

Beispiel-Titel

28. Oktober 2020 | Autor | PGI/JCNS-TA

INHALTSVERZEICHNIS

Zweiter Bildschirm

INHALTSVERZEICHNIS

Zweiter Bildschirm

Varianten dieser Präsentation

INHALTSVERZEICHNIS

Zweiter Bildschirm

Varianten dieser Präsentation

Grafiken

INHALTSVERZEICHNIS

Zweiter Bildschirm

Varianten dieser Präsentation

Grafiken

Codeboxen

INHALTSVERZEICHNIS

Zweiter Bildschirm

Varianten dieser Präsentation

Grafiken

Codeboxen

PDF-Reader

INHALTSVERZEICHNIS

Zweiter Bildschirm

Varianten dieser Präsentation

Grafiken

Codeboxen

PDF-Reader

GitLab CI

INHALTSVERZEICHNIS

Zweiter Bildschirm

Varianten dieser Präsentation

Grafiken

Codeboxen

PDF-Reader

GitLab CI

Vorlage verwenden

ZWEITER BILDSCHIRM

- \LaTeX -Beamer kann PDF-Seiten doppelter Breite mit zwei unterschiedlichen Versionen der Folien erzeugen

ZWEITER BILDSCHIRM

- \LaTeX -Beamer kann PDF-Seiten doppelter Breite mit zwei unterschiedlichen Versionen der Folien erzeugen
- ⇒ Version für Beamer und für den Laptop-Bildschirm (mit Vorschau, Notizen etc.)

ZWEITER BILDSCHIRM

- \LaTeX -Beamer kann PDF-Seiten doppelter Breite mit zwei unterschiedlichen Versionen der Folien erzeugen
- ⇒ Version für Beamer und für den Laptop-Bildschirm (mit Vorschau, Notizen etc.)
- Notizen können über den `\note`-Befehl eingefügt werden

VARIANTEN DIESER PRÄSENTATION

- presentation.pdf: Vortrags-Version ohne zweiten Bildschirm (→ für Adobe Reader)

VARIANTEN DIESER PRÄSENTATION

- presentation.pdf: Vortrags-Version ohne zweiten Bildschirm (→ für Adobe Reader)
- presentation_with_second_screen.pdf: Vortrags-Version mit Vorschau auf zweitem Bildschirm (→ für spezielle Vortragssoftware)

VARIANTEN DIESER PRÄSENTATION

- `presentation.pdf`: Vortrags-Version ohne zweiten Bildschirm (→ für Adobe Reader)
- `presentation_with_second_screen.pdf`: Vortrags-Version mit Vorschau auf zweitem Bildschirm (→ für spezielle Vortragssoftware)
- `presentation_with_notes_on_second_screen.pdf`: Vortrags-Version mit Notizen auf zweitem Bildschirm

VARIANTEN DIESER PRÄSENTATION

- `presentation.pdf`: Vortrags-Version ohne zweiten Bildschirm (→ für Adobe Reader)
- `presentation_with_second_screen.pdf`: Vortrags-Version mit Vorschau auf zweitem Bildschirm (→ für spezielle Vortragssoftware)
- `presentation_with_notes_on_second_screen.pdf`: Vortrags-Version mit Notizen auf zweitem Bildschirm
- `presentation_without_overlays.pdf`: Version ohne Overlays (→ Veröffentlichung der Folien)

VARIANTEN DIESER PRÄSENTATION

- `presentation.pdf`: Vortrags-Version ohne zweiten Bildschirm (→ für Adobe Reader)
- `presentation_with_second_screen.pdf`: Vortrags-Version mit Vorschau auf zweitem Bildschirm (→ für spezielle Vortragssoftware)
- `presentation_with_notes_on_second_screen.pdf`: Vortrags-Version mit Notizen auf zweitem Bildschirm
- `presentation_without_overlays.pdf`: Version ohne Overlays (→ Veröffentlichung der Folien)
- `handout.pdf`: Druckfassung ohne Notizen

VARIANTEN DIESER PRÄSENTATION

- `presentation.pdf`: Vortrags-Version ohne zweiten Bildschirm (→ für Adobe Reader)
- `presentation_with_second_screen.pdf`: Vortrags-Version mit Vorschau auf zweitem Bildschirm (→ für spezielle Vortragssoftware)
- `presentation_with_notes_on_second_screen.pdf`: Vortrags-Version mit Notizen auf zweitem Bildschirm
- `presentation_without_overlays.pdf`: Version ohne Overlays (→ Veröffentlichung der Folien)
- `handout.pdf`: Druckfassung ohne Notizen
- `handout_with_notes.pdf`: Druckfassung mit Notizen

