一个，结束端前面一个，这样你就可以在区段的一开始就插入反引号：

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

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

如果要标记一小段行内代码，你可以用反引号把它包起来（`）：

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

如果要在代码区段内插入反引号，你可以用多个反引号来开启和结束代码区段：

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

代码区段的起始和结束端都可以放入一个空白，起始端后面一个，结束端前面一个，这样你就可以在区段的一开始就插入反引号：

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

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

Markdown 使用一种和链接很相似的语法来标记图片，同样也允许两种样式：行内式和参考式。

到目前为止，Markdown 还没有办法指定图片的宽高，如果你需要的话，你可以使用普通的 `<img>` 标签。

行内式的图片语法：

- 一个感叹号（!）
- 接着一个方括号，里面放上图片的替代文字
- 接着一个普通括号，里面放上图片的网址，最后还可以用引号包住并加上选择性的“title”文字。

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

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

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

“id”是图片参考的名称，图片参考的定义方式和链接参考一样：

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

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

Markdown 支持以比较简短的自动链接形式来处理网址和电子邮件信箱，只要是用尖括号包起来，Markdown 就会自动把它转成链接。一般网址的链接文字就和链接地址一样：

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

电子邮箱地址的自动链接也很类似，只是 Markdown 会先做一个编码转换的过程，把文字字符转成 16 进位码的 HTML 实体，这样的格式可以糊弄一些不好的邮址收集机器人：

```
<address@example.com>
```

（这种作法虽然可以糊弄不少的机器人，但并不能全部挡下来，不过总比什么都不做好些。不管怎样，公开你的信箱终究会引来广告信件的。）

Markdown 支持以比较简短的自动链接形式来处理网址和电子邮件信箱，只要是用尖括号包起来，Markdown 就会自动把它转成链接。一般网址的链接文字就和链接地址一样：

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

电子邮箱地址的自动链接也很类似，只是 Markdown 会先做一个编码转换的过程，把文字字符转成 16 进位码的 HTML 实体，这样的格式可以糊弄一些不好的邮址收集机器人：

```
<address@example.com>
```

（这种作法虽然可以糊弄不少的机器人，但并不能全部挡下来，不过总比什么都不做好些。不管怎样，公开你的信箱终究会引来广告信件的。）

Markdown 可以利用反斜线来插入一些在语法中有其它意义的符号，支持以下这些符号前面加上反斜线来帮助插入普通的符号：

```
\     反斜线
`     反引号
*     星号
_     下划线
{}    大括号
[]    中括号
()    小括号
#     井字号
+     加号
-     减号
.     英文句点
!     感叹号
```

使用这样的占位符号可以将脚注添加到文本中：`[^1]`。另外，你可以使用“n”而不是数字的`[^n]`。所以你可以不必担心使用哪个号码。在您的文章的结尾，你可以如下所示定义匹配的注脚，URL 将变成链接：

```
[^1]: This is my first footnote
[^n]: Visit http://ghost.org
[^n]: A final footnote
```

```
Footnotes[^1] have a label[^label] and a
  definition[^!DEF].

[^1]: This is a footnote
[^label]: A footnote on "label"
[^!DEF]: The definition of a footnote.
```

使用 HTML 语法来设定字体的颜色：

```
<font color="red">我是红色字体</font>

<font color="green">我是绿色字体</font>

<font color="blue">我是蓝色字体</font>

<font color="#FF0000">我是红色字体</font>

<font color="#00FF00">我是绿色字体</font>

<font color="#0000FF">我是蓝色字体</font>
```

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

扩展语法 | 杂项

```
# 目录
[toc]

# 元数据
---
title: 小书匠语法使用手册
tags: 小书匠,语法,MARKDOWN,帮助
---

# 扩展的文字格式
~~添加删除线~~
++插入的文字++
==被记号的文字==
上角文字: 19^th^
下角文字: H~2~O
```

扩展语法 | 杂项

```
# 印刷字替换
(c) (C) (r) (R) (tm) (TM) (p) (P) +-

# 缩写定义
The HTML specification is maintained by the W3C.
*[HTML]: Hyper Text Markup Language
*[W3C]:  World Wide Web Consortium
```

