



**CZECH TECHNICAL
UNIVERSITY
IN PRAGUE**



**Faculty of Electrical Engineering
Department of Mathematics**

CTUslides

**simple slides in CTUstyle design
using OpTeX**

Petr Olšák

petr@olsak.net

<http://petr.olsak.net/ctustyle.html>



Basics

- The document is included in a file (say `file.tex`) and it can be processed by `optex` file command.

- The header of the document should be:

```
\input ctustyle3 % CTUstyle macro for OpTeX
\slides          % slides initialization
\worktype[B/EN]  % type of the work (B,M,D,O) and language (CZ,SK,EN)
\faculty{F3}     % the faculty in short
\department {Department of Mathematics} % department

\slideshow       % begin of the document
... document ...
\pg.
```

- The document must be finished by `\pg` followed by period.
- You need OpTeX in the version Jun 2020 or newer.
See <http://petr.olsak.net/optex>.
- The work type should be set similarly as in **CTUstyle**.
- Only `\worktype`, `\faculty` and `\department` work here. No more declaration sequences from **CTUstyle**.



The structural commands

- You can type `*` for starting of the item.
- Nested items lists (second and more level) are created in the `\beginitems...\enditems` environments.
- The slide titles are created by `\sec Text` followed by end of line. For subsections, you can use `\secc Text` similarly.
- The title page (first slide) can be special if `\tit Title` (followed by end of line) is used here.
- The `\subtit Author name etc.` (followed by end of line) can be used after `\tit` at the first slide.
- The paragraph texts are ragged right.
- You can use `\nl` for new line in the paragraph or titles.
- You can use `\pg` followed by `+` or `;` or `.` for new slide.
- The page-bar in the right corner is clickable and it will be created correctly after second pass of the \TeX run.



Next page (next slide)

- The control sequence `\pg` must be followed by:
 - the character `+` – next page keeps the same text and a next text is added (usable for partially uncovering of ideas),



Next page (next slide)

- The control sequence `\pg` must be followed by:
 - the character `+` – next page keeps the same text and a next text is added (usable for partially uncovering of ideas),
 - the character `;` – normal next page,



Next page (next slide)

- The control sequence `\pg` must be followed by:
 - the character `+` – next page keeps the same text and a next text is added (usable for partially uncovering of ideas),
 - the character `;` – normal next page,
 - the character `.` – the end of the document.



Next page (next slide)

- The control sequence `\pg` must be followed by:
 - the character `+` – next page keeps the same text and a next text is added (usable for partially uncovering of ideas),
 - the character `;` – normal next page,
 - the character `.` – the end of the document.
- Summary:

```
\pg+    ... uncover next text  
\pg;    ... next page  
\pg.    ... the end of the document
```



Next page (next slide)

- The control sequence `\pg` must be followed by:
 - the character `+` – next page keeps the same text and a next text is added (usable for partially uncovering of ideas),
 - the character `;` – normal next page,
 - the character `.` – the end of the document.
- Summary:
 - `\pg+` ... uncover next text
 - `\pg;` ... next page
 - `\pg.` ... the end of the document
- If the control sequence `\slideshow` is removed (or commented out) from the beginning of the document then `\pg+` sequences are deactivated. This is usable for printing version of the document.



Verbatim

Verbatim in paragraph

- Unlike **CTUstyle** for OPmac, you can use `"code text"` inside paragraph directly.
- If you declare `\activettchar`` before `\slides` then you can use ``code text`` like in Markdown.
- You can use `\code{text}` too.
- All these features are described in OpT_EX documentation.



Verbatim

Verbatim in paragraph

- Unlike **CTUstyle** for OPmac, you can use `"code text"` inside paragraph directly.
- If you declare `\activettchar`` before `\slides` then you can use ``code text`` like in Markdown.
- You can use `\code{text}` too.
- All these features are described in OpT_EX documentation.

Multi-line verbatim

- Unlike **CTUstyle** for OPmac, you can use the pair `\begtt... \endtt` directly as described in OpT_EX documentation. No `\pg=` is needed.
- Of course, you can use `\verbinut` too, if you want.