GRAFIKEN

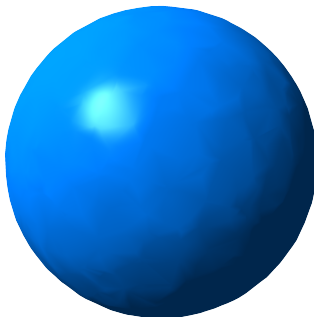
`\includegraphics`

- In `presentation.sty` werden verschiedene Hilfskommandos definiert

GRAFIKEN

`\includegraphics`

- In `presentation.sty` werden verschiedene Hilfskommandos definiert
- `\includegraphicsbeamer` sorgt dafür, dass Grafiken mit Transparenz verwendet werden können (s. `presentation_with_second_screen.pdf`)



- TikZ unterstützt Overlay-Befehle (`\uncover`, `\visible` etc.)

```
\begin{tikzpicture}[x={ (2, 0) }]  
  \uncover<.->{  
    \node[circle, draw, fill=AliceBlue] (a) at (0, 0) {A};  
  }  
  \uncover<.(1)->{  
    \node[circle, draw, fill=AliceBlue] (b) at (1, 0) {B};  
    \draw[>=latex, ->] (a) -- (b);  
  }  
\end{tikzpicture}
```

- TikZ unterstützt Overlay-Befehle (`\uncover`, `\visible` etc.)

```
\begin{tikzpicture}[x={ (2, 0) }]  
  \uncover<.->{  
    \node[circle, draw, fill=AliceBlue] (a) at (0, 0) {A};  
  }  
  \uncover<.(1)->{  
    \node[circle, draw, fill=AliceBlue] (b) at (1, 0) {B};  
    \draw[>=latex, ->] (a) -- (b);  
  }  
\end{tikzpicture}
```



- TikZ unterstützt Overlay-Befehle (`\uncover`, `\visible` etc.)

```
\begin{tikzpicture}[x={ (2, 0) }]  
  \uncover<.->{  
    \node[circle, draw, fill=AliceBlue] (a) at (0, 0) {A};  
  }  
  \uncover<.(1)->{  
    \node[circle, draw, fill=AliceBlue] (b) at (1, 0) {B};  
    \draw[>=latex, ->] (a) -- (b);  
  }  
\end{tikzpicture}
```



CODEBOXEN

- `\useboxbeamer` ergänzt Transparenz-Unterstützung ebenfalls bei Codeboxen

CODEBOXEN

- `\useboxbeamer` ergänzt Transparenz-Unterstützung ebenfalls bei Codeboxen
- Der Code muss hierzu vorher mit Hilfe einer `savebox` vordefiniert worden sein!

```
\newsavebox{\MintedUseboxbeamer}
\begin{lrbox}{\MintedUseboxbeamer}
\begin{minipage}{\itemizeonewidth}
\begin{minted}{tex}
\uncover<+>{%
    \useboxbeamer{\MintedUseboxbeamer}%
}%
\end{minted}
\end{minipage}
\end{lrbox}
```


PDF-READER

Adobe Reader

- Unterstützt alle Techniken für Animationen und Videos

PDF-READER

Adobe Reader

- Unterstützt alle Techniken für Animationen und Videos
- Nicht für Linux verfügbar

PDF-READER

Adobe Reader

- Unterstützt alle Techniken für Animationen und Videos
- Nicht für Linux verfügbar
- Unterstützt keinen zweiten Monitor

PDF-READER

pdfpc

- Unterstützt Videos, aber kein Flash oder Javascript

PDF-READER

pdfpc

- Unterstützt Videos, aber kein Flash oder Javascript
- Kann den zweiten Monitor für Notizen, Zeit oder eine Vorschau verwenden

PDF-READER

pdfpc

- Unterstützt Videos, aber kein Flash oder Javascript
- Kann den zweiten Monitor für Notizen, Zeit oder eine Vorschau verwenden
- Unterstützt den second screen einer Beamer-Präsentation:

```
pdfpc --notes=right <pdf-file>
```

PDF-READER

pdfpc

- Unterstützt Videos, aber kein Flash oder Javascript
- Kann den zweiten Monitor für Notizen, Zeit oder eine Vorschau verwenden
- Unterstützt den second screen einer Beamer-Präsentation:

```
pdfpc --notes=right <pdf-file>
```

