

DAS LTX-TALK-THEME PAKET

VO.5.3 2026-10-05

Themen-Engine für LTX-TALK

Matthias WERNER¹

<https://github.com/werner-matthias/osglecture>

Das Paket gestaltet Präsentationen der Klasse LTX-TALK. Es bietet fertige Layout-, Farb- und Schrifthemes sowie Befehle zum Anpassen und Entwickeln eigener Themes. Es gehört zum [osglecture](#)-Bundle und ist unabhängig von dessen Dokumentklasse nutzbar.

Inhaltsverzeichnis

1. Einführung	2	4.4. Abschnittstitel als Frametitel .	11
1.1. Wer sollte diese Dokumentati-		4.5. Weitere Anpassungen	11
on lesen?	2	4.6. Slotnamen der mitgelieferten	
		Themes	12
I. Nutzerhandbuch	3	5. Barrierefreiheit	12
2. Schnelleinstieg	3	5.1. Lesbare Gestaltung	12
3. Themes verwenden	4	5.2. Grafiken und Logos	13
3.1. Reihenfolge und Gültigkeit . .	4	5.3. Tagging und Prüfung	13
3.2. Mitgelieferte Themes	4	II. Eigene Themes entwickeln	13
3.2.1. Vergleich der mitge-		6. Das Theme-Modell	13
lieferten Themes	6	6.1. Bereiche	13
4. Themes anpassen	8	6.2. Slots	14
4.1. Das Layout anpassen	8	6.3. Slottypen	14
4.2. Farben anpassen	9	6.4. Layout	14
4.3. Titel platzieren	10		

¹ matthias.werner@informatik.tu-chemnitz.de

1. Einführung

6.5. Gestaltung	14	8.3. Layout-Konfiguration	20
6.6. Dekoration von Bereichen . . .	14	8.4. Metadaten	22
7. Ein Theme schrittweise entwickeln	15	8.5. Navigation	23
7.1. Theme registrieren und Geometrie wählen	15	8.6. Inhaltselemente	23
7.2. Bereiche und Slots definieren .	15	8.7. Farbkonfiguration	24
7.3. Slots positionieren	16	8.8. Schriftkonfiguration	24
7.4. Slots gestalten	16	8.9. Inneres Theme	25
7.5. Bereiche dekorieren	17	8.10. Slot-Definition	26
7.6. Vorhandene Themes erweitern	17	8.11. Slot-Positionierung	27
7.7. Innere Elemente abstimmen .	18	8.12. Titelseiten	27
7.8. Vollständiges Beispiel	18	8.13. Slot-Gestaltung	28
7.9. Empfehlungen	18	8.14. Bereichsdekoration	29
8. Referenz	20	8.15. Titelplatzierung	29
8.1. Paketoptionen	20	8.16. Optionen für Barrierefreiheit .	30
8.2. Theme-Verwaltung	20	8.17. Templates und Sockets	30
		Index	32

1. Einführung

LTX-TALK-THEME gestaltet Präsentationen der Klasse **LTX-TALK**: Kopf- und Fußzeilen, Randspalten, Titel, Farben, Schriften, Blöcke und Listenmarken. Wie bei Beamer können Sie ein fertiges Theme auswählen und Farb- oder Schriftthemes ergänzen. Eigene Themes beschreiben Bereiche und darin angeordnete Elemente, die hier *Slots* heißen. Verwenden Sie für die Beispiele Lua^LTeX und eine aktuelle Installation des Bundles einschließlich **osgstyler**. Das Paket ist zuletzt mit **ltx-talk** v0.6.10 vom 13. Oktober 2026 getestet und setzt mindestens v0.5.3 vom 5. August 2026 voraus. Ältere Klassenstände führen zu einem Fehler, neuere als der getestete zu einer Kompatibilitätswarnung. Die Klasse und das Themesystem befinden sich noch in Entwicklung. Laden Sie die Klasse zuerst; Themenpakete ersetzen keine bereits geladene Klasse.

1.1. Wer sollte diese Dokumentation lesen?

Diese Dokumentation richtet sich an zwei Zielgruppen. Anwender möchten ein vorhandenes Theme nutzen und gegebenenfalls an ihre Anforderungen anpassen. Themen-Autoren möchten dagegen ein eigenes Theme erstellen oder ein vorhandenes Theme erweitern.

