

Sử dụng lớp powerdot tạo trình diễn *

Hendri Adriaens Christopher Ellison

Biên dịch: kyanh <<mailto:kyanh@o2.pl>>

Bản dịch số 329

*Lớp này có thể tải về tại CTAN:/macros/latex/contrib/powerdot. Xem powerdot.dtx để biết thông tin về bản quyền của lớp powerdot.

License & Copyright Information

This is the file 'powerdot-doc-vi(-print).pdf'
which is generated from the source 'powerdot-doc-vi.tex'.

Copyright (C) 2005 kyanh <kyanh at o2 dot pl>

This is a compiled work as described in
<http://www.latex-project.org/lppl.txt>

This compiled work may be distributed and/or modified under the
conditions of the LaTeX Project Public License, either version 1.3
of this license or (at your option) any later version.

The latest version of this license is in

<http://www.latex-project.org/lppl.txt>
and version 1.3 or later is part of all distributions of LaTeX
version 2003/12/01 or later.

This work has the LPPL maintenance status "maintained".

Current maintainer of this work is kyanh <kyanh at o2 dot pl>.

The following files constitute the 'powerdot-doc-vi' bundle
and ***MUST*** be distributed as a whole

README

powerdot-doc-vi.pdf

powerdot-doc-vi-print.pdf

img/

lst-bookmarks.png

powerdot-ciment.png

powerdot-default.png

powerdot-fyma.png

powerdot-ikeda.png

powerdot-simple.png

powerdot-tyjca.png

tab-contents.png

tab-slide-contents.png

exa/

README

example-1.tex

example-1.pdf

Tóm tắt nội dung

powerdot là lớp L^AT_EX cho phép tạo trình diễn nhanh chóng, chuyên nghiệp với kiểu dáng dễ dàng thay đổi. Lớp cung cấp nhiều công cụ giúp thiết kế trình diễn: overlay, ghi chú cá nhân, chế độ handout. Để xem trình diễn, bản DVI, PS, PDF của tài liệu đều có thể dùng được. Lớp cung cấp hệ thống mẫu mạnh, giúp phát triển dễ dàng các mẫu mới.

Mục lục

Acknowledgements	4	5.2 Tạo slide Mục lục	16
Ghi chú của kyanh	4	6 Linh tinh	17
1 Giới thiệu	4	6.1 Lệnh \twocolumn chia cột	17
2 Thiết lập cho trình diễn	5	6.2 Ghi chú	19
2.1 Tùy chọn cho lớp	5	6.3 Slide trắng	19
2.2 Thiết lập trình diễn	8	6.4 Slide tài liệu tham khảo	20
3 Tạo trình diễn	10	7 Các kiểu trình diễn	20
3.1 Trang tiêu đề	10	8 Biên dịch. Cài đặt. Xem	21
3.2 Tạo slide	10	8.1 Gói phụ thuộc	21
4 Overlay	11	8.2 Cài đặt	21
4.1 Lệnh \pause	11	8.3 Biên dịch và xem kết quả	22
4.2 Tạo danh sách	12	9 Tạo kiểu mới	23
4.3 Lệnh \item	13	10 Hỏi-Đáp	23
4.4 Lệnh \onslide	13	10.1 Câu hỏi thường gặp	23
4.5 Overlay tương đối	15	10.2 Mailinglist	24
5 Cấu trúc trình diễn	15	11 Mã nguồn powerdot	24
5.1 Tạo mục	15	Tài liệu	25

Acknowledgements

The authors are grateful to Darren Dale and Herbert Voß for testing the class and its output. Further, we like to thank all style contributors (see section 7) and the testers of the first beta releases.

Ghi chú của kyanh

Tài liệu này được biên dịch từ tài liệu chính thức của lớp `powerdot`. Có nhiều phần được thêm vào, một số phần khác được lược bớt. Riêng Mục 7 ở trang 20 chưa được dịch vì chưa cần thiết lắm cho người dùng bình thường.

Nếu bạn muốn tin tài liệu này, hãy sử dụng tập tin `powerdot-doc-vi-print.pdf`.

Là người dùng Việt, bạn có thể tìm kiếm sự giúp đỡ tại <http://www.vietttug.org/>.

Tài liệu này (bản PDF) được phân phối cùng với các ví dụ và hình ảnh như sau đây. Bạn có thể tải về bản đầy đủ của tài liệu ở <http://download.vietttug.org/>.

`img/`

`lst-bookmarks.png` Danh sách `bookmark` của một trình diễn

`tab-contents.png` Bảng Mục lục ở một `slide`

`tab-slide-contents.png` Slide Mục lục và Bảng Mục lục

`powerdot-default.png` Kiểu trình diễn mặc định

`powerdot-simple.png` Kiểu trình diễn `simple`

`powerdot-tyjca.png` Kiểu trình diễn `tyjca`

`powerdot-ikeda.png` Kiểu trình diễn `ikeda`

`powerdot-fyma.png` Kiểu trình diễn `fyma`

`powerdot-ciment.png` Kiểu trình diễn `ciment`

`exa/`

`example-1.tex` Ví dụ đơn giản về `powerdot`.

1 Giới thiệu

Lớp `powerdot` cung cấp khả năng tạo trình diễn dễ dàng, chuyên nghiệp. Lớp được thiết kế để việc thiết kế trình diễn trở nên đơn giản nhất có thể, nhờ đó bạn không phải tốn thời gian với các yếu tố kỹ thuật. Tất nhiên, bạn phải có các kiến thức cơ bản về `LATEX`.

Lớp được xây dựng nhờ mở rộng lớp `prosper` [8] và gói `HA-prosper` [2]. Gói `HA-prosper` có mục đích ban đầu là mở rộng và khắc phục vài lỗi, nhược điểm của lớp `prosper`. Thật không may là, không phải mọi nhược điểm của `prosper` đều có thể khắc phục được. Chính vì lý do này, một dự án mới ra đời nhằm thay thế cho cả `prosper` và `HA-prosper`. Bạn đang đọc tài liệu về chính dự án đó ;)

Phần còn lại của mục này giúp bạn có cái nhìn tổng quan về lớp `powerdot` và tài liệu hướng dẫn này.

Cấu trúc của trình diễn luôn tương tự như ví dụ sau đây:

```

\documentclass[<class options>]{powerdot}
\pdsetup{<pd options>}
\begin{document}
  \begin{slide}{slide}
    noi dung
  \end{slide}
  \section{section}
  \begin{slide}[<slide options>]{slide}
    noi dung
  \end{slide}
  \begin{note}{ghi chu ca nhan}
    ghi chu
  \end{note}
\end{document}

```

Có vài yếu tố tạo nên cấu trúc đó. Đầu tiên, lớp chấp nhận vài tùy chọn (`class options`) cho phép điều khiển kết quả xuất ví dụ, cỡ giấy, kiểu. Các tùy chọn này được bàn kỹ đến trong Mục 2.1. Thứ đến, là các tùy chọn trình diễn (`pd options`) điều khiển toàn cục các tính chất của trình diễn, ví dụ, các ghi chú ở chân trang. Những tùy chọn này được nói đến ở Mục 2.2 ở trang 8.

Sau khi thiết lập với các tùy chọn, bạn có thể dùng môi trường `slide` để tạo các trang (`slide`) trình diễn (xem Mục 3 ở trang 10) và môi trường `note` để tạo các ghi chú đi cùng với `slide` (xem Mục 6.2 ở trang 19). Bạn có thể dùng `overlay` để thể hiện nội dung theo từng bước (xem Mục 4 ở trang 11). Lệnh `\section` giúp bạn tạo cấu trúc cho trình diễn, giống như việc tạo chương, mục với tài liệu \LaTeX thông thường (xem Mục 5 ở trang 15). Bạn cũng có thể lựa chọn các kiểu dáng của trình diễn sau khi xem Mục 7 ở trang 20. Cuối cùng, với Mục 8 ở trang 21, bạn sẽ biết cách biên dịch tài liệu nguồn để có kết quả là trình diễn thật sự. Ở mục này cũng có vài lưu ý về việc cài đặt lớp `powerdot`.

Mục 9 ở trang 23 có lẽ là phần hấp dẫn đối với ai quan tâm đến việc tạo kiểu dáng riêng cho trình diễn của mình, hoặc làm đẹp các kiểu dáng đã có.

Tài liệu này kết thúc với Hỏi-Đáp (Mục 10 ở trang 23), có thể giúp bạn trong những bước đầu làm quen với lớp `powerdot`.

2 Thiết lập cho trình diễn

Mục này mô tả các tùy chọn để điều khiển kết quả xuất của trình diễn.

2.1 Tùy chọn cho lớp

Các tùy chọn được cho khi gọi lệnh `\documentclass`, cách nhau bởi dấu phẩy. Với mỗi tùy chọn dưới đây, giá trị mặc định được nêu trong mô tả – đó là giá trị sẽ được dùng khi bạn không nêu ra tùy chọn khi gọi lệnh `\documentclass`, hoặc khi bạn nêu tùy chọn đó nhưng không đi kèm giá trị nào.

`mode` Tùy chọn này xác định loại kết quả xuất, mặc định là **present**.