Example of multi-line verbatim

The source code includes:

```
\begtt \hisyntac{C}  
#include <stdio.h>  
int main(); // This is a program in C  
{  
    printf("Hello world!\n");  
}  
\endtt
```



Example of multi-line verbatim

The source code includes:

```
\begtt \hisyntac{C}  
#include <stdio.h>  
int main(); // This is a program in C  
{  
    printf("Hello world!\n");  
}  
\endtt
```

and the result is:

```
#include <stdio.h>  
int main(); // This is a program in C  
{  
    printf("Hello world!\n");  
}
```

Note that local declarations can be inserted after `\begtt`, the `\hisyntax` declaration is used in the example here.



Limits of the `\pg+` sequence

- The `\pg+` sequence cannot be used inside a group.
- The exception is the nested environment `\beginitems...\enditems`.



Limits of the `\pg+` sequence

- The `\pg+` sequence cannot be used inside a group.
- The exception is the nested environment `\beginitems...\enditems`.

What to do?

- If you need to partially uncover the multi-line verbatim then you can use:

```
\begtt
... first line of the code ...
\endtt
\par\pg+ \vskip-6.75pt
\begtt
... second line of the code ...
\endtt
```



Limits of the `\pg+` sequence

- The `\pg+` sequence cannot be used inside a group.
- The exception is the nested environment `\begitems...\enditems`.

What to do?

- If you need to partially uncover the multi-line verbatim then you can use:

```
\begtt
... first line of the code ...
\endtt
\par\pg+ \vskip-6.75pt
\begtt
... second line of the code ...
\endtt
```

- If you need to uncover the texts more ingenious then you can use `\layers...\endlayers` environment (see next slide...)



Uncovering by `\layers`, `\pshow`

- You can declare layers inside a slide by `\layers n ... \endlayers` pair. The number n declares the number of layers.
- The page with `\layers ... \endlayers` pair is repeated n -times.
- You can use `\pshow k` inside `\layers ... \endlayers` environment. This macro means *partially show*. It prints the following text to the end of current group:
 - invisible, if the number of the slide layer is less than k ,
 - red, if the number of slide the layer is equal to k ,
 - normal (black), if the number of slide layers is greater than k .
- The verbatim text and `\secc` macros cannot be used inside `\layer` environment.
- See OpTeX documentation for more information.
- Next slide shows the usage of `\pshow`.



Example of `\pshow` usage

Ideas in special order

- **First idea**



```
\secc Ideas in special order
```

```
\layers 3
* {\pshow1 First idea}
* {\pshow3 Second idea}
* {\pshow2 Third idea}
\endlayers
\pg+
```

```
\secc A formula
```

```
\layers 4
Consider
$$
E = {\pshow2 m}{\pshow3 c^2}
$$
\endlayers
That is great!
\pg;
```



Example of \pshow usage

Ideas in special order

- First idea
-
- Third idea

```
\secc Ideas in special order
```

```
\layers 3
* {\pshow1 First idea}
* {\pshow3 Second idea}
* {\pshow2 Third idea}
\endlayers
\pg+
```

```
\secc A formula
```

```
\layers 4
Consider
$$
E = {\pshow2 m}{\pshow3 c^2}
$$
\endlayers
That is great!
\pg;
```



Example of `\pshow` usage

Ideas in special order

- First idea
- Second idea
- Third idea

```
\secc Ideas in special order
```

```
\layers 3
* {\pshow1 First idea}
* {\pshow3 Second idea}
* {\pshow2 Third idea}
\endlayers
\pg+
```

```
\secc A formula
```

```
\layers 4
Consider
$$
E = {\pshow2 m}{\pshow3 c^2}
$$
\endlayers
That is great!
\pg;
```



Example of \pshow usage

Ideas in special order

- First idea
- Second idea
- Third idea

A formula

Consider

$$E =$$

That is great!

```
\secc Ideas in special order
```

```
\layers 3
* {\pshow1 First idea}
* {\pshow3 Second idea}
* {\pshow2 Third idea}
\endlayers
\pg+
```

```
\secc A formula
```

```
\layers 4
Consider
$$
E = {\pshow2 m}{\pshow3 c^2}
$$
\endlayers
That is great!
\pg;
```



Example of \pshow usage

Ideas in special order

- First idea
- Second idea
- Third idea

A formula

Consider

$$E = m$$

That is great!

```
\secc Ideas in special order
```

```
\layers 3
* {\pshow1 First idea}
* {\pshow3 Second idea}
* {\pshow2 Third idea}
\endlayers
\pg+
```

```
\secc A formula
```

```
\layers 4
Consider
$$
E = {\pshow2 m}{\pshow3 c^2}
$$
\endlayers
That is great!
\pg;
```