Für erstere Gruppe ist Teil **I** gedacht. Wer dagegen ein eigenes Theme erstellen möchte, sollte anschließend auch den Teil **II** lesen. Die Referenz (Abschnitt 8) kann beiden Gruppen zum gezielten Nachschlagen dienen.

Teil I.

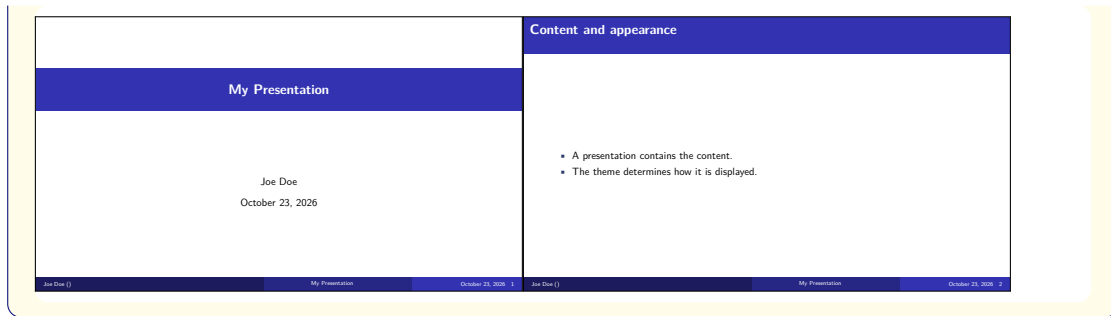
Nutzerhandbuch

2. Schnelleinstieg

Das folgende Beispiel zeigt eine kleine, aber vollständige und lauffähige Präsentation. Es lädt das mitgelieferte Theme magpie, erzeugt eine Titelfolie und eine weitere Folie mit einfachem Inhalt.

```
1 \DocumentMetadata{}
2 \documentclass{ltx-talk}
3
4 \usepackage{ltxtalk-theme-magpie}
5 \useltxtalktheme{magpie}
6
7 \title{My Presentation}
8 \author{Joe Doe}
9 \date{\today}
10
11 \begin{document}
12
13 \begin{frame}
14   \maketitle
15 \end{frame}
16
17 \begin{frame}
18   \frametitle{Content and appearance}
19   \begin{itemize}
20     \item A presentation contains the content.
21     \item The theme determines how it is displayed.
22   \end{itemize}
23 \end{frame}
24
25 \end{document}
```

3. Themes verwenden



3. Themes verwenden

Ein Theme wird in zwei Schritten ausgewählt. Zunächst lädt `\usepackage{ltxtalk-theme-maggie}` die Themendatei. Anschließend aktiviert `\useltxtalktheme{maggie}` das Thema für die Präsentation.

Die Befehle für Titel, Autor und Datum sowie die Umgebung `frame` gehören zur Dokumentklasse `ltx-talk`. Das Thema konfiguriert deren öffentliche Templates und legt damit fest, wo und in welcher Form diese Angaben auf den Folien erscheinen. Das Setzen und semantische Taggen der Inhalte bleibt Aufgabe der Klasse. Der eigentliche Inhalt bleibt dadurch unabhängig vom gewählten Thema.

Zum Wechseln des Erscheinungsbilds genügt es im einfachsten Fall, ein anderes mitgeliefertes Theme-Paket zu laden und den zugehörigen Namen an `\useltxtalktheme` zu übergeben. Welche Themes verfügbar sind und wie Teilthemes kombiniert werden, beschreibt der folgende Abschnitt. Wechseln Sie vollständige Themes zwischen zwei Frames; erneutes Aktivieren setzt die nachträglichen Slot- und Bereichsanpassungen zurück.

3.1. Reihenfolge und Gültigkeit

1. Pakete laden und das vollständige Theme aktivieren.
2. Farb- und Schriftthemes ergänzen.
3. Eigene Anpassungen zuletzt ausführen.