- Für macOS bieten wir eine vorkompilierte Version an:

```
curl -O -L  
↪ https://pgi-jcns.fz-juelich.de/pub/Downloads/pdfpc.tar.gz  
sudo tar -C / -xovf pdfpc.tar.gz  
mkdir -p ~/local/bin  
ln -s /opt/homebrew/bin/pdfpc ~/local/bin/pdfpc  
echo 'export PATH="${HOME}/local/bin:${PATH}"' >> ~/.zshrc
```

PDF-READER

SplitShow

- Lösung für macOS

PDF-READER

SplitShow

- Lösung für macOS
- Unterstützt den second screen einer Beamer-Präsentation

PDF-READER

SplitShow

- Lösung für macOS
- Unterstützt den second screen einer Beamer-Präsentation
- Unterstützt keine Animationen oder Videos

- Lösung für macOS
- Unterstützt den second screen einer Beamer-Präsentation
- Unterstützt keine Animationen oder Videos
- Sehr reduzierter Funktionsumfang verglichen mit pdfpc

- Lösung für macOS
- Unterstützt den second screen einer Beamer-Präsentation
- Unterstützt keine Animationen oder Videos
- Sehr reduzierter Funktionsumfang verglichen mit pdfpc
- Download unter
<https://github.com/mpflanzer/splitshow/releases>

GITLAB CI

- Dieses Projekt enthält eine GitLab CI-Konfiguration
<https://iffgit.fz-juelich.de/imeyer/arbeit-template/blob/master/.gitlab-ci.yml>

GITLAB CI

- Dieses Projekt enthält eine GitLab CI-Konfiguration
`https://iffgit.fz-juelich.de/imeyer/arbeit-template/blob/master/.gitlab-ci.yml`
- Automatisches Erstellen der Arbeit und Folien bei jedem Push

GITLAB CI

- Dieses Projekt enthält eine GitLab CI-Konfiguration
`https://iffgit.fz-juelich.de/imeyer/arbeit-template/blob/master/.gitlab-ci.yml`
- Automatisches Erstellen der Arbeit und Folien bei jedem Push
- Fertige PDF-Dokumente werden mit GitLab Pages ausgeliefert
`<user>.iffgit.fz-juelich.de/<projekt>`

GITLAB CI

- Dieses Projekt enthält eine GitLab CI-Konfiguration
`https://iffgit.fz-juelich.de/imeyer/arbeit-template/blob/master/.gitlab-ci.yml`
- Automatisches Erstellen der Arbeit und Folien bei jedem Push
- Fertige PDF-Dokumente werden mit GitLab Pages ausgeliefert
`<user>.iffgit.fz-juelich.de/<projekt>`
- Anzeige der PDF-Dateien über pdfjs

VORLAGE VERWENDEN

- Der Code dieser Folien kann unter `https://iffgit.fz-juelich.de/imeyer/arbeit-template/blob/master/slides/content.tex` eingesehen werden

VORLAGE VERWENDEN

- Der Code dieser Folien kann unter `https://iffgit.fz-juelich.de/imeyer/arbeit-template/blob/master/slides/content.tex` eingesehen werden
- Die gesamte Vorlage für Arbeit + Folien kann über `https://iffgit.fz-juelich.de/imeyer/arbeit-template/-/archive/master/arbeit-template-master.tar.gz` heruntergeladen werden

VORLAGE VERWENDEN

- Der Code dieser Folien kann unter `https://iffgit.fz-juelich.de/imeyer/arbeit-template/blob/master/slides/content.tex` eingesehen werden
- Die gesamte Vorlage für Arbeit + Folien kann über `https://iffgit.fz-juelich.de/imeyer/arbeit-template/-/archive/master/arbeit-template-master.tar.gz` heruntergeladen werden
- Die README dieses Projektes beschreibt, wie man diese Vorlage für die eigene Arbeit / die eigenen Folien verwenden kann (`https://iffgit.fz-juelich.de/imeyer/arbeit-template/blob/master/README.md`)

VORLAGE VERWENDEN

- Der Code dieser Folien kann unter <https://iffgit.fz-juelich.de/imeyer/arbeit-template/blob/master/slides/content.tex> eingesehen werden
- Die gesamte Vorlage für Arbeit + Folien kann über <https://iffgit.fz-juelich.de/imeyer/arbeit-template/-/archive/master/arbeit-template-master.tar.gz> heruntergeladen werden
- Die README dieses Projektes beschreibt, wie man diese Vorlage für die eigene Arbeit / die eigenen Folien verwenden kann (<https://iffgit.fz-juelich.de/imeyer/arbeit-template/blob/master/README.md>)
- Verbesserungen bitte über Merge Requests vornehmen!