`mode=present`

Dùng **present** nếu bạn muốn tạo trình diễn thật sự (thay vì chỉ để kiểm tra trên màn hình). Với tùy chọn này, bạn có thể dùng `overlay` hoặc các hiệu ứng biến đổi. Đọc thêm ở Mục 4 ở trang 11 về `overlay`.

`mode=print`

Dùng **print** nếu bạn muốn in trình diễn của mình – với bản in, hiển nhiên các hiệu ứng hoặc `overlay` sẽ bị bỏ qua.

`mode=handout`

Dùng **handout** để có được bản xem qua (**overview**) trình diễn, với hai màu đen và trắng.

Bản này thích hợp cho mục đích cá nhân, để phân phối cho sinh viên, để minh họa trong khi bạn đang thuyết trình,...

`nopagebreaks`

Theo mặc định, với bản **handout**, mỗi trang sẽ bố trí hai **slide**. Nếu bạn muốn có nhiều hơn hai **slide** trên một trang, bạn hãy dùng tùy chọn này (tùy chọn này không có giá trị) khi gọi `\documentclass`, nhờ đó **powerdot** giúp **L^AT_EX** lựa chọn cách ngắt trang thích hợp.

paper Tùy chọn này có thể nhận ba giá trị sau. Mặc định là **screen**.

`paper=screen`

Trang trình diễn có tỷ lệ thông thường của màn hình (4/3). Kích thước trang thực sự là 8.25 inch – 11 inch. Kiểu này không phù hợp với **mode=print** hoặc **mode=handout**. Nếu bạn cố dùng **paper=screen** cùng với một trong **mode** vừa nói, **powerdot** sẽ tự động chuyển qua kiểu trang a4 kèm với lời nhắc nhở.

`paper=a4paper`

Trang trình diễn bằng khổ a4, dùng với **mode=present** hoặc **mode=handout**.

`paper=letterpaper`

Trang trình diễn kiểu **letter**

Vài thông tin quan trọng về cỡ trang, biên dịch và xem trình diễn được cho ở Mục 8 ở trang 21.

orient Tùy chọn này điều khiển hướng của trang, mặc định là **landscape** (nằm ngang).

`orient=landscape`

Trang trình diễn nằm ngang. Giá trị này không thích hợp với **mode=handout**. Nếu cố dùng **mode** đó với tùy chọn **landscape**, **powerdot** sẽ tự động chuyển qua kiểu **portrait** kèm theo lời nhắc nhở.

`orient=portrait`

Bố trí trang theo chiều đứng. Chú ý rằng không phải mọi kiểu trình diễn đều hỗ trợ chiều này. Vui lòng xem Mục 7 ở trang 20 để biết thêm chi tiết.

display Điều khiển việc xuất ra các trang trình diễn (**slide**) và ghi chú cá nhân (**note**). Mặc định là **slides**.

`display=slides`

Chỉ trình bày các trang trình diễn.

`display=slidesnotes`

Trình bày đồng thời trang trình diễn và ghi chú cá nhân. Xem thêm ở Mục 6.2 ở trang 19.

`display=notes`

Chỉ xuất ra kết quả gồm các ghi chú cá nhân. Để có thể cho đúng số trang, bạn cần phải biên dịch ở chế độ **slidesnotes** trước.

Dưới đây là vài tùy chọn khác điều khiển kết quả xuất.

size `size`

Xác định cỡ chữ (văn bản). Các giá trị có thể là 8, 9, 10, 11, 12, 14, 17, 20. Giá trị mặc định là 11.¹

style `style`

Xác định kiểu trình diễn, mặc định là `default`. Xem thêm Mục 7 ở trang 20 để biết thêm chi tiết về các kiểu.

fleqn `fleqn`

Đánh chỉ số cho phương trình ở bên trái.

leqno `leqno`

Đánh chỉ số cho phương trình ở bên phải.

nopsheader `nopsheader`

Theo mặc định, `powerdot` sẽ ghi các lệnh `postscript` vào kết quả xuất dạng PS, nhờ đó các chương trình tương tự `ps2pdf` khi chuyển kết quả xuất PS sang dạng PDF sẽ xác định đúng kiểu trang – bạn không cần phải chỉ ra khi gọi các chương trình đó ở dòng lệnh. Xem thêm Mục 8 ở trang 21. Nếu bạn gặp trở ngại khi chuyển sang dạng PDF với các chương trình đó, hoặc khi in ấn, bạn có thể dùng tùy chọn này – và khi đó bạn nhớ chỉ ra kiểu giấy khi gọi các chương trình chuyển.

hlentries `hlentries`

Ở Bảng Mục lục,² mỗi thành phần sẽ được tô màu nổi bật (`highlight`) nếu nó tương ứng với `slide` hiện tại. Nếu bạn không muốn điều này, hãy dùng `hlentries=false`. Mặc định là `hlentries=true`. Xem thêm ở Mục 5 ở trang 15.

hlsections `hlsections`

Tương tự như trên, nhưng là các thành phần tương đương với `section`. Giá trị mặc định là `false`. Giá trị `true` có ích khi bạn dùng các kiểu trình diễn chia nhỏ bản Mục lục.

blackslide `blackslide`

Với tùy chọn này, một trang `slide` đen sẽ chen vào đầu trình diễn (như vậy `slide` đen ở trang 1); trình diễn của bạn sẽ tự động nhảy qua trang 2 khi xem trong chương trình xem PDF (ví dụ Acrobat Reader). Trang `slide` đen đã được nhúng trước một đích gọi là `blackslide` nhờ đó bạn có thể thêm ở trang `slide` khác, chẳng hạn liên kết sau

```
\hyperlink{blackslide}{Click here to go to the black slide}
```

Khi bạn `click` ở bất kỳ vị trí nào của `slide` đen, bạn sẽ đi về đúng trang `slide` ban đầu. Tùy chọn này có ích, chẳng hạn khi bạn muốn tạm ngưng trình diễn để thuyết minh trên bảng đen.

Dưới đây là vài ví dụ giúp bạn hình dung lại các điều đã nói.

```
\documentclass[
  size=12,
  paper=screen,
  mode=present,
  display=slidesnotes,
  style=tycja,
```

¹Chú ý rằng, các cỡ không phải là 10, 11 hay 12 không phải là kiểu chuẩn – nên nếu dùng, trên hệ thống của bạn phải có cài đặt gói `extsizes` [10].

²Liệt kê các mục và các slide của trình diễn; bảng Mục lục xuất hiện ở mọi slide, trừ vài trường hợp đặc biệt.

```
nopagebreaks,
blackslide,
fleqn
]{powerdot}
```

Với ví dụ này, trình diễn sẽ dùng kiểu `tycja`, với trang `slide` đen, cỡ chữ 12 điểm, các phương trình được đánh số bên trái.

```
\documentclass[
size=12,
paper=letterpaper,
mode=handout,
display=slidesnotes,
style=tycja,
nopagebreaks,
blackslide,
fleqn
]{powerdot}
```

Ở ví dụ này, ta thay đổi kiểu trang (`paper`) và chọn `mode=handout` cùng với tùy chọn `nopagebreaks` – nhờ đó nhiều hơn hai trang `slide` sẽ được bố trí trên một trang giấy!

2.2 Thiết lập trình diễn

`\pdsetup` Dưới đây là các tùy chọn giúp bạn điều khiển tốt hơn nữa trình diễn của mình. Các tùy chọn này không phải chỉ ra khi gọi lệnh `\documentclass`, mà được chỉ ra trong tham số của lệnh `\pdsetup`. Có sự phân biệt này là vì lý do kỹ thuật.³ Lệnh `\pdsetup` chấp nhận chỉ một tham số, là chuỗi các tùy chọn cách nhau bằng dấu phẩy.

`lf` `lf`

Xác định nội dung của **footer** (chân trang) bên trái. Mặc định là rỗng.

`rf` `rf`

Như trên, nhưng cho bên phải.

`theslide` `theslide`

Tùy chọn này xác định cách đánh số các `slide`. Giá trị mặc định là

```
\arabic{slide}~/~\pageref*{lastslide}
```

mà với nó, các `slide` sẽ đánh số tương tự như 5/22 (slide 5, tổng số slide 22). Chú ý rằng, `\arabic{slide}` cho biết `slide` hiện tại, và `\pageref*{lastslide}` tổng số `slide`.⁴

`thenote` `thenote`

Tương tự như tùy chọn `theslide`, nhưng dành cho các ghi chú. Giá trị mặc định là

```
note~\arabic{note}~of~slide~\arabic{slide}
```

ở đây `\arabic{note}` là số chỉ ghi chú hiện tại. Kết quả sẽ tương tự như `note 2 of slide 7`.

Với người dùng Việt, có lẽ phải định nghĩa lại như sau: `ghi chú~\arabic{note}~/~\arabic{slide}`.

`trans` `trans`

Xác định hiệu ứng biến đổi mặc định cho trình diễn. Các hiệu ứng này chỉ thấy được trong kết quả PDF. Xem thêm ở Mục 8 ở trang 21. Các hiệu ứng được hỗ trợ bao gồm: `Split`,

³Nếu bạn quan tâm, thử tìm đọc mục về `xkvtxp` trong tài liệu [3] về gói `xkeyval`.

⁴Ta phải dùng `\pageref*`, phiên bản có sao của `\pageref`, được định nghĩa bởi `hyperref` nhờ đó ta thu được số trang chứ không phải là liên kết đến trang.