Ein erneutes `\useltxtalktheme` löscht unter anderem Slots, Bereichsdekorationen, Raster und benannte Layouts. `\addltxtalktheme` ergänzt die vorhandene Konfiguration. Slotdefinitionen, Slotpositionen und Rasteränderungen gelten global; eine Gruppe um einen Frame nimmt sie nicht zurück. Bei Layouts sind die Geometrieschlüssel lokal, ihr Aufbaucode kann jedoch globale Änderungen ausführen. Definieren Sie für wechselnde Layouts beide Zustände ausdrücklich und wenden Sie den passenden vor dem Frame an.

3.2. Mitgelieferte Themes

Die sechs Layout-Themes unterscheiden sich vor allem in der Aufteilung der Seite. `minimal` verwendet eine zurückhaltende Kopf- und Fußzeile. `maggie` zeigt den Folientitel in einem

3. Themes verwenden

farbigen Kopf und Metadaten in einem dreigeteilten Fuß. hawk ergänzt eine linke Navigationsspalte; sparrow ordnet Abschnitt und Unterabschnitt im Kopf an und setzt den Folientitel darunter. bluebird führt im Kopf die Abschnichtsliste, eine Punktreihe und den Fortschritt (siehe Abschnitt 8.5) und bringt ein alternatives Titel-Layout bluebird/band sowie ein Layout bluebird/plain ohne Kopf und Fuß mit. goose stellt rechts ein mitlaufendes Inhaltsverzeichnis auf; sein Layout goose/wide blendet die Leiste für breite Abbildungen aus. Die Namen der jeweiligen Theme-Pakete lauten `ltxtalk-theme-⟨Name⟩`.

Theme	Kennzeichnendes Layout	Anregung
minimal	schlichte Kopf- und Fußzeile	–
magpie	farbiger Kopf und Fuß	BEAMER Madrid
hawk	linker Navigationsbereich	BEAMER Hannover
sparrow	zweistufige Kopfnavigation	BEAMER Szeged
bluebird	Abschnittsnavigation und Fortschritt im Kopf	BEAMER Berlin
goose	rechtes Inhaltsverzeichnis	BEAMER Goettingen

anchovy und carp sind dagegen reine Farbthemes. Sie werden nach einem Layout-Theme mit `\addltxtalktheme` ergänzt und ändern nur die semantischen Farben. anchovy verwendet ein blau-goldenes, carp ein rotes Farbschema. Weil sie keine Bereiche oder Slots neu anlegen, bleibt das zuvor gewählte Layout erhalten.

```

1 \usepackage{ltxtalk-theme-minimal}
2 \usepackage{ltxtalk-theme-anchovy}
3 \useltxtalktheme{minimal}
4 \addltxtalktheme{anchovy}

```

Font-Themes werden ebenso additiv über ein Layout-Theme gelegt. Neben den einfachen Varianten `font-sans`, `font-serif` und `font-mixed` gibt es echte Schrift-Sippen: `font-libertinus`, `font-newcm` und `font-texgyre`. Jede Sippe besetzt semantische Rollen für Fließtext, Begleittext, Überschriften, Hervorhebungen, Code, Mathematik und Symbole mit aufeinander abgestimmten Familien. Alle verwendeten Schriften sind Bestandteil von T_EX Live.

```

1 \usepackage{ltxtalk-theme-sparrow}
2 \usepackage{ltxtalk-theme-font-libertinus}
3 \useltxtalktheme{sparrow}
4 \addltxtalktheme{font-libertinus}

```

Die Rollen heißen `body`, `support`, `heading`, `emphasis`, `code`, `math` und `symbol`. Sie können bei Slots mit `font-slot=heading` oder direkt mit `\UseLTXTalkFont{heading}` gewählt werden. Die

3. Themes verwenden

bisherigen portablen Namen `main`, `companion`, `display`, `accent` und `mono` bleiben als kompatible Synonyme erhalten. Der Mathematik-Slot wird nur als Ressource für eine spätere [unicode-math](#)-Anbindung registriert und nicht als Textschrift aktiviert.

