

A JÖNKÖPING UNIVERSITY BEAMER THEME

BASED ON METROPOLIS, WITH MANY CUSTOMIZATIONS

Marcel Bollmann

04.10.2021

✉ marcel.bollmann@ju.se

🏠 <https://marcel.bollmann.me/>



JÖNKÖPING UNIVERSITY
School of Engineering

A WORD OF WARNING

- This theme is **not officially endorsed** by Jönköping University.
- It does **not fully adhere** to JU's Graphic Manual.
- Use at your own discretion!

CONTENTS

1. Features

- How to use
- Theme options
- Layout

2. Colors

- Tints
- Usage

3. Fonts

- Selecting a font theme
- Samples

4. Environments

- Enumerations
- Quotations
- Blocks
- Code
- Bibliography

Features



- Copy the contents of the `sty/` directory into your project.
- If you downloaded this template from GitHub, you also need to [download the JU logotypes](#) from the intranet and extract the PNG versions into the `sty/img/` directory.
- You can then use this theme by including:

```
\usetheme[maincolor=purple]{metropolis-ju}
```

- Use LuaLaTeX for best results.

- The `maincolor` option sets the background colors of title/standout frames as well as the headline.

It can be either `purple` (default) or `grey`.

- This slide shows what the theme looks like with `maincolor=grey`.
- You can change colors in the middle of your presentation via:

```
\metroset{maincolor=grey}
```

Note: For changing the font theme, see the later “Fonts” section.

- The **titlelogo** option sets the logotype to put on the title frame. It can be either:
 - **none**: no logo
 - **ju**: JU's main logotype (**this is the default**)
 - **jth**: JTH's logotype
- The **headlogo** option does the same thing, but for the smaller logotype in the headline.
- Logotypes from the other schools aren't currently included, but could of course also be added.

The logotypes can be inserted with the following commands:

`\juLogo`



JÖNKÖPING UNIVERSITY

`\juLogoSmall`



Scaling options can be given as an argument, e.g. `\juLogo[height=2cm]`.

- To get the white versions of the logos, append `w` to the command; e.g. `\juLogoSmallw`.
- To get the JTH logos, use `\jthLogo...` equivalents.

In addition to all the options the standard METROPOLIS theme provides, I added the following options, implementing features from COLORFUL-DREAM:

- **footer=crumbs** will put the “breadcrumbs” line in the footer with the current section/subsection titles.
- **sectionpage=numbered** will produce the section pages with numbered circles.
- **subsectionpage=numbered** will do the same thing for subsections (*not enabled in this presentation*).

The frame title on the **colored bar** at the top of the slide comes from the METROPOLIS theme.

This can look nice, but **does not match JU's own presentation templates.**

However, you can also get something closer to JU's official style...

TYPESETTING THE FRAME TITLE

A frame *without* a title will look like this.

The heavy, all-uppercase heading can be produced via the command:

```
\juHeading{A title in JU style}
```

The automatic upper-casing might cause some issues with certain symbols or commands inside the heading. If that's the case, you can use this version without the upper-casing magic:

```
\juHeadingCased{A HEADING WITH ``QUOTES''}
```

It's nice to have less text per slide!

- Frames now have an additional **stretch** key with a stretch factor as an optional value (defaults to 2). It will **increase spacing** between paragraphs and list items.
- The idea is to make it easier to stretch slide contents, *without* littering the code with `\vspace{...}` commands everywhere.
- This slide uses `stretch=3.5`.

To quickly get a two-column layout, you can use:

```
\twocol{First column here.}{Second column here.}
```

By default, this will make both columns equally wide, namely `0.475\linewidth`.

An optional argument can be used to specify a different factor for the first column. Here, I used `0.3`.

The second column will automatically expand so that the two columns combined take up `0.95\linewidth`. *(This is less than 1 so that there is some padding between them.)*

Colors



This theme defines the following colors that you can use anywhere in your presentation. They are all based on the color values in JU's official Graphic Manual.

juPurple



juGrey



juYellow



juTurquoise



juDarkBlue



juBeige



Different color tints are also defined.

`juPurpleDarker`



`juPurpleFade1`



`juPurpleFade2`



`juPurpleFade3`



`juGreyFade1`



`juGreyFade2`



`juGreyFade3`



Color tints for the secondary colors.

juYellowFade1



juYellowFade2



juYellowFade3



juTurquoiseFade1



juTurquoiseFade2



juTurquoiseFade3



juDarkBlueFade1



juDarkBlueFade2



juDarkBlueFade3



Colors can be used with any commands, for example:

```
\textcolor{juYellow}{Lorem ipsum}  Lorem ipsum
```

Some predefined commands that use these colors:

<code>\alert{Lorem ipsum}</code>	Lorem ipsum
<code>\alertExample{Lorem ipsum}</code>	Lorem ipsum
<code>\highlight{Lorem ipsum}</code>	Lorem ipsum

The colors are also used in different predefined environments and blocks.

Fonts



JU's Graphic Manual uses **BentonSans** and **Scala (Sans) OT** as their main fonts, but they are not free and not available on Overleaf.

Therefore, this theme provides a number of alternatives:

- **arial** uses the recommended web font from JU's Graphic Manual, which is plain old Arial.
- **franklin** uses Libre Franklin, which supposedly is the closest free equivalent to BentonSans. (It doesn't support SMALL CAPS though.)
- **source** uses Adobe's Source fonts, which are slightly more different from BentonSans, but look potentially nicer.
- **fira** is the default font of the METROPOLIS theme, but has no similarity to JU's recommended fonts.