Example of \pshow usage

Ideas in special order

- First idea
- Second idea
- Third idea

A formula

Consider

$$E = mc^2$$

That is great!

```
\secc Ideas in special order
```

```
\layers 3
* {\pshow1 First idea}
* {\pshow3 Second idea}
* {\pshow2 Third idea}
\endlayers
\pg+
```

```
\secc A formula
```

```
\layers 4
Consider
$$
E = {\pshow2 m}{\pshow3 c^2}
$$
\endlayers
That is great!
\pg;
```



Example of \pshow usage

Ideas in special order

- First idea
- Second idea
- Third idea

A formula

Consider

$$E = mc^2$$

That is great!

```
\secc Ideas in special order
```

```
\layers 3
* {\pshow1 First idea}
* {\pshow3 Second idea}
* {\pshow2 Third idea}
\endlayers
\pg+
```

```
\secc A formula
```

```
\layers 4
Consider
$$
E = {\pshow2 m}{\pshow3 c^2}
$$
\endlayers
That is great!
\pg;
```



Tables, pictures

- Tables can be created by `\table` or `\ctable` macro.
- Pictures can be included by `\inspic` macro.
- See **CTUstyle** and OpTeX documentation for more details.
- The centering would be done by the `\centerline` macro.
- Example:



Tables, pictures

- Tables can be created by `\table` or `\ctable` macro.
- Pictures can be included by `\inspic` macro.
- See **CTUstyle** and OpTeX documentation for more details.
- The centering would be done by the `\centerline` macro.
- Example:

```
\centerline{\picw=5cm \inspic cmelak1.jpg }
```





Tables, pictures

- Tables can be created by `\table` or `\ctable` macro.
- Pictures can be included by `\inspic` macro.
- See **CTUstyle** and OpTeX documentation for more details.
- The centering would be done by the `\centerline` macro.
- Example:

```
\centerline{\picw=5cm \inspic cmelak1.jpg }
```



- You can use `\puttext` or `\putpic` macro for arbitrary positioning of texts or images.



Comparison CTUslides with Beamer*

The \LaTeX package Beamer gives much more features and many themes are prepared for Beamer, **but**

* <http://www.ctan.org/pkg/beamer>



Comparison CTUslides with Beamer*

The \LaTeX package Beamer gives much more features and many themes are prepared for Beamer, **but**

- the user of Beamer is forced to *program* his/her document using dozens of `\begin{foo}` and `\end{foo}` and many another programming constructions,
- on the other hand, plain \TeX gives you a possibility to simply *write* your document with minimal markup. The result is more compact.

* <http://www.ctan.org/pkg/beamer>



Comparison CTUslides with Beamer*

The \LaTeX package Beamer gives much more features and many themes are prepared for Beamer, **but**

- the user of Beamer is forced to *program* his/her document using dozens of `\begin{foo}` and `\end{foo}` and many another programming constructions,
- on the other hand, plain \TeX gives you a possibility to simply *write* your document with minimal markup. The result is more compact.
- You need to read 250 pages of doc for understanding Beamer,
- on the other hand, you need to read only ten slides** and you are ready to use **CTUslides**.

* <http://www.ctan.org/pkg/beamer>

** this eleventh slide isn't counted



Comparison CTUslides with Beamer*

The \LaTeX package Beamer gives much more features and many themes are prepared for Beamer, **but**

- the user of Beamer is forced to *program* his/her document using dozens of `\begin{foo}` and `\end{foo}` and many another programming constructions,
- on the other hand, plain \TeX gives you a possibility to simply *write* your document with minimal markup. The result is more compact.
- You need to read 250 pages of doc for understanding Beamer,
- on the other hand, you need to read only ten slides** and you are ready to use **CTUslides**.
- A notice for programmers: to create another individual typographical design for \LaTeX is much more complicated than to do the same in plain \TeX . And you need to seriously understand plain \TeX if you want to do something more complicated in \LaTeX .

* <http://www.ctan.org/pkg/beamer>

** this eleventh slide isn't counted



Thanks for your attention



Thanks for your attention

Questions?