扩展语法 | 杂项

```
# 待办事项
[ ] 未完成事项
[-] 未完成事项
[x] 完成事项
[X] 完成事项

# 定义
苹果
: 一种水果
: 一种品牌，计算机，手持设备
桔子
: 一种水果
```

标准 Markdown 基于缩进代码行或者 4 个空格位:

```
    #!/usr/bin/perl
    print "Hello, world!\n";
```

GFM（GitHub Flavored Markdown）使用三个反引号```:

```
```perl
#!/usr/bin/perl
print "Hello, world!\n";
```
```

扩展语法 | 表格

```
Left align | Right align | Center align
:----------|------------:|:------------:
 This      |        This |     This
 column    |      column |    column
 will      |        will |     will
 be        |          be |      be
 left      |       right |    center
 aligned   |     aligned |   aligned

* Outer pipes on tables are optional
* Colon used for alignment (right versus left
  )
```

扩展语法 | 公式

```
基于LaTeX公式语法，可以创建行内公式，例如 $\Gamma(n) = (n-1)!\quad\forall n\in\mathbb N$。或者块级公式：

$$E=mc^2$$

$$ x = \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a} $$

$$
\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt\,.
$$
```

扩展语法 | 流程图

```
```flow
st=>start: Start
op=>operation: Your Operation
cond=>condition: Yes or No?
e=>end

st->op->cond
cond(yes)->e
cond(no)->op
```
```

扩展语法 | 序列图/时序图

```
```seq
Alice->Bob: Hello Bob, how are you?
Note right of Bob: Bob thinks
Bob-->Alice: I am good thanks!
```
```

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

语法参考卡片

Headers

```
# Header 1
## Header 2
### Header 3
```

Text Annotations

```
*This text will be emphasized.*
**This text will be strong.**
Inline `code` in typewriter style
```

Links

```
[GitHub](http://github.com)
```

Images

```
![Alt text](/images/logo.png)
```

Blockquotes

```
William Zinser:

> Writing is thinking on paper. (no-more)
```

Unordered Lists

```
* Item 1
* Item 2
  * Item 2a
  * Item 2b
```

Ordered Lists

```
1. Item 1
2. Item 2
   * Item 2a
   * Item 2b
```

Code Blocks

```
Use 4 spaces indentation:

    function identity(x) {
      return x;
    }
```

语法参考卡片

Markdown Cheat Sheet

Format Text

Headers

```
# This is an <h1> tag
## This is an <h2> tag
###### This is an <h6> tag
```

Text styles

```
*This text will be italic*
_This will also be italic_
**This text will be bold**
__This will also be bold__

*You **can** combine them*
```

Lists

Unordered

```
* Item 1
* Item 2
  * Item 2a
  * Item 2b
```

Ordered

```
1. Item 1
2. Item 2
3. Item 3
   * Item 3a
   * Item 3b
```

Miscellaneous

Images

```
![GitHub Logo](/images/logo.png)
Format: ![Alt Text](url)
```

Links

```
http://github.com - automatic!
[GitHub](http://github.com)
```

Blockquotes

```
As Kanye West said:

> We're living the future so
> the present is our past.
```

Code Examples in Markdown

Syntax highlighting with GFM

```javascript
function fancyAlert(arg) {
  if(arg) {
    $.facebox({div:'#foo'})
  }
}
```

Or, indent your code 4 spaces

```
Here is a Python code example
without syntax highlighting:

    def foo:
      if not bar:
        return true
```

Inline code for comments

```
I think you should use an
`<addr>` element here instead.
```

语法参考卡片

Syntax Guide

Phrase Emphasis

```
*italic*   **bold**
_italic_   __bold__
```

Links

Inline:

```
An [example](http://url.com/ "Title")
```

Reference-style labels (titles are optional):

```
An [example][id]. Then, anywhere
else in the doc, define the link:

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

Images

Inline (titles are optional):

```
![alt text](/path/img.jpg "Title")
```

Reference-style:

```
![alt text][id]

[id]: /url/to/img.jpg "Title"
```

Headers

Setext-style:

```
Header 1
========

Header 2
--------
```

atx-style (closing #'s are optional):

```
# Header 1 #

## Header 2 ##

###### Header 6
```

Code Spans

```
`<code>` spans are delimited
by backticks.