Blinds, Box, Wipe, Dissolve, Glitter và Replace. Nếu bạn dùng chương trình xem hiểu dạng PDF 1.5 trở lên, bạn có thêm các hiệu ứng Fly, Push, Cover, Uncover và Fade. Điều quan trọng cần phải nhớ là, hầu hết các trình xem PDF đều phân biệt hoa thường, nên bạn phải chỉ ra Box thay vì box,...

Hiệu ứng mặc định là Replace – theo đó, khi xem trình diễn, slide sau sẽ thay thế slide trước. Để ý rằng, vài trình xem PDF (ví dụ Acrobat Reader 5) chỉ tạo các hiệu ứng khi xem ở chế độ toàn màn hình. Nếu bạn dùng một hiệu ứng riêng không có trong danh sách trên (ví dụ hiệu ứng Wipe với hướng theo ý bạn), powerdot sẽ nhắc nhở về hiệu ứng đó (rằng có thể chúng không xem được). Dưới đây là ví dụ

```
trans=Wipe /Di 0
```

Trong Acrobat (Reader), hiệu ứng Wipe khai báo như trên sẽ có hướng từ trái sang phải, thay vì từ trên xuống dưới như mặc định. Để biết thêm chi tiết, xem tài liệu tham khảo về định dạng PDF.

counters

counters

Danh sách các counter (bộ đếm) cần được bảo vệ khi overlay. Tại sao? Các phương trình chẳng hạn, khi overlay (xem Mục 4 ở trang 11 sẽ được xử lý nhiều lần, do đó, số đếm của chúng không ngừng tăng lên nếu không được bảo vệ. Các bộ đếm được bảo vệ theo mặc định là equation, table, figure, footnote và mpfootnote. Nếu bạn muốn bảo vệ thêm chẳng hạn bộ đếm theorem và lemma, hãy dùng như ví dụ sau:

```
counters={theorem,lemma}
```

list

list

Tùy chọn này chấp nhận một tham số, là danh sách các tùy chọn dành riêng cho gói enumitem để điều khiển kết quả của danh sách tạo bởi các môi trường enumerate và itemize. Ví dụ

```
list={labelsep=1em,leftmargin=*,itemsep=0pt,topsep=5pt,parsep=0pt}
```

Hãy xem tài liệu hướng dẫn của gói enumitem để có thêm các tùy chọn.

enumerate

enumerate

Tương tự tùy chọn list, nhưng chỉ dành cho môi trường enumerate.

itemize

itemize

Tương tự tùy chọn list, nhưng chỉ dành cho môi trường itemize.

Dưới đây là ví dụ về cài đặt cho trình diễn dùng lệnh \pdfsetup.

```
\pdfsetup{
  lf=Trinh dien cua toi,
  rf=VietTUG,
  trans=Wipe,
  theslide=\arabic{slide}
}
```

Với thiết lập như trên, sẽ có chân trang bên trái và bên phải, hiệu ứng mặc định cho trình diễn là Wipe; ngoài ra, chỉ số slide sẽ không bao gồm tổng số slide như mặc định, mà chỉ gồm số chỉ slide hiện tại.

Một chú ý nhỏ về cách thể hiện ở chân trang (footer). Chỉ số slide (điều khiển bởi tùy chọn theslide) sẽ được thêm vào chân trang (footer). Hầu hết các kiểu trình diễn đều thêm vào bên phải. Nếu cả footer và chỉ số slide đều không rỗng, ~~~ sẽ được thêm vào để ngăn cách chúng. Khi thiết kế kiểu, ta có thể thay đổi cách xử lý này.

3 Tạo trình diễn

3.1 Trang tiêu đề

`\title` Trang tiêu đề của trình diễn được tạo bởi `\maketitle`, tương tự như cách quen thuộc trong `\author` L^AT_EX chuẩn. Ví dụ

`\and`

`\date`

`\maketitle`

```
\documentclass{powerdot}
\title{Title}
\author{You \and me}
\date{September 14, 2005}
\begin{document}
\maketitle
...
\end{document}
```

Các khai báo `author`, `title` và `date` cho biết lần lượt tác giả, tiêu đề và ngày tháng của trình diễn. Trang tiêu đề được trình bày theo thiết kế của kiểu trình diễn. Xem tài liệu [12] về chi tiết của các lệnh `\title` và `\author`.

3.2 Tạo slide

`slide` Phần trung tâm của mọi trình diễn là `slide`. Với `powerdot`, nội dung của mỗi `slide` được đặt trong môi trường `slide`.

```
\begin{slide}[\langle options \rangle]{\langle slide title \rangle}
\langle body \rangle
\end{slide}
```

Ở Mục 4 ở trang liền sau, bạn sẽ biết tinh chỉnh với tùy chọn `overlay`. Bây giờ, hãy xem một ví dụ đơn giản:

```
\begin{slide}{First slide}
Hello World.
\end{slide}
```

Môi trường `slide` có một tham số bắt buộc, là tựa của `slide`. Khi một `slide` được tạo ra, tựa của nó sẽ được liệt kê trong Bảng Mục lục và trong danh sách các `bookmark`. Bảng Mục lục liệt kê tựa của tất cả các `slide` và các mục của trình diễn, và nó sẽ xuất hiện ở mọi `slide`.

Ở Bảng Mục lục, có các liên kết (khi biên dịch qua dạng PDF) nhờ đó bạn có thể dễ dàng tìm kiếm các `slide` trong trình diễn của mình. Trong khi đó, danh sách `bookmark` cũng tương tự như Bảng Mục lục, nhưng nó không xuất hiện bất kỳ `slide` nào, mà ở một cửa sổ đặc biệt của trình xem PDF. Trong ví dụ trên, tựa `First slide` sẽ xuất hiện trong cả Bảng Mục lục và danh sách `bookmark`. Danh sách `bookmark` chỉ có khi biên dịch kết quả thành dạng PDF.

Phần tham số bổ sung (hay tùy chọn) (`\langle options \rangle`) của `slide` cho phép chỉ định tựa sẽ xuất hiện ở Bảng Mục lục và danh sách `bookmark`. Tùy chọn cũng giúp ta chỉ định hiệu ứng (`trans`) dùng riêng cho `slide` đó thôi.

`toc` `toc`

Chỉ định tựa sẽ xuất hiện trong Bảng Mục lục, thay vì tựa của `slide`. Nếu dùng `toc=`, sẽ không có phần tử tương ứng nào được tạo ra ở Bảng Mục lục.

`bm` `bm`

Tương tự như trên, nhưng cho danh sách `bookmark`.

`trans` `trans`

Cách dùng tùy chọn này hoàn toàn tương tự như mô tả ở Mục 2.2 ở trang 8; các thay đổi mà tùy chọn này tạo ra chỉ có tác dụng trong `slide` đang xét, không phải cho toàn bộ trình diễn hay các `slide` khác.

Các tùy chọn `toc` và `bm` đặc biệt hữu ích nếu tựa của `slide` quá dài hoặc khi nó chứa các lệnh `LaTeX` mà kết quả của lệnh đó không thể hiện tốt trong danh sách `bookmark`.⁵ Khi dùng tùy chọn `toc` và `bm`, bạn nhớ đặt các ký tự đặc biệt ‘,’ và ‘=’ bên trong cặp dấu ngoặc ‘{’ và ‘}’. Dưới đây là ví dụ:

```
\begin{slide}[toc=,bm={LaTeX, i*i=-1}]{\color{red}\LaTeX, $i^2=-1$}
  My slide contents.
\end{slide}
```

Ở ví dụ này, tựa của `slide` sẽ là `LaTeX, $i^2 = -1$` . Kết quả này không thể nào thể hiện đúng trong danh sách `bookmark`. Đôi khi, việc cố gắng sửa chữa tựa có thể mang lại kết quả, nhưng thường thì cố gắng không như ý. Chính vì thế, ta đã dùng tùy chọn `bm`, nhờ đó kết quả thu được ở danh sách `bookmark` sẽ là `LaTeX, i*i=-1`. Chú ý rằng, không có tựa nào được tạo ra ở Bảng Mục lục, do ta đã chỉ định `toc=`.

Cùng với môi trường `slide`, các kiểu trình diễn có cung cấp các môi trường riêng để tạo `slide`. Phổ biến nhất là môi trường `wideslide` để giải quyết vấn đề: đôi khi, nội dung của `slide` quá khổ, không thể bố trí trên một trang cùng với Bảng Mục lục; môi trường `wideslide` cho ta `slide` đặc biệt ở đó không xuất hiện Mục lục. Xem thêm Mục 7 ở trang 20 để biết thêm các môi trường cung cấp bởi các kiểu trình diễn.

4 Overlay

Với trình diễn, nhiều khi bạn không muốn mọi thông tin trên `slide` xuất hiện cùng một lúc, mà tuần tự từng ý từng ý một. Với `powerdot`, bạn có thể đạt kết quả này nhờ `overlay`. Mỗi `slide` hiển nhiên có thể gồm nhiều `overlay`; tại mỗi thời điểm, chỉ có một `overlay` được thể hiện.