3.2.1. Vergleich der mitgelieferten Themes

Die folgenden Beispielfolien verwenden bewusst denselben Inhalt. Dadurch werden Unterschiede in Satzspiegel, Navigation, Titelplatzierung und semantischen Farben unmittelbar vergleichbar. Die reinen Farbthemes werden über einem Layout gezeigt, das ihre Palette sichtbar macht: `carp` färbt `primary` um und wirkt daher unter `magpie` mit seinen großen Kopf- und Fußflächen am deutlichsten; `anchovy` legt das Gold in die Akzentslots und braucht ein Layout wie `sparrow`, das `secondary` für Nebentext nutzt.

```
1 \DocumentMetadata{}
2 \documentclass{ltx-talk}
3
4 \usepackage{ltxtalk-theme-minimal}
5 \usepackage{ltxtalk-theme-magpie}
6 \usepackage{ltxtalk-theme-hawk}
7 \usepackage{ltxtalk-theme-sparrow}
8 \usepackage{ltxtalk-theme-bluebird}
9 \usepackage{ltxtalk-theme-goose}
10 \usepackage{ltxtalk-theme-anchovy}
11 \usepackage{ltxtalk-theme-carp}
12
13 \title[Theme comparison]{Theme comparison}
14 \author[A. Example]{Ada Example}
15 \institute{Example Institute}
16 \date{2 September 2026}
17
18 \newcommand*\ThemeExampleFrame[1]{
19   \begin{frame}
20     \frametitle{#1}
21     \framesubtitle{The same content in every theme}
22     A short paragraph demonstrates type, text colour, and the text area.
23     \begin{itemize}
24       \item first list item
25       \item second item with an \alert{alert}
26     \end{itemize}
27     \begin{block}{Example block}
28       Block content with the formula  $a^2+b^2=c^2$ .
29     \end{block}
30   \end{frame}
31 }
32
33 \begin{document}
```

3. Themes verwenden

```
34 \section{First topic}
35 \subsection{An aspect}
36 \useltxtalktheme{minimal}
37 \ThemeExampleFrame{minimal}
38 \useltxtalktheme{magpie}
39 \ThemeExampleFrame{magpie}
40 \section{Second topic}
41 \useltxtalktheme{hawk}
42 \ThemeExampleFrame{hawk}
43 \useltxtalktheme{sparrow}
44 \ThemeExampleFrame{sparrow}
45 \useltxtalktheme{bluebird}
46 \ThemeExampleFrame{bluebird}
47 \section{Third topic}
48 \subsection{Its detail}
49 \useltxtalktheme{goose}
50 \ThemeExampleFrame{goose}
51 \useltxtalktheme{sparrow}
52 \addltxalktheme{anchovy}
53 \ThemeExampleFrame{sparrow + anchovy}
54 \useltxtalktheme{magpie}
55 \addltxalktheme{carp}
56 \ThemeExampleFrame{magpie + carp}
57 \end{document}
```

4. Themes anpassen

Theme comparison First topic minimal The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. A. Example	magpie The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. A. Example (Example Institute) Theme comparison 2 September 2024 2
Theme comparison hawk The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. A. Example	Second topic sparrow The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. Example Institute Theme comparison 4
First topic • Second topic • Third topic 2 / 3 bluebird The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. A. Example Theme comparison Example Institute 3	Its detail goose The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. A. Example Theme comparison First topic Second topic Third topic Its detail 3 / 3
Third topic Its detail sparrow + anchovy The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. Example Institute Theme comparison 2	magpie + carp The same content in every theme A short paragraph demonstrates type, text colour, and the text area. - first list item - second item with an alert Example block Block content with the formula $a^2 + b^2 = c^2$. A. Example (Example Institute) Theme comparison 4 September 2024 4

4. Themes anpassen

4.1. Das Layout anpassen

Nach der Aktivierung lassen sich die vom Theme gewählten Abmessungen mit `\ltxtalksetup` überschreiben. Dabei werden nur die angegebenen Eigenschaften geändert. Das folgende Beispiel vergrößert Kopf- und Fußbereich und schaltet den rechten Randbereich ab:

4. Themes anpassen

```
1 \useltxtalktheme{hawk}
2 \ltxtalksetup{
3   header-height = 2.8cm,
4   footer-height = 0.8cm,
5   right-margin = false
6 }
```

Die Schlüssel von `\ltxtalksetup` sind in Abschnitt 8.3 vollständig aufgeführt. Für einfache Anpassungen genügen meist die Schalter `header`, `left-margin`, `right-margin` und `footer` sowie die Längen `header-height`, `footer-height` und `margin-width`. Eine Änderung der Bereichsgröße verschiebt vorhandene Slots nicht automatisch. Größere Eingriffe sollten deshalb in einem eigenen Theme oder Teiltheme vorgenommen werden; Abschnitt 7 beschreibt das Vorgehen.