You can include literal backticks
like `` `this` ``.
```

Lists

Ordered, without paragraphs:

```
1.  Foo
2.  Bar
```

Unordered, with paragraphs:

```
*   A list item.

    With multiple paragraphs.

*   Bar
```

You can nest them:

```
*   Abacus
    * answer
*   Bubbles
    1.  bunk
    2.  bupkis
        * BELITTLER
    3. burper
*   Cunning
```

Blockquotes

```
> Email-style angle brackets
> are used for blockquotes.

> > And, they can be nested.

> #### Headers in blockquotes
>
> * You can quote a list.
> * Etc.
```

Preformatted Code Blocks

Indent every line of a code block by at least 4 spaces or 1 tab.

```
This is a normal paragraph.

    This is a preformatted
    code block.
```

Horizontal Rules

Three or more dashes or asterisks:

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

语法参考卡片

Markdown Syntax

dashkards
http://dashkards.com/

Headers

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

This is an H2

This is an H1

This is an H2

This is an H6

Close headers with as many hashes as you like

General Formatting

Emphasis _Emphasis_

Strong __Strong__

Insert <br /> space space ↩

Blockquote >

Use multiple > for nested blockquotes

Horizontal Rule ***|* * *|---|- - -

Links

Inline Link [Anchor Text](URL "Title")

Reference Link [Anchor Text][ID]

Omit ID for implicit reference, i.e. reference ID=anchor text

Reference Definition [ID]: URL "Title"

Case-insensitive alphanumeric characters, spaces and punctuation for ID

Automatic Link <http://www.example.com/>

<mail@example.com>

Lists

Begin and end lists with a blank line.

Unordered List *|+|-

Ordered List n.(space(s)|tab)

n can be any number, start list with 1.

Rules: Indent paragraphs by 4 spaces or 1 tab, code blocks by 8 spaces or 2 tabs, escape a period following a number at beginning of line (1986\. What a great season.)

Images

Inline ![Alt Text](URL "Title")

Reference ![Alt Text][ID]

Reference Definition [ID]: URL "Title"

Code

Code Block indent by 4 spaces or 1 tab

& and <> automatically converted to HTML entities

Inline Code `code`

Use `` for literal backticks

Markdown v 1.0.1 • Copyright © 2004, John Gruber • http://daringfireball.net/

Backslash-escapable characters: \ ` * _ { } [ ] ( ) # + - . !

语法参考卡片

syntax

```
Plain text
End a line with two spaces to start a new paragraph.
*italics* and _italics_
**bold** and __bold__
superscript^2^
~~strikethrough~~
[link](www.rstudio.com)

# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6

endash: --
emdash: ---
ellipsis: ...
inline equation: $A = \pi*r^{2}$
image: ![](path/to/smallorb.png)

horizontal rule (or slide break):

