J’ai fait mes slides !

2024-06-27

Guilhem Saurel

```latex
\documentclass[11pt]{report}
\usepackage[latin1]{inputenc}
\usepackage[francais]{babel}

\begin{document}
Hello Capitoul !
\end{document}
```

Beamer

```
\documentclass{beamer}

\title{my document title}
\author{Me}

\begin{document}
  \begin{frame}{my slide title}
    \begin{itemize}
      \item bla bla
      \item blah
    \end{itemize}
  \end{frame}
\end{document}
```

Theme

```
\defbeamertemplate*{frametitle}{laas}[1][] {
  \vskip.5cm%
  \begin{beamercolorbox}[wd=\paperwidth,ht=0cm]{frame
    \begin{tikzpicture}
      \useasboundingbox[fill=white](0,0) rectangle(1
      \node[inner sep=0pt] (laas) at (1,0.6) {
        \includegraphics[height=1cm]{LogoLAAS-2016.pn
      \fill[color=laasbleuclair] (1.98,0) rectangle(
      \fill[color=laasbleuanthracite] (2,0) rectangle
      \fill[color=white] (1.99,0.1) rectangle(12.65,
      \node[color=laasbleuclair,anchor=west,font=\la
        \insertframetitle };
    \end{tikzpicture}
  \end{beamercolorbox}
}
```

pandoc: markdown + yaml

```
---
title: my document title
author: me
---

# my slide title

- bla bla
- blah
```

Makefile

```
SOURCES = $(wildcard talks/*.md)
OUTPUTS = $(SOURCES:talks/%.md=public/%.pdf)

all: ${OUTPUTS}

public/%.pdf: talks/%.md references.bib
	pandoc -s \
	       -t beamer \
	       --citeproc \
	       --bibliography references.bib \
	       --highlight-style kate \
	       --pdf-engine xelatex \
	       --fail-if-warnings \
	       -o $@ $<
```

Watchexec

```
watch:
    watchexec -r -e md -c reset make -j
```

Index.html

```
from pathlib import Path


html = "<html><head><title>My talks</title></head>"
html += "<body><h1>My talks:</h1><ul>"

for f in sorted(Path("talks").glob("*.md")):
    html += f'<li><a href="{f.stem}.pdf">{f.stem}.pd

html += "</ul></html>"

with Path("public/index.html").open("w") as f:
    f.write(html)
```

pyyaml + jinja

```
<html>
    <head>
        <meta charset="utf-8">
        <title>My talks</title>
    </head>
    <body>
        <h1>My talks</h1>
        {% for pdf, head in talks %}
            <h2>{{ head.title }}</h2>
            {% if head.subtitle %}
                <h3>{{ head.subtitle }}</h3>
            {% endif %}
            <a href="{{ pdf }}">{{ pdf }}</a>
        {% endfor %}
    </body>
</html>
```

Tailwind CSS

```
<html>
    <head>
        <meta charset="utf-8">
        <title>My talks</title>
        <link href="./style.css" rel="stylesheet">
    </head>
    <body class="bg-indigo-50 text-indigo-950
            dark:bg-indigo-800 dark:text-indigo-50">
            <!-- contenu -->
    </body>
</html>
```

https://homepages.laas.fr/gsaurel/ https://nim65s.github.io/

Github Actions + Docker

```yaml
jobs:
  build:
    name: markdown → PDF
    runs-on: ubuntu-latest
    container:
      image: nim65s/talks
    steps:
      - uses: actions/checkout@v3
      - run: make
      - uses: actions/upload-pages-artifact@v1
        with:
          path: public/
```

Un setup simple

- latex + beamer + theme (texlive / miktex + bidouilles)
- markdown + yaml
- pandoc (haskell: cabal / stack)
- fontes (bidouilles)
- watchexec (rust: cargo install / cargo binstall)
- pdfpc (vala, gtk: cmake)
- makefile
- html
- python + pyyaml + jinja (pip / poetry / ...)
- tailwind css (npm / yarn / ...)
- github & gitlab + ci/cd
- docker
- ssh / rsync / git
- vim / emacs / ...

Notre seigneur et sauveur

Figure 1: Nix

Nix: paquet

```
stdenvNoCC.mkDerivation {
  nativeBuildInputs = [
    nodePackages.tailwindcss
    pandoc
    (python3.withPackages (p: [
      p.jinja2
      p.pyyaml
    ]))
    source-code-pro
    (texlive.combined.scheme-full.withPackages (_: [
      laas-beamer-theme
    ]))
  ];
  installPhase = "install -Dm 644 public/* -t $out";
}
```

Nix: flake

```
inputs.nixpkgs.url = "github:NixOS/nixpkgs";
outputs =
  inputs@{ flake-parts, ... }:
  flake-parts.lib.mkFlake { inherit inputs; } {
    perSystem = { pkgs, self', ... }: {
      packages.default = pkgs.callPackage ./talks.ni
      devShells.default = pkgs.mkShell {
        inputsFrom = [ self'.packages.default ];
        packages = [
          pkgs.pdfpc
          pkgs.watchexec
        ];
      };
    };
```

Questions ?

Cette présentatation

`https://gitlab.laas.fr/gsaurel/talks : slides.md`
`https://homepages.laas.fr/gsaurel/ slides.pdf`

Sous license

https://creativecommons.org/licenses/by-sa/4.0/

Bonus: Et si je veux patcher pandoc ?

```
--- a/LaTeX.hs  2024-06-27 09:32:42.946386964 +0200
+++ b/LaTeX.hs  2024-06-27 09:32:55.573589497 +0200
@@ -449,5 +449,10 @@
    let wrapColumns = if "columns" `elem` classes
        then \contents ->
-         inCmd "begin" "columns" <> brackets "T"
+         let fromPct xs =
+               case reverse xs of
+                 '%':ds -> '0':'.': reverse ds
+                 _      -> xs
+             pos = maybe "c" fromPct (lookup "pos" kv
+         in inCmd "begin" "columns" <> brackets (text
              $$ contents
              $$ inCmd "end" "columns"

pandoc.overrideAttrs { patches = [ ./columns.patch ]
```

Bonus: stalwart, nextcloud

```
{
  services = {
    stalwart-mail = {
      enable = true;
      settings.server.listener = {
        "smtp" = {
          bind = [ "[::]:25" ];
          protocol = "smtp";
        };
      };
    };
    nextcloud = {
      enable = true;
      package = pkgs.nextcloud29
      hostName = "localhost";
      config.adminpassFile = "/etc/nextcloud-pass";
```