

Das 'sopra-documentation'-Paket

Dokumentation für das 'sopra-documentation'-Paket | Version v1.1.02

2. Dezember 2019


Florian Sihler (florian.sihler@uni-ulm.de)

1 Allgemeines

1.1 Warum, wieso, weshalb?

Dieses \LaTeX 2 ϵ -Paket wurde im Rahmen des Sopras im Wintersemester 2019 und Sommersemester 2020 verfasst und dient als Grundlage für die Erstellung von Dokumentationen für die anderen Pakete und Klassen des *Teams 20*, welche zusammen mit der `sopra-base.cls` kreiert werden.

Zum Visualisieren der einzelnen Code-Ausschnitte wird das `sopra-listings`-Paket verwendet. Es ist für die Verwendung des Pakets (`nolistings`) nicht relevant.

Das zugehörige Paket sollte ebenfalls in dieses Dokument eingebettet sein: .

1.2 Abhängigkeiten

Dieses Paket bindet die folgenden Paketen mit ein:

- `xcolor`
- `multicol`
- `blindtext`^(nur dummy)
- `attachfile2`
- `sopra-listings`^(nur listings)
- `hyperref`

All diese Pakete sollten Teil der gängigen \LaTeX -Distribution/mit diesem ausgeliefert worden sein.

1.3 Die Installation

Das Paket wird nicht als `.dtx` ausgeliefert, weswegen sich die folgenden Möglichkeiten ergeben:

- Das Paket kann in dasselbe Verzeichnis wie das Dokument gesetzt werden. In diesem Fall lautet die Einbindungsanweisung:

```
\usepackage{sopra-documentation}
```

- Das Paket kann in ein Unterverzeichnis/in ein mit dem Dokument ausgeliefertes Verzeichnis gelegt werden. In diesem Fall erfolgt die Angabe durch den (relativen-) Pfad:

```
\usepackage{./Mein/Pfad/zu/sopra-documentation}
```

- Man kann das Paket (mittels eines Symlinks oder ähnlichem) in einen eigenen `texmf`-Baum ablegen. So kann zum Beispiel auf Linux unter der Verwendung von `texlive` das Paket hier abgelegt werden: `~/texmf/tex/latex/`. Das Verzeichnis kann erstellt und anschließend mittels `texhash ~/texmf` aktualisiert werden. Nun kann das Paket wie jede andere installierte Paket verwendet werden:

```
\usepackage{sopra-documentation}
```

1.4 Weitere Besonderheiten

v1.0.0:

In dieser Version gibt es keine weiteren Besonderheiten.

v1.1.02:

In dieser Version wurde LILLYxLISTINGS durch sopra-listings ersetzt!

2 Paket-Konfiguration

2.1 Akzeptierte Parameter

Das Paket akzeptiert, so wie die meisten, Argumente. Bei Argumenten mit einer „Counter“-Option wird das jeweils standardmäßig aktive zuerst und das andere in Klammern geschrieben. So wird implizit:

```
\usepackage[nodebug,dummy,listings]{sopra-documentation}
```

aufgerufen. Während wir mit:

```
\usepackage[nolistings]{sopra-documentation}
```

das Dokument ohne das LILLYxLISTINGS-Paket kompilieren.

- ▷ nodebug (debug)

Im debug-Modus wird der Log, dank `\errorcontextlines` ausführlicher gefasst (im Falle eines Fehlers).

- ▷ dummy (nodummy)

Im dummy-Modus wird das Paket `blindtext` eingebunden, sonst nicht.

- ▷ listings (nolistings)

Im listings-Modus wird das Paket LILLYxLISTINGS eingebunden, sonst nicht.

3 Befehle- und Umgebungen

Es gilt zu beachten, dass das Präfix `env@` nur auf die Natur einer Umgebung hinweist und nicht zum eigentlichen Bezeichner zuzuordnen ist!