***

> block quote

* unordered list
* item 2
    + sub-item 1
    + sub-item 2

1. ordered list
2. item 2
    + sub-item 1
    + sub-item 2

Table Header   | Second Header
-------------- | -------------
Table Cell     | Cell 2
Cell 3         | Cell 4
```

becomes

Plain text
End a line with two spaces to start a new paragraph.
italics and *italics*
bold and **bold**
superscript^2
~~strikethrough~~
link

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

endash: –
emdash: —
ellipsis: …
inline equation: $A = \pi * r^2$
image:
horizontal rule (or slide break):

> block quote

- unordered list
- item 2
  - sub-item 1
  - sub-item 2

1. ordered list
2. item 2
   - sub-item 1
   - sub-item 2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

可重复性研究

可重复性研究（Reproducible Research）

- 代码、数据、结果集成在一起。
- RStudio (R + Markdown + knitr)

参考链接

- 自动化报告
- knitr 与可重复的统计研究（花絮篇）

可重复性研究

可重复性研究（Reproducible Research）

- 代码、数据、结果集成在一起。
- RStudio (R + Markdown + knitr)

参考链接

- 自动化报告
- knitr 与可重复的统计研究（花絮篇）

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

格式转换 | pandoc | 简介

pandoc 是由 John MacFarlane 开发的标记语言转换工具，可实现不同标记语言间的格式转换，堪称该领域中的“瑞士军刀”。

pandoc 使用 Haskell 语言编写，以命令行形式实现与用户的交互，可支持多种操作系统。

pandoc 采用 GNU GPL 授权协议发布，属于自由软件。

格式转换 | pandoc | 支持格式

Pandoc可读取的源格式	Pandoc可生成的目标格式
• Markdown • reStructuredText • textile • HTML • DocBook • LaTeX • MediaWiki标记语言 • OPML • Org-Mode • Haddock	• HTML格式：包括XHTML，HTML5及HTML slide • 文字处理软件格式：包括docx、odt、OpenDocument XML • 电子书格式：包括EPUB（第2版及第3版）、FictionBook2 • 技术文档格式：包括DocBook、GNU TexInfo、Groff manpages、Haddock • 页面布局格式：InDesign ICML • 大纲处理标记语言格式：OPML • TeX格式：包括LaTeX、ConTeXt、LaTeX Beamer • PDF格式：需要LaTeX支持 • 轻量级标记语言格式：包括Markdown、reStructuredText、textile、Org-Mode、MediaWiki标记语言、AsciiDoc • 自定义格式：可使用lua自定义转换规则

格式转换 | pandoc | 使用

```
# 安装
sudo apt install pandoc
#sudo apt-get install pandoc

# 使用帮助
man pandoc
pandoc -h

# 命令语法
pandoc [options] [input-file] ...
```

格式转换 | pandoc | 使用

```
# Pandoc会根据文件的后缀名自动判断格式
pandoc -o output.html input.md

# 用户也可以显式地指定输入文件和输出文件格式
pandoc -f markdown -t html -o output.html input.md

# Markdown转HTML
pandoc input.md -o output.html -c Github.css
# Markdown转word
pandoc input.md -o output.docx -c Github.css
# Markdown转PDF（需要LaTeX）
pandoc input.md -o output.pdf --latex-engine=xelatex -V mainfont="SimSun" --template=template.latex
```

格式转换 | unoconv | 简介

unoconv

Universal Office Converter (unoconv) is a command line tool to convert any document format that **LibreOffice** can import to any document format that LibreOffice can export. It makes use of the LibreOffice′s UNO bindings for non-interactive conversion of documents.

使用

```
# odt to *
unoconv -f pdf file.odt
unoconv -f doc file.odt
unoconv -f html file.odt
```

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

格式转换 | ImageMagick | 简介

ImageMagick 是一个用于查看、编辑位图文件以及进行图像格式转换的开放源代码软件套装。它可以读取、编辑超过 200 种图象格式。ImageMagick 以 ImageMagick 许可证（一个类似 BSD 的许可证）发布。

ImageMagick 主要由大量的命令行程序组成，而不提供像 Adobe Photoshop、GIMP 这样的图形界面。但是，ImageMagick 也提供了一个基于 X Window 的简易 GUI：IMDisplay。

ImageMagick 为很多程序语言提供了 API 库。在 Perl 语言中，ImageMagick 的 API 叫做 PerlMagick。

格式转换 | ImageMagick | 功能

ImageMagick 最基本的一个功能是准确高效地转换超过 68 种图片的格式，包括众所周知的 TIFF、JPEG、PNG、PDF、PhotoCD，以及 GIF。Imagemagick 使用特征签名识别文件类型。

ImageMagick 包括了大量用于特效的滤镜和扩展功能。

Use ImageMagick to create, edit, compose, or convert bitmap images. It can read and write images in a variety of formats (over 200) including PNG, JPEG, JPEG-2000, GIF, TIFF, DPX, EXR, WebP, Postscript, PDF, and SVG. Use ImageMagick to resize, flip, mirror, rotate, distort, shear and transform images, adjust image colors, apply various special effects, or draw text, lines, polygons, ellipses and Bézier curves.

格式转换 | ImageMagick | 使用

convert

Use the convert program to convert between image formats as well as resize an image, blur, crop, despeckle, dither, draw on, flip, join, re-sample, and much more.