4.1 Lệnh `\pause`

`\pause` Cách đơn giản nhất để tuần tự hóa thông tin trên `slide` là dùng lệnh `\pause`.

```
\pause[⟨number⟩]
```

Dưới đây là ví dụ đơn giản:

```
\begin{slide}{Simple overlay}
  power\pause dot
\end{slide}
```

Thông tin của `slide` được thể hiện và tiếp tục cho đến khi gặp lệnh `\pause` đầu tiên. Không có thêm kết quả nào xuất hiện cho đến khi bạn `click` chuột hoặc gõ phím bất kỳ. Sau đó, nội dung `slide` sẽ được thể hiện cho đến khi mọi thông tin đều xuất hiện hoặc cho tới khi gặp lệnh `\pause` kế tiếp. Trong ví dụ trên, `power` sẽ xuất hiện trong `overlay` đầu tiên, và `powerdot` sẽ xuất hiện ở `overlay` tiếp theo. Lệnh `\pause` thường được dùng cùng với các môi trường `itemize` hoặc `enumerate`. Ví dụ

```
\begin{slide}{Multiple pauses}
  power\pause dot \pause
  \begin{itemize}
    \item Let me pause\ldots \pause
    \item \ldots while I talk \pause and chew bubble gum. \pause
    \item Perhaps you'll be persuaded.
    \item Perhaps not.
  \end{itemize}
\end{slide}
```

⁵Các bookmark được tạo bằng cách dùng `\pdfstringdef` từ gói `hyperref`, có thể chấp nhận vài ký tự có dấu chẳng hạn `"i`.

Bởi vì `\pause` được dùng trước môi trường `itemize`, không có phần tử nào của danh sách xuất hiện cho tới `overlay` thứ ba. Sau đó, các phần tử sẽ lần lượt xuất hiện trong các `overlay` kế tiếp. Việc điều khiển danh sách sẽ được bàn kỹ hơn ở Mục kế tiếp.

Lệnh `\pause` có thể tham số bổ sung cho phép chỉ ra bao nhiêu `overlay` phải dừng. Ví dụ

```
\begin{slide}{Pause longer}
\begin{itemize}
\item A \pause
\item B \pause[2]
\item C
\end{itemize}
\end{slide}
```

Trong ví dụ trên, phần tử C của danh sách xuất hiện ở `overlay` thứ bốn. Tùy chọn có vẻ vô dụng này sẽ được đề cập lại ở Mục tiếp theo, ở đó ta sẽ gặp lại ví dụ tương tự ví dụ trên đây.

4.2 Tạo danh sách

Các môi trường tạo danh sách `itemize` và `enumerate` được xử lý đặc biệt trong `powerdot`. Chúng đều có tham số tùy chọn điều khiển bởi gói `enumitem` (xem [4]). Lớp `powerdot` cung cấp nhiều khóa cho tùy chọn này. Trong các ví dụ sau đây, các tính năng được minh họa với môi trường `itemize`, nhưng bạn cũng có thể áp dụng cách làm tương tự cho môi trường `enumerate`.

Dưới đây là ví dụ đơn giản nhất về cách dùng môi trường `itemize`:

```
\begin{slide}{Basic itemize}
\begin{itemize}
\item A \pause
\item B \pause
\item C
\end{itemize}
\end{slide}
```

Kết quả của ví dụ là danh sách đơn giản, mỗi phần tử của nó được thể hiện ở một `overlay`.

type Giả định rằng ta muốn mọi phần tử của danh sách sẽ được thể hiện, nhưng tại mỗi thời điểm chỉ một và một phần tử của danh sách xuất hiện ở `slide`. Điều này có thể đạt được bằng cách chỉ định kiểu với `type` trong phần tham số bổ sung của môi trường (giá trị mặc định là 0).

```
\begin{slide}{Type 1 itemize}
\begin{itemize}[type=1]
\item A \pause
\item B \pause
\item C
\end{itemize}
\end{slide}
```

Bây giờ, mọi phần tử đều được thể hiện với *màu nhạt* (màu được định nghĩa bởi kiểu trình diễn); khi `overlay` của một phần tử được thể hiện, phần tử đó sẽ được tô đậm để làm nổi bật nó so với các phần tử khác. Sự phân biệt của hai `type` có thể minh họa như sau (dấu * chỉ màu nhạt):

```
1: x----- x-----
2: xx----- *x-----
3: xxx----- **x-----
4: xxxx--- ***x---
5: xxxxx-- ****x--
6: xxxxxx- *****x-
```

Các danh sách có thể lồng nhau. Khi xảy ra trường hợp này, các danh sách thứ cấp sẽ thừa hưởng giá trị của `type` ở môi trường cấp cao hơn. Tất nhiên, với bất kỳ danh sách nào, ta cũng có thể đặt lại giá trị cho `type`. Dưới đây là ví dụ về hai danh sách lồng nhau:

```
\begin{slide}{Nested lists}
\begin{itemize}
\item A \pause
\begin{itemize}[type=1]
\item B \pause
\item C
\end{itemize}
\end{itemize}
```

```

\item B\pause
\end{itemize}
\item C
\end{itemize}
\end{slide}

```

Kết quả là phần tử A và B được thể hiện ở **overlay** đầu tiên, nhưng phần tử B được tô màu nhạt. Ở **overlay** thứ hai, phần tử B sẽ được tô màu đậm, và ở **overlay** thứ ba, C sẽ được tô đậm.

4.3 Lệnh \item

\item Lệnh **\item** để tạo phần tử cho danh sách có thể nhận tham số bổ sung (*optional*) cho phép tạo các **overlay** linh hoạt hơn lệnh **\pause**.

```
\item[⟨label⟩]<⟨overlays⟩>
```

Tham số bổ sung $\langle overlays \rangle$ này là danh sách các **overlay** mà phần tử sẽ xuất hiện (có nghĩa, một phần tử có thể xuất hiện ở một hay nhiều **overlay** được chỉ ra). Trong danh sách này, các **overlay** được cho bởi số, cách nhau bởi dấu phẩy. Tham số $\langle label \rangle$ được hiểu tương tự như trong

Cú pháp	Ý nghĩa
x	Chỉ ở overlay x
-x	Các overlay nhỏ hơn hoặc bằng x
x-	Các overlay lớn hơn hoặc bằng x
x-y	Các overlay từ x đến y, bao gồm cả x và y

Bảng 1: Quy ước cho **\item** và **\onslide**

L^AT_EX chuẩn, xem chi tiết ở chẳng hạn [12].

Dưới đây là ví dụ

```

\begin{slide}{Active itemize}
\begin{itemize}[type=1]
\item<1> A
\item<2> B
\item<3> C
\end{itemize}
\end{slide}

```

Với ví dụ này, phần tử A chỉ xuất hiện với màu tô đậm ở **overlay** 1, phần tử B chỉ xuất hiện + tô đậm ở **overlay** thứ hai,... Để ý rằng, do chỉ định **type=1**, nếu phần tử không phải là phần tử hiện tại, màu của nó được tô nhạt. Ta có thể minh họa như sau (dấu THƯỜNG chỉ màu nhạt):

```

1: --A----
2: ---aB---
3: ---abC--

```

Nếu chỉ định **type=0** và nếu mỗi phần tử của danh sách đều có thêm tham số bổ sung chỉ định **overlay**, thì một phần tử sẽ bị giấu hoàn toàn chứ không phải được tô màu nhạt. Ví dụ:

```

1: --A----
2: ---B---
3: ----C--

```

Ví dụ nhiều hơn về cú pháp của $\langle overlays \rangle$ được cho ở Mục tiếp theo.

4.4 Lệnh \onslide

\onslide Các **overlay** có thể thu được nhờ lệnh **\onslide**.

```
\onslide{<overlays>}{<text>}
```

Ở lệnh này, tham số *<overlays>* ở vị trí thứ nhất; tham số *<text>* ở vị trí thứ hai chỉ nội dung cần biểu diễn. Tham số *<overlays>* là danh sách các **overlay**, với cú pháp được nêu ở Bảng 1 ở trang trước. Xét ví dụ sau đây.

```
\begin{slide}{Simple onslide}
\onslide{1,2}{power}\onslide{2}{dot}
\end{slide}
```

Ở đây, chúng ta muốn **power** xuất hiện ở **overlay** thứ nhất và hai, và **dot** chỉ xuất hiện ở **overlay** thứ hai. Bạn có thể đoán được rằng, kết quả của ví dụ trên giống như khi dùng lệnh `\pause` ở ví dụ đầu tiên. Bạn có thể ngạc nhiên là, nếu vậy thì việc đưa ra lệnh `\oneslide` có vẻ rắc rối. Thực tế thì điều này có dụng ý, giúp cho bạn linh hoạt hơn trong quá trình tạo **overlay**.

`\onslide+` Ta hãy xét ví dụ trên, nhưng với một chút thay đổi như sau:

```
\begin{slide}{Simple onslide+}
\onslide+{1}{power}\onslide+{2}{dot}
\end{slide}
```

Bây giờ, cụm từ **dot** xuất hiện ở mọi **overlay**, nhưng nó có màu tô nhạt, và được tô màu bình thường chỉ khi nó xuất hiện ở **overlay** thứ hai. Điều này có nét tương tự như khi dùng `type=1` đối với danh sách (xem Mục 4.2 ở trang 12). Để ý rằng, ở đây ta nói đến các **overlay** tổng quát, không chỉ đơn giản là danh sách. Ta minh họa kết quả trên như sau đây, chữ HOA chỉ rằng chữ được tô màu bình thường, còn chữ THƯỜNG chỉ màu nhạt

```
1: POWERdot
2: powerDOT
```

Ta để ý rằng, mặc dù thực hiện 'giấu' (`\oneslide`) hoặc tô màu nhạt (`\onslide+`) cho nội dung *<text>*, nó vẫn giữ một khoảng trắng cho phần nội dung đó – hệt như khi nội dung đó xuất hiện. Lệnh `\onslide*` trong được mô tả sau đây sẽ không làm như vậy.

