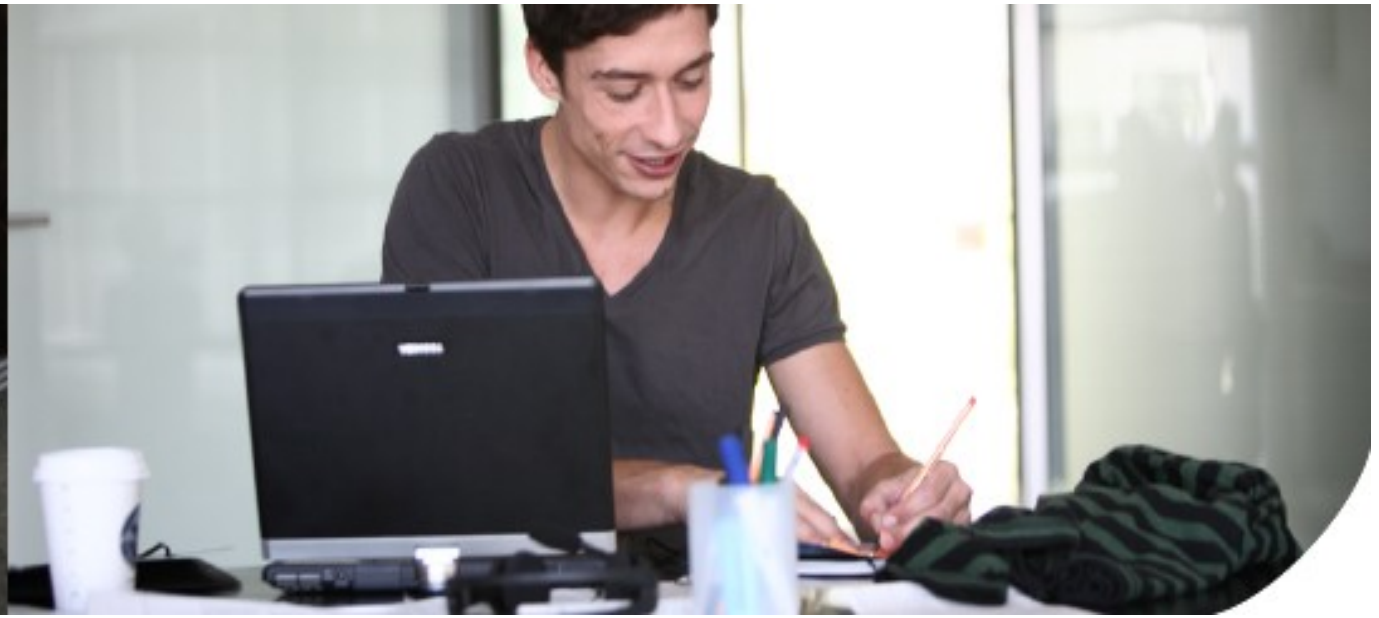
CIB GROUP

odpdown - markdown to slides

Nice slides from your favourite text editor!

Thorsten.Behrens@cib.de



Ideas

- auto-generate slides from *pure* text
  - like latex beamer, pandoc, or showoff
- make it blend nicely with existing slides and templates (Impress, PowerPoint etc)
  - have people **reuse** their corporations' materials 1:1

Previous Slide

```
## Ideas

* auto-generate slides from *pure* text

  * like latex beamer, pandoc, or
    [showoff]
(https://github.com/puppetlabs/showoff)

* make it blend nicely with existing slides
  and templates (Impress, PowerPoint etc)

  * have people **reuse** their
corporations'
    materials 1:1
```

Basic Markup

```
First Level Header
==================

Second Level Header
-------------------

This is a simple paragraph of text, written in a
plain text file and flown with fixed 50
characters
per line, like a plaintext email.

You can also write _emphasis_, or **strong**
emphasis, or even ***double-emphasis*** in your
text.

Here's how the slides look:
```

First Level Header

Second Level Header

This is a simple paragraph of text, written in a plain text file and flown with fixed 50 characters per line, like a plaintext email.

You can also write *emphasis*, or **strong** emphasis, or even ***double-emphasis*** in your text.

Blockquote Markup

```
## An alternative way to mark 2nd Level Headers

> This is a blockquote. You can write entire
paragraphs
> like that. Again, this is very similar to how
emails
> are formatted and written.
> Consider markdown inspired by plain-text email.
>
> And some more text in this blockquote, of course
you
> can again use _emphasis_, or **strong**
emphasis, or
> even ***double-emphasis*** in your text. But
wait,
> there's more!
>
```

CIB

An alternative way to mark 2nd Level Headers

> This is a blockquote. You can write entire paragraphs like that. Again, this is very similar to how emails are formatted and written. Consider markdown inspired by plain-text email.And some more text in this blockquote, of course you can again use *emphasis*, or **strong** emphasis, or even ***double-emphasis*** in your text. But wait, there's more!