```
# 安装
sudo apt install imagemagick
man convert
convert -h

# convert an image in the JPEG format to PNG
convert rose.jpg rose.png
# reduce the image size before it is written to the PNG format
convert rose.jpg -resize 50% rose.png
```

格式转换 | ImageMagick | 使用

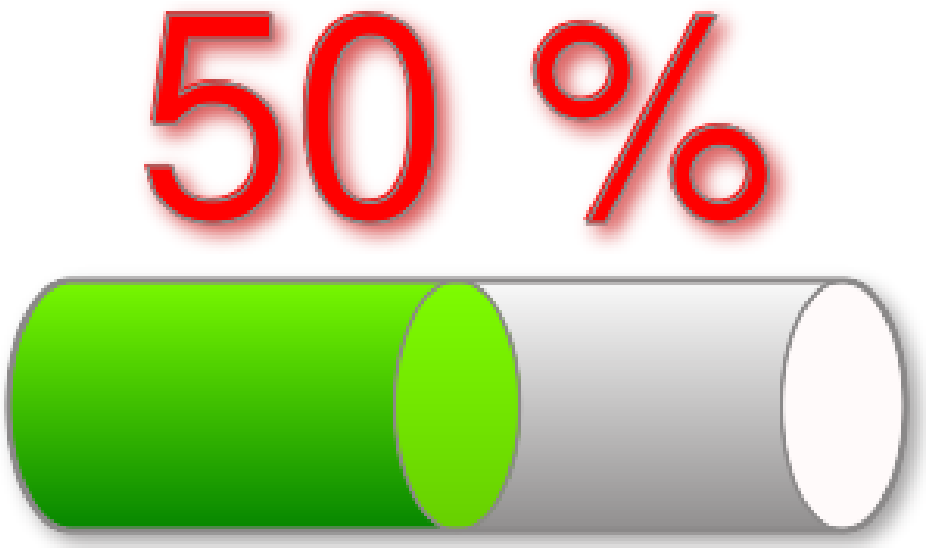
```
# combine multiple image-processing operations to produce complex results
convert -size 320x85 canvas:none -font Bookman-DemiItalic -pointsize 72 -draw "text 25,60 \'Magick\'" -channel RGBA -blur 0x6 -fill darkred -stroke magenta -draw "text 20,55 \'Magick\'" fuzzy-magick.png

# resize an image with improved quality
convert input.png -colorspace RGB +sigmoidal-contrast 11.6933 -define filter:filter=Sinc -define filter:window=Jinc -define filter:lobes=3 -resize 400% -sigmoidal-contrast 11.6933 -colorspace sRGB output.png
```

格式转换 | ImageMagick | 实例

```
convert label.gif +matte \( +clone -shade 110x90 -normalize -negate +clone -compose Plus -composite \) \( -clone 0 -shade 110x50 -normalize -channel BG -fx 0 +channel -matte \) -delete 0 +swap -compose Multiply -composite button.gif
```

格式转换 | ImageMagick | 实例