Select a font theme with the option `typeface=<...>`:

Typeface	Sans font	Mono font	Serif font
<code>arial</code>	Arial/Inter	Courier New	Times New Roman
<code>franklin</code>	Libre Franklin	Hack	Libre Baskerville
<code>source</code>	Source Sans Pro	Source Code Pro	Source Serif Pro
<code>fira</code>	Fira Sans	Fira Code/Mono	Charter

Fonts currently in use

`SourceSansProLight(0)`, `SourceCodePro(0)`, `SourceSerifProLight(0)`

Here's a sample of `typeface=arial`.

Sans (default)	The five boxing wizards jump quickly.
<code>\textit</code>	<i>The five boxing wizards jump quickly.</i>
<code>\textbf</code>	The five boxing wizards jump quickly.
<code>\textsc</code>	The five boxing wizards jump quickly.
<code>\extraboldsans</code>	The five boxing wizards jump quickly.
<code>\rmfamily</code>	The five boxing wizards jump quickly. The five <i>boxing</i> wizards jump quickly .
<code>\texttt</code>	The five boxing wizards jump quickly.

Serif font is not used anywhere by default.

The `\extraboldsans` variant is only used for the `\juHeading`.

Here's a sample of `typeface=franklin`.

Sans (default)	The five boxing wizards jump quickly.
<code>\textit</code>	<i>The five boxing wizards jump quickly.</i>
<code>\textbf</code>	The five boxing wizards jump quickly.
<code>\textsc</code>	The five boxing wizards jump quickly.
<code>\extraboldsans</code>	The five boxing wizards jump quickly.
<code>\rmfamily</code>	The five boxing wizards jump quickly. The <i>five boxing wizards</i> jump quickly.
<code>\texttt</code>	The five boxing wizards jump quickly.

Serif font is not used anywhere by default.

The `\extraboldsans` variant is only used for the `\juHeading`.

Here's a sample of `typeface=source`.

Sans (default)	The five boxing wizards jump quickly.
<code>\textit</code>	<i>The five boxing wizards jump quickly.</i>
<code>\textbf</code>	The five boxing wizards jump quickly.
<code>\textsc</code>	THE FIVE BOXING WIZARDS JUMP QUICKLY.
<code>\extraboldsans</code>	The five boxing wizards jump quickly.
<code>\rmfamily</code>	The five boxing wizards jump quickly. The <i>five boxing wizards</i> JUMP QUICKLY.
<code>\texttt</code>	The five boxing wizards jump quickly.

Serif font is not used anywhere by default.

The `\extraboldsans` variant is only used for the `\juHeading`.

Here's a sample of `typeface=fira`.

Sans (default)	The five boxing wizards jump quickly.
<code>\textit</code>	<i>The five boxing wizards jump quickly.</i>
<code>\textbf</code>	The five boxing wizards jump quickly.
<code>\textsc</code>	THE FIVE BOXING WIZARDS JUMP QUICKLY.
<code>\extraboldsans</code>	The five boxing wizards jump quickly.
<code>\rmfamily</code>	The five boxing wizards jump quickly. The five <i>boxing</i> wizards JUMP QUICKLY .
<code>\texttt</code>	The five boxing wizards jump quickly.

Serif font is not used anywhere by default.

The `\extraboldsans` variant is only used for the `\juHeading`.

Environments



Combine `\metroset{itemize=colored}` with
`\setbeamertemplate{enumerate items}[mycircle]` to get:

1 One

a alpha

i foo

ii bar

iii baz

b omega

2 Two

3 Three

“*Beautiful is better than ugly. Explicit is better than implicit. Simple is better than complex. [...] Readability counts.*”

— Tim Peters, from THE ZEN OF PYTHON

The quotation above can be produced via:

```
\begin{quote}  
  Beautiful is better than ugly.  
  Explicit is better than implicit.  
  Simple is better than complex. [\ldots]  
  Readability counts.  
\end{quote}  
\attribution{Tim Peters, from \textsc{The Zen of Python}}
```

These are beamer blocks with `block=transparent`.

This is a regular block.

Here is some content.

This is an alertblock.

Here is some content.

This is an exampleblock.

Here is some content.

This is a warningblock.

Here is some content.

These are beamer blocks with `block=fill`.

This is a regular `block`.

Here is some content.

This is an `alertblock`.

Here is some content.

This is an `exampleblock`.


Here is some content.

This is a `warningblock`.

Here is some content.

Since I frequently need to show code examples, I defined some styles and custom commands that can be included with:

```
\usepackage{myminted}
```

This uses the  **minted** package to typeset code with automatic syntax highlighting.

Important

Every frame that contains code must have the `[fragile]` option set, or compilation will break with many cryptic errors!

I also defined some commands to easily produce code blocks like the following; see the \LaTeX source for details:

```
1 // Your First Program
2
3 class HelloWorld {
4     public static void main(String[] args) {
5         System.out.println("Hello, World!");
6     }
7 }
```

Listing 1: A hello world program in Java.

I use **biblatex** with some custom definitions, which I also bundled in their own package:

```
\usepackage{mybiblatex}
```

Mainly, this will suppress output of URLs and DOIs, and instead turn the paper title into hyperlinks. Here's an example:

- Taichi Iki and Akiko Aizawa (Nov. 2020). “Language-Conditioned Feature Pyramids for Visual Selection Tasks”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 4687–4697

BIBLIOGRAPHY

- ❏ Iki, Taichi and Akiko Aizawa (Nov. 2020). “Language-Conditioned Feature Pyramids for Visual Selection Tasks”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 4687–4697.
- ❏ Krüger, Marcel and Font Awesome (2021). *The fontawesome5 package*. CTAN.
- ❏ Poore, Geoffrey M. (2017). *The minted package: Highlighted source code in L^AT_EX*. CTAN.

This slide also demonstrates that appendices work & play nicely with the other features!



JÖNKÖPING UNIVERSITY