List Markup

```
## Lists

Unordered (bulleted) lists use either
asterisks, plusses, or hyphens (*, +,
and -) interchangeably as list markers:

* item one
* item two
+ item three
- item four
```

Lists

Unordered (bulleted) lists use either asterisks, plusses, or hyphens (*, +, and -) interchangeably as list markers:

- item one
- item two
- item three
- item four

Nested List Markup

```
## Nested Lists

Nest lists by indenting subordinate
items:

* item one
  - item one.one
  * item one.two
+ item three
- item four
```

Nested Lists

Nest lists by indenting subordinate items:

- item one
  - item one.one
  - item one.two
- item three
- item four

Hyperlink Markup

```
## Links

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

This is [an example][1] reference-style
link.

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

Links

This is an example inline link.

This is an example reference-style link.

Advanced Markup

Embedded Image Markup

```
## Embed images

![This is alt text]
(http://upload.wikimedia.org/wikipedia/
commons/0/02/LibreOffice_Logo_Flat.svg
"This is an optional title for a direct
img")
```

Embed images

This is an optional title for a direct img

Referenced Images Markup

```
## Embed images via reference

![This is alt text][2]

[2]:
https://wiki.documentfoundation.org/ima
ges/8/87/LibreOffice_external_logo_600p
x.png  "This is an optional title
attribute for a ref img"
```

Embed images via reference

This is an optional title attribute for a ref img

Inline Code Markup

```
## Inline Code

You can do html-alike inline <code>
display with the following markup:
`$ tail -f /var/log/messages`
```

Inline Code

You can do html-alike inline <code> display with the following markup: `$ tail -f /var/log/messages`

Preformatted Content Markup

```
## Preformatted Content

    Just indent your text by four or more
spaces,
    to have it rendered in monospaced as
    pre-formatted content:

          --------
          |      |
          |      |
          \      /
           \    /
            \  /
             V
```

Preformatted Content

```
Just indent your text by four or more
spaces,
to have it rendered in monospaced as
pre-formatted content:
```

```
    --------
    |      |
    |      |
    \      /
     \    /
      \  /
       V
```

Syntax-Highlighted Code Markup

```
## Code - C++

By using a start/end marker as below (~~~ or ```), and specifying
one of the 100+ supported pygments language identifiers
(http://pygments.org/languages/):

~~~ c++
::basegfx::B2DPolyPolygon VeeWipe::operator () ( double t )
{
    ::basegfx::B2DPolygon poly;
    poly.append( ::basegfx::B2DPoint( 0.0, -1.0 ) );
    const double d = ::basegfx::pruneScaleValue( 2.0 * t );
    poly.append( ::basegfx::B2DPoint(
           0.0, d - 1.0 ) );
    poly.append( ::basegfx::B2DPoint(
           0.5, d ) );
    poly.append( ::basegfx::B2DPoint(
           1.0, d - 1.0 ) );
    poly.append( ::basegfx::B2DPoint(
           1.0, -1.0 ) );
    poly.setClosed(true);
    return ::basegfx::B2DPolyPolygon( poly );
}
~~~
```

Code - C++

```
::basegfx::B2DPolyPolygon VeeWipe::operator () ( double t )
{
    ::basegfx::B2DPolygon poly;
    poly.append( ::basegfx::B2DPoint( 0.0, -1.0 ) );
    const double d = ::basegfx::pruneScaleValue( 2.0 * t );
    poly.append( ::basegfx::B2DPoint(
            0.0, d - 1.0 ) );
    poly.append( ::basegfx::B2DPoint(
            0.5, d ) );
    poly.append( ::basegfx::B2DPoint(
            1.0, d - 1.0 ) );
    poly.append( ::basegfx::B2DPoint(
            1.0, -1.0 ) );
    poly.setClosed(true);
    return ::basegfx::B2DPolyPolygon( poly );
}
```

Or the same for syntax-highlighting Python

```
## Code - Python

~~~ python
# helper for ODFFormatter and ODFRenderer
def add_style(document, style_family, style_name,
              properties, parent=None):
    """Insert global style into given document"""
    style = odf_create_style(style_family,
                             style_name,
                             style_name,
                             parent)
    for elem in properties:
        # pylint: disable=maybe-no-member
        style.set_properties(properties=elem[1],
                             area=elem[0])
    document.insert_style(style, automatic=True)
~~~
```

Code - Python

```
# helper for ODFFormatter and ODFRenderer
def add_style(document, style_family, style_name,
              properties, parent=None):
    """Insert global style into given document"""
    style = odf_create_style(style_family,
                             style_name,
                             style_name,
                             parent)
    for elem in properties:
        # pylint: disable=maybe-no-member
        style.set_properties(properties=elem[1],
                             area=elem[0])
    document.insert_style(style, automatic=True)
```

Or for Bash-script syntax-highlighting

```
## Code - Bash

~~~ bash
# sanity checks
which safecat > /dev/null 2>&1 || {
    echo "You need safecat for this!"
    exit 1
}

umask 077

# enqueue mail, and params.
QUEUE_NAME=`safecat $BASE_DIR/tmp $BASE_DIR/mails`
if [ $? -eq 0 ]; then
    echo -e "$QUEUE_NAME\n$@" | \
      safecat $BASE_DIR/tmp $BASE_DIR/queue 1>/dev/null && \
      exit 0
    rm $BASE_DIR/mails/$QUEUE_NAME
fi

exit 1
~~~
```

Code - Bash

```
# sanity checks
which safecat > /dev/null 2>&1 || {
    echo "You need safecat for this!"
    exit 1
}

umask 077

# enqueue mail, and params.
QUEUE_NAME=`safecat $BASE_DIR/tmp $BASE_DIR/mails`
if [ $? -eq 0 ]; then
    echo -e "$QUEUE_NAME\n$@" | \
      safecat $BASE_DIR/tmp $BASE_DIR/queue 1>/dev/null && \
      exit 0
    rm $BASE_DIR/mails/$QUEUE_NAME
fi

exit 1
```

CIB

Want to know more?

https://github.com/thorstenb/odpgen/

```
git clone
https://github.com/thorstenb/odpgen/
```

Thanks for watching!

Q & A