```
convert -size 320x90 canvas:none -stroke snow4 -size 1x90 -tile gradient:white-snow4 -
  draw 'roundrectangle 16, 5, 304, 85 20,40' +tile -fill snow -draw 'roundrectangle 264,
   5, 304, 85  20,40' -tile gradient:chartreuse-green -draw 'roundrectangle 16, 5, 180,
  85  20,40' -tile gradient:chartreuse1-chartreuse3 -draw 'roundrectangle 140, 5, 180,
  85  20,40' +tile -fill none -draw 'roundrectangle 264, 5, 304, 85 20,40' -strokewidth
  2 -draw 'roundrectangle 16, 5, 304, 85 20,40' \( +clone -background snow4 -shadow 80x3
  +3+3 \) +swap -background none -layers merge \( +size -font Helvetica -pointsize 90 -
  strokewidth 1 -fill red label:'50 %' -trim +repage \( +clone -background firebrick3 -
  shadow 80x3+3+3 \) +swap -background none -layers merge \) -insert 0 -gravity center -
  append -background white -gravity center -extent 320x200 cylinder_shaded.png
```



教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

格式转化 | FFmpeg | 简介

FFmpeg 是一个自由软件，可以运行音频和视频多种格式的录影、转换、流功能，包含了 libavcodec——这是一个用于多个项目中音频和视频的解码器库，以及 libavformat——一个音频与视频格式转换库。

组件

ffmpeg	一个命令行工具，用来对视频文件转换格式，也支持对电视卡即时编码
ffserver	一个 HTTP 多媒体即时广播流服务器，支持时光平移
ffplay	一个简单的播放器，基于 SDL 与 FFmpeg 库
libavcodec	包含全部 FFmpeg 音频/视频编解码库
libavformat	包含 demuxers 和 muxer 库
libavutil	包含一些工具库
libpostproc	对于视频做前处理的库
libswscale	对于视频作缩放的库

格式转化 | FFmpeg | 使用

```
# 安装
sudo apt install ffmpeg
# 帮助与手册
man ffmpeg
ffmpeg -h
ffmpeg -formats

# convert a file with the default settings
ffmpeg -i input.avi output.mpg
```

格式转化 | FFmpeg | 使用

```
# encode a video with a 128kbps audio bitrate and 1,200kbps video stream
ffmpeg -i input.avi -ab 128 -b 1200 output.mpg

# convert a WAV file to a 128kbps MP3 file
ffmpeg -i input.wav -ab 128 output.mp3

# encode using the MPEG-4 codec at 1,200kbps video bitrate and 128kbps audio bitrate
ffmpeg -i input.mpg -ab 128 -b 1200 -vcodec mpeg4 output.avi
```

格式转化 | FFmpeg | 使用

```
# Convert a video into mp3 format
ffmpeg -i video.flv -vn -ar 44100 -ac 2 -ab 192 -f mp3 audio.mp3

# Convert video from one format to another
ffmpeg -i video.flv video.mpg
ffmpeg -i video.mpg -ab 26k -f flv video1.flv
ffmpeg -i youtube.flv -c:v libx264 filename.mp4
ffmpeg -i video.wmv -c:v libx264 -preset ultrafast video.mp4

# Convert a video to CD or DVD format
ffmpeg -i video.mpg -target vcd vcd_video.mpg
ffmpeg -i video.avi -target pal-dvd -ps 2000000000 -aspect 16:9 video.mpeg
```

格式转化 | FFmpeg | 使用

```
# Get Video File Information
ffmpeg -i video.flv -hide_banner

# Compare/Test Video and Audio Quality
ffplay video1.mp4
ffplay audio_filename1.mp3

# Cut video file into a smaller clip
ffmpeg -i input.mp4 -ss 00:00:50.0 -codec copy -t 20 output.mp4
ffmpeg -ss 00:01:30 -t 30 -acodec copy -i input.mp3 output.mp3

# Split a video into multiple parts
ffmpeg -i video.mp4 -t 00:00:50 -c copy small-1.mp4 -ss 00:00:50 -codec copy small-2.mp4

# Join (concatenate) video files
ffmpeg -f concat -i file-list.txt -c copy output.mp4
```

格式转化 | FFmpeg | 使用

```
# Resize a video
ffmpeg -i input.mp4 -s 480x320 -c:a copy output.mp4

# Change the audio volume
ffmpeg -i input.wav -af 'volume=0.5' output.wav

# Add subtitles to a movie
ffmpeg -i movie.mp4 -i subtitles.srt -map 0 -map 1 -c copy -c:v libx264 -crf 23 -preset veryfast output.mkv