`\onslide*` Thay vì 'giấu' (hoặc tô màu nhạt) cho phần nội dung *<text>*, nếu nội dung *<text>* không được thiết lập để hiển thị ở *<overlay>*, lệnh này bỏ qua hoàn toàn nội dung đó. Để rõ hơn, ta hãy xét ví dụ sau đây:

```
\begin{slide}{Simple onslide*}
\onslide{1}{power}\onslide{2}{dot}\\
\onslide+{1}{power}\onslide+{2}{dot}\\
\onslide*{1}{power}\onslide*{2}{dot}
\end{slide}
```

Kết quả của hai dòng đầu tiên chúng ta đã quen thuộc. Hãy xét dòng thứ ba. Chữ **power** xuất hiện ở **overlay** thứ nhất, còn **dot** ở **overlay** thứ hai. Tuy nhiên, không có khoảng trắng nào được dành cho **power** ở **overlay** thứ hai. Do đó, **dot** sẽ xuất hiện ở vị trí mà **power** xuất hiện. Kết quả có thể minh họa như sau (chữ HOA: tô màu thường, chữ THƯỜNG: tô màu nhạt; mỗi chữ x tượng trưng cho một vị trí bỏ trống).

```
1: POWERxxx 2: xxxxxDOT
1: POWERdot 2: powerDOT
1: POWERxxx 2: DOTxxxxx
```

Ta kết thúc mục này với ví dụ phức tạp. Nhớ rằng những lệnh `\item` và `\onslide` nhận tham số *<overlays>* là danh sách các **overlay** theo cú pháp ở Bản 1 ở trang trước.

```
\begin{slide}{Lists}
\onslide{10}{chi o overlay 10}\par
\onslide{-5}{cac onverlay <= 5}\par
\onslide{5-}{cac overlay >= 5}\par
\onslide{2-5}{cac overlay 2,3,4,5}\par
\onslide{-3,5-7,9-}{overlay >= 1, tru 4 va 8}
\end{slide}
```

4.5 Overlay tương đối

Thật phiền phải nhớ thứ tự các **overlay** để biểu diễn đúng nội dung nào trước, nội dung nội sau. Trong thực tế, ta chỉ cần nhớ, chẳng hạn chữ **power** sẽ xuất hiện trước chữ **dot** là đủ. Điều này có thể thực hiện nhờ **overlay** tương đối và môi trường tạo danh sách. Hãy xem ví dụ đơn giản sau đây

```
\begin{slide}{Relative overlays}
\begin{itemize}
\item A \pause
\item B \onslide{+1}{(visible 1 overlay after B)}\pause
\item C \onslide{+2-}{(appears 2 overlays after C, visible until the end)}
\pause
\item D \onslide{+1-6}{(appears 1 overlay after D, visible until overlay 6)}
\pause
\item E \pause
\item F \pause
\item G \onslide{+1-+3}{(appears 1 overlay after G for 3 overlays)}\pause
\item H \pause
\item I \pause
\item J \pause
\item K
\end{itemize}
\end{slide}
```

Ở đây, ta vẫn dùng `\oneslide`, nhưng với cú pháp mới của các **overlay**. Đó là ta có thể sử dụng dấu ‘+’ trong danh sách. Ở ví dụ đơn giản nhất, `\onslide{+1}` sẽ thể hiện nội dung ở một **overlay** kế tiếp. Ta vẫn có thể dùng cú pháp đã nêu ở Bảng 1 ở trang 13. Ví dụ, `\onslide{+1-6}{power}` sẽ thể hiện chữ **power** ở các **overlay** kế tiếp cho đến **overlay** mang số 6. Ở ví dụ cuối cùng, `\onslide{+1-+3}{power}`, chữ **power** sẽ xuất hiện ở **overlay** kế tiếp và sẽ tiếp tục xuất hiện thêm ở 3 **overlay** nữa. Ta minh họa kết quả trên như sau đây (dấu * đại diện cho tham số của các lệnh `\onslide` ở trên:

```
1: A
2: A_B
3: A_B*_C
4: A_B__C__D
5: A_B__C*_D*_E
6: A_B__C*_D*_E_F
7: A_B__C*_D__E_F_G_
8: A_B__C*_D__E_F_G*_H
9: A_B__C*_D__E_F_G*_H_I
10: A_B__C*_D__E_F_G*_H_I_J
11: A_B__C*_D__E_F_G__H_I_J_K
```

5 Cấu trúc trình diễn

5.1 Tạo mục

`\section` Lệnh `\section` cho phép tạo một mục mới trong trình diễn, tương tự như cách dùng lệnh `\section` của lớp `article`.

```
\section[<options>]{<section title>}
```

Lệnh này sẽ tạo một **slide** chỉ với nội dung là (tựa đề) *<section title>* (hãy xem thêm về tùy chọn **slide** ở bên dưới). Tựa đề *<section title>* cũng sẽ xuất hiện ở Bảng Mục lục và danh sách **bookmark**. Có vài tùy chọn cho lệnh này như sau:

`tocsection` Tùy chọn điều khiển việc tạo phần tử tương ứng cho **section** ở Bảng Mục lục. Giá trị mặc định **true**.

`tocsection=true`

Tạo mục tương ứng với `section` hiện tại trong Bảng Mục lục.

`tocsection=false`

Chỉ tạo `slide` cho mục nhưng không tạo phần tử tương ứng trong trang Mục lục.

`tocsection=hidden`

Tạo mục tương ứng trong Bảng Mục lục, nhưng mục này chỉ nhìn thấy khi bạn đang xem một trong các `slide` của Mục đang xét. Điều này có ích, chẳng hạn khi bạn có thêm thời gian để trình bày mục này (một cách dự trữ!).

slide Tùy chọn này cho phép hay không lệnh `\section` tạo riêng cho nó một `slide`. Mặc định là `true`.

`slide=true`

Một `slide` được tạo ra với nội dung là tựa của mục.

`slide=false`

Không `slide` được tạo ra. Nếu đồng thời `tocsection` nhận giá trị `false`, thì lệnh `\section` không làm gì cả. Nếu `tocsection` nhận giá trị `true` hoặc `hidden`, thì chọn mục tương ứng trong Bảng Mục lục, ta sẽ nhảy đến `slide` đầu tiên của mục (bởi không có `slide` riêng cho Mục).

template Tùy chọn này cho phép Mục đang xét chọn một mẫu khác. Theo mặc định, môi trường `slide` được dùng để tạo ra `slide` cho mục, nhưng nếu bạn muốn mẫu khác được dùng cho mục đích này, ví dụ `wideslide`, thì bạn có thể dùng tùy chọn này để chỉ định mẫu đó. Xem thêm ở Mục 7 ở trang 20 để biết thêm chi tiết về các mẫu của các kiểu khác nhau.

Cuối cùng, mọi tùy chọn của `slide` đều có thể dùng cho `\section`, ví dụ `toc=`, `bm=`,... Xem thêm ở Mục 3 ở trang 10.

5.2 Tạo slide Mục lục

\tableofcontents Lệnh này tạo `slide` Mục lục giúp bạn có cái nhìn tổng quan về trình diễn hoặc một phần (mục) của trình diễn. Lệnh này chỉ được dùng bên trong môi trường tạo `slide`, và bạn có thể gọi nó bao nhiêu lần tùy thích.

Nội dung được nói đến trong mục này là nội dung của `slide` Mục lục. Để ý rằng, `slide` Mục lục khác với Bảng Mục lục. Một trình diễn chỉ có một Bảng Mục lục nhưng có thể có nhiều `slide` mục lục.

`\tableofcontents[options]`

Có vài tùy chọn cho lệnh như sau:

type Xác định giấu hoặc tô nhạt một vài phần (phụ thuộc vào giá trị của `content` dưới đây). Giá trị mặc định là 0. So sánh với tùy chọn `type` của môi trường tạo danh sách ở Mục 4.2 ở trang 12.

`type=0`

Nếu nội dung không đúng kiểu như chỉ ra ở tùy chọn `content`, nó sẽ được giấu đi.

`type=1`

Như trên, nhưng thay vì giấu đi, nội dung sẽ được hiện với màu tô nhạt.

content Tùy chọn này cho phép xác định những phần tử nào sẽ được thể hiện ở `slide` mục lục. Giá trị mặc định là `all`. Mô tả dưới đây giả định rằng `type=0` được chọn. Bạn có thể dễ dàng suy ra kết quả khi `type=1` từ mô tả này.

`content=all`

Cho ra **slide** đầy đủ, gồm mọi mục và **slide** trong trình diễn của bạn, trừ các mục ẩn (xem mô tả trong Mục 5.1 ở trang 15).

`content=sections`

Chỉ liệt kê các Mục của trình diễn.