3.1 Kompatibilitätsmodus

Dieser Modus steht in v1.1.02 nicht zur Verfügung!

Diese Befehle werden nur mit `nolistings` definiert, da sie sonst von LILLYxLISTINGS eingebunden werden würden:

- ▷ `\T{Text}`

Wird hier als Alias für `\texttt` umgesetzt; setzt `Text` also in Schreibmaschinenschrift.

- ▷ `\lstcomment{Text}`

Analog existieren `\lstkwone`, `\lstkwtwo`, `\lstkwthree`, `\lstkwfour`, `\lstkwfive`, `\lstkwsix`, `\lststring` und `\lstnumber` die Text farblich hervorheben.

3.2 Generelle Befehle

▷ `\email{E-Mail}`

Setzt E-Mail als klickbare E-Mail-Adresse.

▷ `\thesodversion`

Liefert die aktuelle Version des Pakets. So ergibt: `\thesodversion`: v1.1.02

Hinweis: über `\value{sodversion}` lässt sich die Version als 4-stellige Nummer erhalten: 1102.

▷ `\manArg{Text}`

Erlaubt es in `env@command`^{→P. 3} oder vergleichbaren Umgebungen ein verpflichtendes Argument anzugeben. So ist Text in der Befehlsdefinition mittels `\manArg` gesetzt!

▷ `\optArg{Text}`

Erlaubt es in `env@command`^{→P. 3} oder vergleichbaren Umgebungen ein optionales Argument anzugeben.

▷ `\cmd{Name}`

Setzt Name als Befehl: `\cmd{Hi}`: `\Hi`.

▷ `\env{Name}`

Setzt Name als Umgebung: `\env{Hi}`: `env@Hi`.

▷ `\cmdref{Name}`

Analog zu `\cmd`^{→P. 3}, allerdings wird (so wie genau hier auch) ein Link zur Erklärung von Name erstellt.

▷ `\envref{Name}`

Analog zu `\env`^{→P. 3}, allerdings wird ein Link zur Erklärung von Name erstellt.

▷ `\argref{Name}{Counter}`

Setzt ein Argument mit entsprechender Counter-Option.

▷ `env@command{Name}{Argumente}`

Startet die Erklärung eines neuen Befehls mit Bezeichner Name (für `\cmdref`^{→P. 3}) und den Argumenten Argumente.

▷ `env@environment{Name}{Argumente}`

Startet die Erklärung einer neuen Umgebung mit Bezeichner Name (für `\envref`^{→P. 3}) und den Argumenten Argumente (so wie hier).

▷ `env@argument{Name}{Conter-Name}`

Startet die Erklärung eines neuen Arguments für das Paket/die Klasse mit Bezeichner `Name` und dem Konter-Bezeichner `Conter-Name`.

▷ `\say{Text}`

Setzt `Text` in „Anführungszeichen“.

▷ `\jmark[Text]{Linkziel}`

Funktioniert analog zum Pendant in Lilly, weil ich es so lieber mag und aus Reflex schreibe :D.

3.3 Weitere Befehle

Diese Befehle entstammen den eingebundenen Paketen und werden lediglich konfiguriert.

▷ `\attachfile[Options]{Pfad}`

Bindet die Datei in `Pfad` ein. Als `Options` bietet sich zumindest `subject` an. So wird in dieser Dokumentation zum Beispiel das Paket mit folgender Zeile eingebunden:

```
\attachfile[subject={sopra-documentation.sty}]{sopra-documentation.sty}
```

▷ `\textattachfile[Options]{Pfad}{Text}`

Bindet die Datei in `Pfad` ein, macht aber anstelle eines Symbols `Text` zum klickbaren Link zum öffnen. Die Optionen sind analog zu `\attachfile` → p. 4.

Für alle anderen verwendeten Methoden empfiehlt es sich, den Quellcode selbst oder den Quellcode der Dokumentation näher zu betrachten.