# Rotate a video
ffmpeg -i input.mp4 -filter:v 'transpose=1' rotated-video.mp4
ffmpeg -i input.mp4 -filter:v 'transpose=2,transpose=2' rotated-video.mp4
```

格式转化 | FFmpeg | 使用

```
# Mute a video (Remove the audio component)
ffmpeg -i video.mp4 -an mute-video.mp4
# Extract audio (Remove the video component)
ffmpeg -i video.mp4 -vn -ab 256 audio.mp3
ffmpeg -i video1.avi -vn -ar 44100 -ac 2 -ab 192 -f mp3 audio3.mp3

# Merge an audio and video file
ffmpeg -i audio.mp3 -i video.avi video_audio_mix.mpg
ffmpeg -i video.mp4 -i audio.mp3 -c:v copy -c:a aac -strict experimental output.mp4

# Speed up or Slow down the audio
ffmpeg -i input.mkv -filter:a "atempo=2.0" -vn output.mkv
# Speed up or Slow down the video
ffmpeg -i video.mpg -vf "setpts=0.5*PTS" highspeed.mpg
ffmpeg -i video.mpg -vf "setpts=4.0*PTS" lowerspeed.mpg -hide_banner
```

格式转化 | FFmpeg | 使用

```
# Convert a video into animated GIF
ffmpeg -i video.flv animated.gif.mp4
ffmpeg -i video.mp4 -vf scale=500:-1 -t 10 -r 10 image.gif
# Extract image frames from a video
ffmpeg -ss 00:00:15 -i video.mp4 -vf scale=800:-1 -vframes 1 image.jpg
# Convert Video into Images
ffmpeg -i video.flv image%d.jpg
ffmpeg -i movie.mp4 -r 0.25 frames_%04d.png
# Create video slideshow from images
ffmpeg -f image2 -i image%d.jpg imagestovideo.mpg
ffmpeg -r 1/5 -i img%03d.png -c:v libx264 -r 30 -pix_fmt yuv420p slideshow.mp4
# Add a poster image or banner to audio
ffmpeg -loop 1 -i image.jpg -i audio.mp3 -c:v libx264 -c:a aac -strict experimental -b:a 192k -shortest output.mp4
# Convert a single image into a video
ffmpeg -loop 1 -i image.png -c:v libx264 -t 30 -pix_fmt yuv420p video.mp4
```

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

格式转换 | 三合一 | FF Multi Converter

FF Multi Converter

FF Multi Converter is a simple graphical application for Linux which enables you to convert audio, video, image and document files between all popular formats, by utilizing other command-line tools. It uses **FFmpeg** for audio/video files, the **ImageMagick** software suite for image conversions and **unoconv** for document files.

Homepage

- FF Multi Converter@Google
- FF Multi Converter@GitHub
- FF Multi Converter@SourceForge

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

Markdown | 总结

知识点

- Markdown 的基本语法：标题、引用、列表、代码、链接、强调……
- Markdown 相关：knitr，pandoc
- 格式转换：pandoc，unoconv，ImageMagick，FFmpeg

技能

- 熟练使用 Markdown 撰写文档
- 能够使用 pandoc 转换文档格式
- 能够使用 ImageMagick 转换图片格式
- 能够使用 FFmpeg 转换视频格式

教学提纲

1. Markdown 简介
2. 基本语法
   - 区块元素
   - 区段元素
   - 其他
3. 扩展语法
4. 语法参考卡片
5. 可重复性研究
6. 格式转换
   - 文档
   - 图片
   - 音/视频
   - 三合一
7. 回顾与总结
   - 总结
   - 思考题

Markdown | 思考题

1. 常见的文档样式（标题、强调等）如何使用 Markdown 语法来实现?
2. 如何把 Markdown 文件转换成其他常见的文档格式?
3. 如何转换图片和视频格式?

下节预告

回顾 shell 编程的基础知识：

- 变量
- 操作符
- 条件流程控制
- 迭代流程控制
- 语法检查与调试
- ……

Powered by

open source™

GNOME™

TEX LATEX XETEX Beamer