`content=currentsection`

Chỉ liệt kê các **slide** của mục hiện tại.

`content=future`

Liệt kê mọi mục và **slide** bắt đầu từ **slide** hiện tại.

`content=futuresections`

Liệt kê mọi mục bắt đầu từ mục hiện tại.

Dưới đây là ví dụ nhỏ. Trình diễn ở ví dụ này gồm các mục, đầu mỗi mục là **slide** mục lục có nhiệm vụ tóm tắt (liệt kê) các phần của Mục đó.

```
\begin{slide}[toc,bm=]{Overview}
  \tableofcontents[content=sections]
\end{slide}
\section{First section}
\begin{slide}[toc,bm=]{Overview of the first section}
  \tableofcontents[content=currentsection,type=1]
\end{slide}
\begin{slide}{Some slide}
\end{slide}
\section{Second section}
...
```

6 Linh tinh

6.1 Lệnh `\twocolumn` chia cột

`\twocolumn` Lệnh `\twocolumn` cho phép bố trí nội dung ở hai cột của trang.

`\twocolumn{<options>}{<left>}{<right>}`

Lệnh này sẽ bố trí `<left>` và `<right>` vào hai cột bên trái và bên phải của **slide**. Kích thước của các cột được cho ở tùy chọn `<options>`.

`lineheight` `lineheight`

Nếu `lineheight` được chỉ ra, một dòng kẻ (tạo ra nhờ lệnh `\psline`) với chiều cao chỉ định được chèn để phân cách hai cột. Ví dụ `lineheight=6cm`.

`lineprop` `lineprop`

Các khai báo `pstricks` để chỉ thuộc tính của dòng kẻ. Ví dụ

`lineprop={linestyle=dotted,linewidth=3pt}`

`lfrheight` `lfrheight`

Tạo ra một khung (**frame**) với chiều cao chỉ định xung quanh cột bên trái.

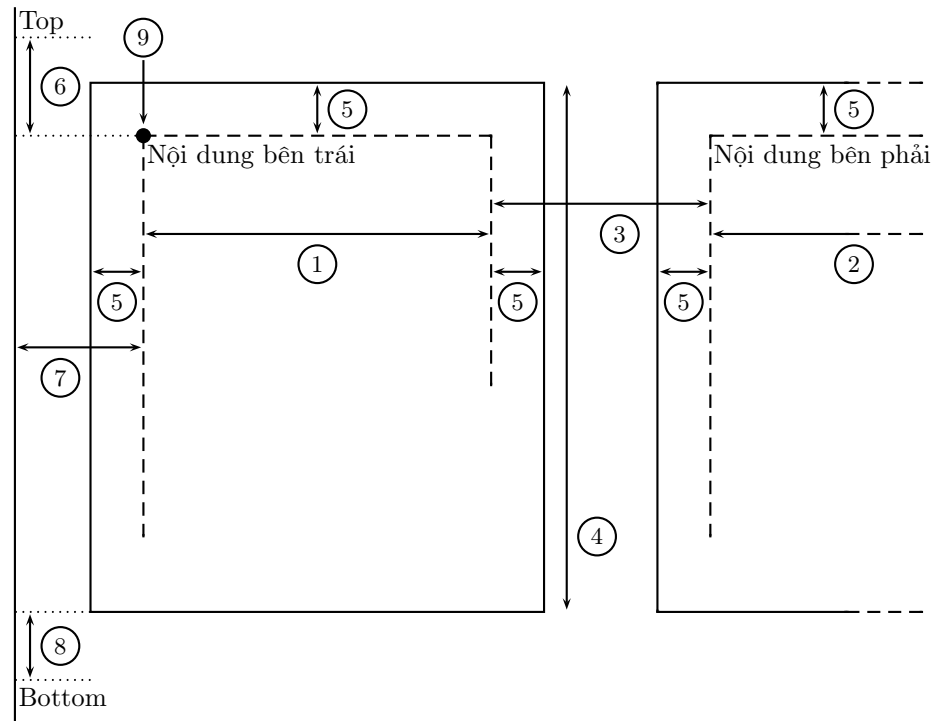
<code>lfrprop</code>	<code>lfrprop</code>	Như <code>lineprop</code> , nhưng cho đường kẻ của khung bên trái.
<code>rfrheight</code>	<code>rfrheight</code>	Tạo khung bên phải với chiều cao chỉ định.
<code>rfrprop</code>	<code>rfrprop</code>	Như <code>lineprop</code> nhưng dành cho đường kẻ của khung phải.
<code>frsep</code>	<code>frsep</code>	Khoảng cách giữa nội dung và khung (<code>frame</code>). Mặc định 1.5mm.
<code>colsep</code>	<code>colsep</code>	Khoảng cách giữa hai cột. Mặc định $0.06\backslash linewidth$.
<code>lcolwidth</code>	<code>lcolwidth</code>	Chiều rộng của khung bên trái. Mặc định $0.47\backslash linewidth$.
<code>rcolwidth</code>	<code>rcolwidth</code>	Chiều rộng của khung bên phải. Mặc định $0.47\backslash linewidth$.
<code>topsep</code>	<code>topsep</code>	Khoảng trắng thêm vào bên trên cột và bên trên các dòng nội dung (cộng thêm vào giá trị đã có của <code>\baselineskip</code>). Mặc định 0cm.
<code>bottomsep</code>	<code>bottomsep</code>	Nhưng trên, nhưng cho bên dưới cột và dòng. Mặc định 0cm.
<code>indent</code>	<code>indent</code>	Khoảng cách thụt đầu dòng so với cột bên trái. ⁶ Mặc định 0cm.

Mô tả hình học của các tùy chọn trên được cho ở Hình 1 ở trang sau. Điều chú ý quan trọng là, lệnh `\twocolumn` dùng vị trí hiện tại của con trỏ như là điểm tham khảo đến dòng đầu tiên của nội dung ở cột bên trái (xem ở Bảng 1 ở trang liền sau). Điều này có nghĩa rằng, khung được tạo ra sẽ ảnh hưởng đến dòng trước đó. Hãy dùng chẳng hạn `topsep=0.3cm` để thêm khoảng trắng thích hợp vào giữa hai dòng này. Giá trị mặc định của `topsep` là 0cm được lấy với giả định rằng không có nội dung nào trước khi bắt đầu chế độ hai cột. Trong trường hợp này, dòng đầu tiên của cột bên trái có vị trí như các dòng (đầu tiên) tạo ở các `slide` khác. Việc cài đặt `topsep=0cm` bảo đảm điều này. Tuy nhiên, việc phối hợp hai giá trị `topsep` và `indent` cho phép bạn định vị dòng đầu tiên của cột bên trái theo ý bạn.

Lệnh `\twocolumn` tính toán chiều cao của cột dựa trên nội dung của nó, nhờ đó mới có thể bố trí đúng nội dung vào các cột. Việc tính toán này được tinh chỉnh nhờ lấy giá trị lớn nhất trong hai giá trị `lfrheight` và `rfrheight`.

```
\begin{slide}{Two columns}
Here are two columns.
\twocolumn{lfrprop={linestyle=dotted,linewidth=3pt},
lfrheight=4cm,rfrheight=5cm,lineheight=3cm,topsep=0.3cm}
}{left}{right}
That were two columns.
\end{slide}
```

⁶Horizontal indent left to the left column



Ý nghĩa của nhãn

1	lcolwidth	5	frsep
2	rcolwidth	6	topsep
3	colsep	7	indent
4	lfrheight, rfrheight, lineheight	8	bottomsep
		9	Điểm tham khảo

Hình 1: Các kích thước ở chế độ hai cột.

6.2 Ghi chú

note Môi trường **note** dùng để có các ghi chú (cá nhân) đi cùng với trình diễn. Có thể điều làm xuất hiện các ghi chú này nhờ tùy chọn **display** đã nói ở Mục 2.1. Dưới đây là ví dụ

```
\begin{slide}{Chewing gum}
...
\end{slide}
\begin{note}{Reminder for chewing gum}
Don't forget to mention that chewing gum is sticky.
\end{note}
```

6.3 Slide trắng

emptyslide Môi trường **emptyslide** tạo ra một **slide** trắng (không có nội dung gì). Nhờ đó, bạn có thể nạp vào hình vẽ chẳng hạn. Nên nhớ rằng, nếu không dùng **slide** trắng, bạn phải chỉ ra tựa đề của **slide** và điều này sẽ ảnh hưởng đến Bảng Mục lục cũng như danh sách **bookmakr**. Ví dụ

```
\begin{emptyslide}{}
\centering
\vspace{\stretch{1}}
\includegraphics[height=0.8\slideheight]{me_chewing_gum.eps}
\vspace{\stretch{1}}
\end{emptyslide}
```

Lệnh `\includegraphics` được lấy từ gói `graphicx` [5]. Lệnh `\stretch` dùng để canh hình vẽ theo chiều đứng. Cả hai lệnh này được mô tả trong [12]. Chú ý rằng, các biến độ dài `\slideheight` và `slidewidth` được dùng để bố trí hình vẽ vừa vện lên `slide`.

