Mass correspondence with Inkscape and Python

Ondřej Caletka

16th June 2019

About CESNET

About LinuxDays.cz

- something like PyCon
- a weekend in Prague
- 4+ talk tracks
- 150+ talks
- free entrance
- 1100+ attendees

cesnet

Badging attendees

- because it's cool
- you know who you talk to

But...

- people register at last moment
- nobody likes cutting the paper
- badges with lanyard usually hang damn too low
- if not double sided, they can be flipped over

cesnet

Avery Zweckform

- pre-cut forms printable by standard printers
- textile-based adhesive stickers
- up to 27 stickers per sheet
- 20 sheets for 30 €
- 18 stickers for 1 €

cesnet

The software

- web-based WYSIWYG app from the vendor
- only for US paper sizes
- looks functional until you try to use it
- wasted one day trying to get it to work
- even if it works (it didn't back then), it requires lots of human intervention

cesnet

In this workshop, we will:

1. design a machine-editable SVG badge in Inkscape
2. write a simple templating engine in Python
3. design a SVG template for a form
4. link templated badges into the templated forms
5. convert everything into one PDF

cesnet

Disclaimer

- I'm not a graphical designer
- Neither am I a Python programmer
- Yet I like to overthink solutions
- Some code is already done, feel free to use and improve it!

cesnet

Designing the badge

- draw a rectangle of desired size
- set document size to it
- use flowed text for placeholders
- adjust `id` attribute of the element containing the text
- accomodate the longest possible field value
- also design a dummy badge for hand-written registrations

cesnet

Templating the badges

1. parse XML template
2. read CSV database
3. edit text content of certain elements
4. save to a new XML document

cesnet

My solution

```
$ git clone https://github.com/LinuxDays/badge_card_generator.git
$ cd badge_card_generator
$ python3
>>> import svgtemplate
>>> tpl = svgtemplate.SVGTemplate("linuxdays-badge.svg", "./out")
>> from collections import namedtuple
>>> Record = namedtuple("Record", "name, role")
>>> r = Record("John Smith", "director")
>>> tpl.templatetext(r, "johnsmith.svg")
```

cesnet

Design the form

- there are MS Word templates available by the vendor
- place rectangles to the position of badges
- name them `pos0` – `posN`, first rows, then columns
- use script `template_to_linked.py` to change all `<rect/>` tags into `<use/>` tags

cesnet

Templating the forms

- similar to templating the badges
- just edit `href` attribute instead of text

cesnet

Ordering the badges

- top down first, then left to right
- support for multiple rows of sheets requires some clever reordering

```
def reordercards(cards, rows=4, sheetrows=9, sheetcols=3,
                 dummycard=""):
    """
    Reorder items in a way their ordering
    spans across ROWS rows of paper sheets.
    Fill empty positions with dummy cards.
    [0:8]   [36:44] [72:80]  <- first sheet
    [9:17]  [45:53] [81:89]  <- second sheet
    [18:26] [54:62] [90:98]
    [27:35] [63:71] [99:107]
    """
```

Almost done

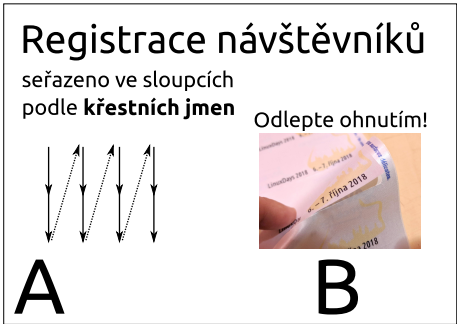
- we should have lots of SVG files, each containing one badge
- we should have a SVG file for each templated form sheet, linking to badge files

Batch mode of Inkscape

```
$ inkscape --shell
Inkscape 0.92.2 2405546, 2018-03-11
 interactive shell mode. Type 'quit' to quit.
> sheet01.svg -T -A=sheet01.pdf
> sheet02.svg -T -A=sheet02.pdf
> quit
$ pdfunite sheet*.pdf output.pdf
```

Printing

- watch out for “Scale to fit page” printer setting
- put some text to the edge of sheet, it should be cropped by the printer



cesnet

Thank you!

Ondřej Caletka
Ondrej.Caletka@cesnet.cz
https://**Ondřej.Caletka.cz**

Slides are already online.