kyasual : The modern beamer theme

-smart casual beamer theme-

January 27, 2023 Sample Univ. 1XXXXX Sample Author

Abstract

I propose the smart-casual beamer theme † **kyasual** † !

1. Introduction

2. Samples

3. How to Use?

4. Conclusion

Introduction

Purpose

- The conventional beamer template is not very pretty.

→ I design a **pretty** theme.

Proposition

1. **pretty**.
   - Color scheme like sweets
2. Using in **Lightning Talk**.
3. But I'd like to use in **math seminar** (≒**formal**).

→ No guile casual, pseudo casual. i.e. **kyasual**.

※ like **smart-casual**

How to Setup

Preparation

- **Fork** this repository.
- install **LaTeX** and **latexmk**.

Build

1. `mv sampleslide.tex (your slide title)`
2. `make`

→ (your slide title).pdf should be generated.

1. Introduction

2. Samples

3. How to Use?

4. Conclusion

Overview

This is Sample section.

1. text
2. itemize
3. block and box
4. code
5. math
6. image
7. tree

text sample

This is Sample Page.
You can make
new line
whenever you want to do.
You can typing about this match. If a character overflows, a new line is inserted by itself.
If you want to enphasys word, you can use **bold**, **alert**, or italic.
font :

- Main: GenshinGothic
- Italic: TimesItalic

日本語サンプル

これは日本語のサンプルページです．
好きなタイミングで
改行
できます．
一行あたりはだいたいこのくらい書けて，余ったら勝手に改行します．
強調は，**太字**と**アラート**があります．
フォント：源真ゴシック

itemize sample

- sipmle item
- ✓ ok item
- ✗ neg item
  - ※ annotate item
- → thus item
- ↔ but item

1. enumerate item 1
2. enumerate item 2
   2.1 enumerate subitem

block and box sample

block

block

example

example

alert

alert

simple box

code sample


```haskell
class Monad m where
    (>>=) :: m a -> (a -> m b) -> m b
    return :: a -> m a

instance Monad (King k) where
    f >>= m = State $ \s ->
        let (k', a) = runState f k
            in runState (m a) k'
```

KING MONAD

`inline`

math sample1

Def. 1 Sample

The pair (A, R) of any set A and the sum $R = \bigcup_{a \in I} \to_a$ of the binary relation $\to_\alpha$ defined on A is called Sample.

math sample 2

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4}}}} \tag{1}$$

$$\sqrt[n]{1 + x + x^2 + x^3 + \cdots + x^n}$$

tree sample

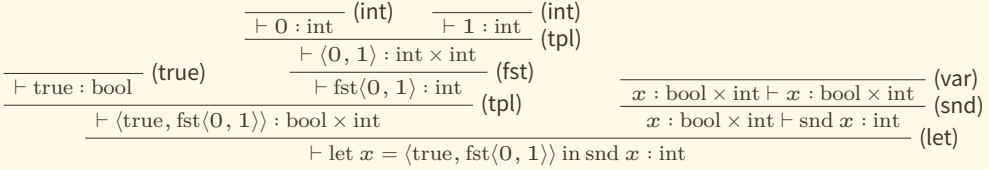
$\vdash 0 : \mathrm{int}$ (int)
$\vdash 1 : \mathrm{int}$ (int)
$\vdash \langle 0, 1 \rangle : \mathrm{int} \times \mathrm{int}$ (tpl)
$\vdash \mathrm{fst}\langle 0, 1 \rangle : \mathrm{int}$ (fst)
$\vdash \mathrm{true} : \mathrm{bool}$ (true)
$\vdash \langle \mathrm{true}, \mathrm{fst}\langle 0, 1 \rangle \rangle : \mathrm{bool} \times \mathrm{int}$ (tpl)
$x : \mathrm{bool} \times \mathrm{int} \vdash x : \mathrm{bool} \times \mathrm{int}$ (var)
$x : \mathrm{bool} \times \mathrm{int} \vdash \mathrm{snd}\ x : \mathrm{int}$ (snd)
$\vdash \mathrm{let}\ x = \langle \mathrm{true}, \mathrm{fst}\langle 0, 1 \rangle \rangle\ \mathrm{in}\ \mathrm{snd}\ x : \mathrm{int}$ (let)

image sample

1. Introduction

2. Samples

3. How to Use?

4. Conclusion

How to Use?

Basically, it is the same as an ordinary beamer.

- Or compare sampleslide.pdf and sampleslide.tex.
- If you wish, I will add documentation.
  - Please send your requests to issue or @kyawaway.

Features1: textbox


```
% using like other blocks

\begin{textblock}{textblock title1}
    textblock body1
\end{textblock}

\begin{textblock}{textblock title2}
    \begin{itemize}
        \item{textblock body2}
        \item{Usign with \alert{itemize
            },}
        \okitem{become beautiful}
    \end{itemize}
\end{textblock}
```

textblock title1

textblock body1

textblock title2

- textblock body2
- Using with **itemize**,
- ✓ become beautiful.

Features2: itemize icon


```
% using feature command
% in itemize environment

\begin{itemize}
    \okitem{ok}
    \negitem{neg}
        \begin{itemize}
            \thusitem{thus}
            \butitem{but}
        \end{itemize}
        \annoitem{annotate}
\end{itemize}

\begin{enumerate}
    \item{enum1}
    \item{enum2}
        \begin{enumerate}
            \item{enum2.1}
        \end{enumerate}
\end{enumerate}
```

- ✓ ok
- ✗ neg
  - → thus
  - ↔ but
- ※ annotate

1. enum1
2. enum2
   - 2.1 enum2.1

Features3: simplebox


```
\begin{simplebox}
    simple box
\end{simplebox}

\begin{simplebox}
    Not expected to be set up
        accordingly.
\end{simplebox}
```

simple box

Not expected to be set up accordingly.

1. Introduction

2. Samples

3. How to Use?

4. Conclusion

Conclusion

The theme finished up being very casual ....

- ✓ but I'm satisfied because it looks **good**:)