6.4 Slide tài liệu tham khảo

`thebibliography` powerdot định nghĩa lại môi trường `thebibliography` của lớp chuẩn `article` để tạo trang tài liệu tham khảo. Sự khác biệt là môi trường mới không tạo ra tựa đề và không tạo các dòng chữ ở đầu trang (`header`); các tính chất khác đều được bảo toàn. Bạn có thể dùng một trong hai cách sau đây (tùy thuộc bạn có dùng `BiBTeX` hay không):

```
\begin{slide}{Slide}
  \cite{someone}
\end{slide}
\begin{slide}{References}
  \begin{thebibliography}{1}
    \bibitem{someone} Article of someone.
  \end{thebibliography}
\end{slide}
```

```
\begin{slide}{Slide}
  \cite{someone}
\end{slide}
\begin{slide}{References}
  \bibliographystyle{plain}
  \bibliography{YourBib}
\end{slide}
```

Trong trường hợp bạn có một danh sách rất dài các tài liệu tham khảo, bạn có thể muốn chia danh sách đó thành nhiều `slide`, giống như cách làm của gói `natbib` và `bibentry` [7]. Việc dùng cả hai gói đó sẽ làm bạn vừa ý:

```
\begin{slide}{References (1)}
  \bibliographystyle{plain}
  \nobibliography{YourBib}
  \bibentry{someone1}
  \bibentry{someone2}
\end{slide}
\begin{slide}{References (2)}
  \bibentry{someone3}
\end{slide}
```

Bạn nên tìm kiếm tài liệu về việc trích dẫn và các tạo danh sách tài liệu tham khảo trong [12].

7 Các kiểu trình diễn

Lớp `powerdot` được phân phối cùng với một số kiểu (`style`, xì-tin :) được mô tả dưới đây. Bạn nên thử qua chúng để chọn lấy kiểu ưa thích. Việc lựa chọn kiểu đã nói đến trong Mục 2.1 ở trang 5.

default

Kiểu mặc định, chỉ với màu xanh sáng và trắng. Một bông hoa ở góc trên bên trái của mỗi `slide`. Kiểu này cung cấp môi trường `wideslide` và hướng `portrait`. Mỗi `slide` có Bảng Mục lục ở bên trái (chế độ `landscape`) hoặc bên dưới (chế độ `portrait`). Kiểu này đòi hỏi gói `pifont`.

simple

Kiểu đơn giản chỉ với hai màu đen, trắng. Hỗ trợ môi trường `wideslide` và hướng `portrait`. Bảng Mục lục được bố trí bên trái (`landscape`) hoặc bên dưới (`portrait`) của `slide`. Kiểu này cần gói `amssymb` và gói `pifont`.

tycja

Kiểu với màu vàng và xanh thẫm. Hỗ trợ môi trường `widestyle` và hướng `portrait`. Bảng Mục lục bố trí bên phải (`landscape`) hoặc bên dưới (`portrait`). Đòi hỏi gói `pst-grad` [15, 16] và gói `pifont`.

ikedat

Kiểu với màu đỏ, xanh (màu khung, bóng) và màu sáng (cho nội dung). Kiểu này thiết kế các đường khung, viền khá đẹp. Hỗ trợ môi trường `wideslide` và hướng `portrait`. Bảng mục lục bố trí bên trái (`landscape`) hoặc bên dưới (`portrait`). Đòi hỏi gói `calc` và gói `pifont`.

fyma

Nguyên gốc của kiểu này là của Laurent Jacques dành cho `prosper`. Dựa trên bản gốc đó, L. Jacques tạo ra bản cho `HA-prosper` với vài tính năng nổi trội. Được phép của L. Jacques, kiểu này được Shun'ichi J. Amano chuyển thành phiên bản cho `powerdot`. Kiểu được thiết kế gọn đẹp, với màu xanh sáng và màu nền `gradient`. Kiểu hỗ trợ môi trường `wideslide` và hướng `portrait`; ngoài ra, còn cung cấp các mẫu đặc biệt để tạo Mục trong trình diễn. Bảng Mục lục được bố trí bên trái (`landscape`) hoặc bên dưới (`portrait`). Đòi hỏi các gói `pst-grad` [15, 16].

ciment

Bản nguyên thủy của kiểu này là của Matheiu Goutelle dành cho `prosper` và `HA-prosper`. Được phép của tác giả, bản đó được thay đổi để dành cho `powerdot`. Với kiểu này, màu nền `slide` được trang điểm bằng nhiều đường kẻ ngang màu xám. Bảng mục lục được bố trí bên trái (`landscape`) hoặc bên phải (`portrait`). Đòi hỏi gói `pifont`.

8 Biên dịch. Cài đặt. Xem

8.1 Gói phụ thuộc

Ở Bảng 2 là danh sách các gói đòi hỏi bởi `powerdot` (gói phụ thuộc của các gói trong bảng này không được liệt kê). Chú ý rằng mỗi kiểu có những đòi hỏi riêng (xem Mục 7 ở trang trước). Trong bảng dưới đây, các gói ‘bắt buộc’ cho biết bạn không thể dùng phiên bản cũ hơn của gói đó; trong khi các gói ‘đã kiểm tra’ có nghĩa `powerdot` đã được kiểm tra sự tương thích với phiên bản đã chỉ ra, nhưng có thể vẫn gặp lỗi với các phiên bản cũ hơn hoặc mới hơn của gói đó. Để biết phiên bản của các gói được dùng, bạn hãy thêm lệnh `\listfiles` vào đầu tài liệu, biên dịch bằng `LATEX` rồi mở tập tin `.log` để xem kết quả. Để có bản cập nhật của các gói trên, bạn có thể tải chúng về từ CTAN [6] hoặc VietTUG [1].

Gói/tập tin	Phiên bản	Ngày	Mức độ
xkeyval [3]	2.5c	2005/07/10	bắt buộc
pstricks.sty [15, 16]	0.2l	2004/05/12	bắt buộc
xcolor [9]	1.11	2004/05/09	bắt buộc
enumitem [4]	1.1	2005/05/12	bắt buộc
article class	1.4f	2004/02/16	đã kiểm tra
geometry [14]	3.2	2002/07/08	đã kiểm tra
hyperref [13]	6.74m	2003/11/30	đã kiểm tra
graphicx [5]	1.0f	1999/02/16	đã kiểm tra
float [11]	1.3d	2001/11/08	đã kiểm tra

Bảng 2: Gói phụ thuộc

8.2 Cài đặt

Bạn tải về các gói sau đây:

`ftp://tug.ctan.org/pub/tex-archive/macros/latex/contrib/powerdot.zip`

`ftp://tug.ctan.org/pub/tex-archive/macros/latex/contrib/enumitem.zip`

`ftp://tug.ctan.org/pub/tex-archive/macros/latex/contrib/xcolor.zip`

`ftp://tug.ctan.org/pub/tex-archive/macros/latex/contrib/xkeyval.zip`

`ftp://ftp.jaist.ac.jp/pub/TeX/tex-archive/graphics/pstricks/latex/pstricks.sty`

Sau khi xả nén các tập tin `.zip` (vào cùng thư mục nào đó), bạn có được cấu trúc thư mục như sau.

```
enumitem
'-- enumitem.sty
powerdot
|-- doc
|-- run
|   |-- powerdot-*.sty
|   '-- powerdot.cls
'-- source
    '-- powerdot.dtx
xcolor
|-- xcolor.ins
xkeyval
|-- doc
|-- run
|   |-- *.tex
|   |-- *.sty
```

Bạn phải làm công việc tương đối phức tạp sau đây: biên dịch tập tin `xcolor.ins`. Nếu chương trình soạn thảo của bạn không hỗ trợ việc biên dịch tập tin này, bạn thử chạy từ dòng lệnh (tại thư mục `xcolor`)

```
latex xcolor.ins
```

Việc biên dịch này đảm bảo rằng trong thư mục `xcolor` ở trên sẽ xuất hiện thêm các tập tin `.def`, `.sty` và `.pro`.

Bây giờ hãy tìm đến thư mục `LocalTeXMF` trên hệ thống của bạn. Đó thường là `c:\localtexmf` đối với MikTeX hoặc là `~/texmf` đối với TeX. Hãy tạo các thư mục sau bên dưới thư mục `LocalTeXMF`

```
tex/latex/xcolor
tex/latex/enumitem
tex/latex/xkeyval
tex/latex/powerdot
tex/latex/pstricks
dvips/xcolor
```

Hãy chép các tập tin vào thư mục tương ứng

```
dvips/xcolor      <= xcolor/*.pro
tex/latex/xcolor  <= xcolor/*.def xcolor/*.sty
tex/latex/enumitem <= enumitem/enumitem.sty
tex/latex/xkeyval <= xkeyval/run/*
tex/latex/powerdot <= powerdot/run/*
tex/latex/pstricks <= pstricks.sty
```

Cuối cùng, hãy cập nhật hệ thống TeX của bạn. Với TeX, bạn thi hành lệnh sau

```
$ texhash ~/texmf
```

còn với MikTeX, bạn chạy chương trình `mo.exe` (từ Menu `RUN` chẳng hạn) rồi chọn **‘Refresh now’** để cập nhật hệ thống TeX.

8.3 Biên dịch và xem kết quả

Để có được trình diễn thực sự, đầu tiên, bạn biên dịch tài liệu nguồn bằng L^AT_EX. Kết quả DVI cho bởi bước này có thể xem bằng trình YAP của MikTeX.⁷ Thật không may, các chương trình `xdvi` và `kdvi` trên hệ thống `*nix` không hỗ trợ tập tin DVI có những mã lệnh `PostScript`, do đó, bạn không thể thấy kết quả như ý với các trình xem này. Nếu trình xem DVI hỗ trợ `PostScript`, hãy chắc rằng các cài đặt của trình xem đó khớp với các thuộc tính tương ứng của trình diễn. Khi

⁷Trừ khi bạn sử dụng gói `pstricks-add` làm ảnh hưởng đến cách tính tọa độ của YAP.

bạn tạo ra trình diễn **screen**, bạn nên cài đặt để trình xem hiển thị khổ xem **letter**. Tốt hơn nữa, nếu trình xem cho phép, bạn hãy chỉnh cỡ trang xem là 8.25 inch và 11 inch.

Chú ý rằng, các kết quả, hiệu ứng tạo với mã lệnh **PostScript** hoặc **PDF** không thể thấy được với trình xem **DVI**. Ví dụ, việc che giấu các phần tử của danh sách, việc tạm dừng **slide** bằng lệnh **\pause**, ...

Nếu bạn muốn có kết quả **postscript** của tài liệu, sử dụng chương trình **dvips** để chuyển từ tập tin **DVI** thu được sang sang **PS**. *Nhớ đừng chỉ ra cỡ trang ghi gọi lệnh này*, bởi **powerdot** đã ghi sẵn các mã lệnh **PostScript** để định cỡ trang.

Cuối cùng, từ kết quả **PS**, bạn có thể có được kết quả **PDF** bằng cách sử dụng chương trình chuyển đổi **ps2pdf**. Để ý rằng, chương trình **ps2pdf** cần sử dụng các công cụ của **PostScript**. Trên hệ thống Slackware Linux chẳng hạn, sau khi cài te \TeX , bạn có thể cài thêm các gói **espgs** và **gnu-gs-fonts** để đảm bảo chương trình **ps2pdf** làm việc tốt. Với hệ thống \TeX trên Windows, bạn phải cài thêm chương trình **ghostscript** chẳng hạn.

Từ kết quả **PS**, bạn cũng có thể dùng chương trình **psnup** để đặt nhiều **slide** lên cùng một trang.

Trong quá trình chuyển đổi **DVI** sang sang **PS** rồi **PDF**, nếu bạn gặp trục trặc với cỡ trang hoặc hướng trang, thử dùng tùy chọn **nopsheader** như đã nói ở Mục 2.1 ở trang 5; khi đó, bạn nhớ chỉ ra thiết lập của mình ở dòng lệnh khi gọi các chương trình **dvips** hoặc **ps2pdf**.

Tóm lại, nếu bạn có tài liệu **foo.tex**, thì quá trình biên dịch để tạo trình diễn gồm các bước sau (dấu **\$** chỉ dấu nhắc ở dòng lệnh)

```
$ latex foo
$ latex foo
$ dvips -o foo.ps foo.dvi
$ ps2pdf foo.ps
```

(phải biên dịch một, hai hoặc nhiều lần hơn bằng **latex** để các tham khảo chéo được chính xác.)

9 Tạo kiểu mới

Mục này bàn về việc thiết kế kiểu mới cho **powerdot**.

Mục này sẽ được dịch trong bản tiếp theo của tài liệu ;)

10 Hỏi-Đáp

10.1 Câu hỏi thường gặp

Dưới đây là các câu hỏi thường gặp. Hãy đọc kỹ, vì trục trặc của bạn có thể tìm thấy lời giải ở đây.

Q1 Lớp **powerdot** có cung cấp ví dụ? Tìm chúng ở đâu?

A1 Trong thư mục tài liệu **doc** của bộ cài đặt \LaTeX , bạn có thể tìm thấy các ví dụ cho **powerdot**. Nếu không thấy ở đó, bạn có thể tải chúng về từ **CTAN:/macros/latex/contrib/powerdot** [6].

Q2 Sao tôi gặp lỗi chỉ với ví dụ đơn giản nhất?

A2 Bạn đã đọc Mục 8.1 ở trang 21 chưa?

Q3 Tôi gặp lỗi **typo** trong tài liệu nguồn, do đó gặp lỗi khi biên dịch tài liệu. Sau đó, tôi đã sửa lỗi **typo** đó và biên dịch, nhưng lỗi đã gặp vẫn xuất hiện.

A3 Hãy xóa các tập tin **.bm** và **.toc** và thử lại xem.

Q4 Các môi trường **figure** và **table** sinh ra lỗi: **Note in outer par mode**.

A4 Bạn đã yêu cầu \LaTeX dùng các môi trường **float**, tương tự như ví dụ sau

```
\begin{figure}[htb]
```

Nguyên nhân sinh ra lỗi là \LaTeX không có chỗ để bố trí `float`. Vì thế, hãy thử bỏ đi tham số của môi trường `figure` hoặc `table`, trong ví dụ trên là `[htb]`.

Q5 Tôi có thể đóng góp cho dự án `powerdot`?

A5 Tất nhiên. Nếu bạn gặp (lỗi) `BUGS`⁸ hoặc lỗi `typo`, vui lòng gửi thư đến `mailinglist` (xem Mục 10.2). Nếu bạn thiết kế kiểu trình diễn riêng và muốn chúng có trong bản phân phối của `powerdot`, vui lòng gửi email riêng cho tác giả. Chú ý rằng, tác phẩm của bạn sẽ được phân phối theo `license` và `copyright` của `powerdot`; tên của bạn sẽ được nêu trong tài liệu của `powerdot`.

Nếu bạn có các câu hỏi khác, không nằm trong danh sách trên, hãy xem Mục kế tiếp để biết cách tìm câu trả lời.

10.2 Mailinglist

`powerdot` có `mailinglist` riêng ở

<http://www.freelists.org/list/powerdot>

Với vấn đề bạn gặp phải, thử tìm kiếm ở phần lưu trữ (`archive`) của `mailinglist`. Nếu không tìm ra, bạn có thể đăng ký làm thành viên của `mailinglist` bằng cách vào phần `subscribe` và nhấn nút `go`. Hãy kiểm tra hộp thư của bạn vào theo các hướng dẫn trong đó. Sau khi là thành viên của `mailinglist`, bạn có thể gửi các câu hỏi mới, bằng cách gửi email tới địa chỉ

`powerdot@freelists.org`

Khi gửi câu hỏi đến `mailinglist`, hãy nhớ kỹ những điều quan trọng sau

1. Trả lời cho bạn là những người tình nguyện.
2. Luôn cung cấp ví dụ đơn giản nhất có thể để minh họa cho trục trặc của bạn.
3. Không gửi kèm các tập tin kích thước lớn đến `mailinglist`.

Hy vọng rằng `mailinglist` sẽ giúp ích cho bạn.

11 Mã nguồn `powerdot`

Mã nguồn của lớp `powerdot` là tập tin `powerdot.dtx`. Biên dịch mã nguồn này bạn thu được tài liệu đầy đủ nhất về `powerdot`, kể cả mã \LaTeX của lớp với các chú thích kỹ thuật.

Bạn có thể biên dịch mã nguồn này theo các bước sau:

```
latex powerdot.dtx
latex powerdot.dtx
bibtex powerdot
makeindex -s gglo.ist -o powerdot.gls powerdot.glo
makeindex -s gind.ist -o powerdot.ind powerdot.idx
latex powerdot.dtx
latex powerdot.dtx
```

⁸Hãy chắc rằng đó là lỗi do `powerdot` sinh ra chứ không phải bởi gói khác.

Tài liệu

- [1] ViệtTUG. Cộng đồng người Việt sử dụng \TeX . <http://www.vietttug.org/>.
- [2] Hendri Adriaens. Gói HA-prosper. CTAN:/macros/latex/contrib/HA-prosper.
- [3] Hendri Adriaens. Gói xkeyval. CTAN:/macros/latex/contrib/xkeyval.
- [4] Javier Bezos. Gói enumitem. CTAN:/macros/latex/contrib/enumitem.
- [5] David Carlisle. Họ gói graphics. CTAN:/macros/latex/required/graphics.
- [6] CTAN crew. The Comprehensive TeX Archive Network. <http://www.ctan.org>.
- [7] Patrick W. Daly. Gói natbib. CTAN:/macros/latex/contrib/natbib.
- [8] Frédéric Goualard và Peter Møller Neergaard. Lớp prosper. CTAN:/macros/latex/contrib/prosper.
- [9] Uwe Kern. Gói xcolor. CTAN:/macros/latex/contrib/xcolor.
- [10] James Kilfiger and Wolfgang May. Họ gói extsizes. CTAN:/macros/latex/contrib/extsizes.
- [11] Anselm Lingnau. Gói float. CTAN:/macros/latex/contrib/float.
- [12] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The L^AT_EX Companion, Second Edition*. Addison-Wesley, 2004.
- [13] Sebastian Rahtz và Heiko Overdiek. Gói hyperref. CTAN:/macros/latex/contrib/hyperref.
- [14] Hideo Umei. Gói geometry. CTAN:/macros/latex/contrib/geometry.
- [15] Herbert Voß. Trang chủ PSTricks. <http://pstricks.tug.org>.
- [16] Timothy Van Zandt et al. Gói PSTricks, v1.07, 2005/05/06. CTAN:/graphics/pstricks.