Ein benanntes Layout kann auch vorübergehend einen Randbereich abschalten. Im Theme `goose` kehren Sie anschließend ausdrücklich zu `goose/standard` zurück:

```
1 % Preamble:
2 \usepackage{ltxtalk-theme-goose}
3 \useltxtalktheme{goose}
4 \title{A presentation}
5 \author{A. Author}
6 % In the document:
7 \UseLTXTalkLayout{goose/wide}
8 \begin{frame}
9   \frametitle{A wide illustration}
10   Content using the extra width.
11 \end{frame}
12 \UseLTXTalkLayout{goose/standard}
```

4.2. Farben anpassen

Die acht semantischen Theme-Farben lassen sich mit `\setltxtalkcolors` neu belegen. Jeder Wert nennt, wie bei `\definecolor`, Farbmodell und Farbangabe, hier als `model=...`, `value=...`-Paar:

```
1 \setltxtalkcolors{
2   primary    = {model=HTML, value=005B50},
3   secondary  = {model=HTML, value=3C7A73},
4   accent     = {model=HTML, value=D6A800},
5   text       = {model=HTML, value=202020},
6   background = {model=HTML, value=FFFFFF},
```

4. Themes anpassen

```
7  structure = {model=HTML, value=005B50},  
8  alert     = {model=HTML, value=A00020},  
9  example   = {model=HTML, value=166534}  
10 }
```

primary prägt das Theme; secondary dient ergänzenden Angaben, accent hebt Details hervor. text und background bilden Vorder- und Hintergrund. structure, alert und example gestalten Listen und Blockvarianten. Die Farbnamen tragen das Präfix ltxtalk-, mit ltxtalk-bg für den Hintergrund. Die letzten drei Farben sind auch als structure, alert und example verfügbar. Änderungen wirken dort, wo das Theme diese Namen verwendet.

Nicht angegebene Farben werden in einer festen Kette abgeleitet: secondary, accent und structure erben von primary; text und background verwenden ihre neutralen Vorgaben, alert und example ihre semantischen Vorgaben. Eine abgeleitete Farbe wird beim Aufruf festgelegt; spätere Änderungen der Ausgangsfarbe berechnen sie nicht nachträglich neu.

Achtung!

`\setltxtalkcolors` ist eine *vollständige* Neudeklaration, kein inkrementelles Update: Nicht genannte Slots fallen nicht auf ihren vorherigen Wert zurück, sondern auf die übliche Vererbungskette (secondary etc. auf primary). Nur primary selbst ist zwingend anzugeben; fehlt es oder ist eine Angabe unvollständig, schlägt die gesamte Neudeklaration fehl und die zuvor gültigen Farben bleiben unverändert bestehen – nicht weil einzelne Slots verschont würden, sondern weil die fehlgeschlagene Deklaration insgesamt nichts ändert. Wer nur einen einzelnen Slot anpassen möchte, ohne die anderen erneut aufzuzählen, sollte deshalb den vollständigen letzten Aufruf wiederholen und nur den gewünschten Wert ändern.

4.3. Titel platzieren

Mit `\setltxtalktitleplacement` wird festgelegt, in welchem Bereich ein Theme Folien- und Abschnittstitel vorsieht. Ein Folientitel kann im Inhaltsbereich, im Kopf oder in beiden Bereichen erscheinen; ein Abschnittstitel kann im Kopf ein- oder ausgeblendet werden:

```
1 \setltxtalktitleplacement{  
2   frametitle = both,  
3   section = header  
4 }
```

Das Theme muss passende Slots für die gewählte Platzierung enthalten. Fehlt ein solcher Slot, erzeugt die Option allein kein zusätzliches sichtbares Element.

4.4. Abschnittstitel als Frametitel

Mit `\setltxtalkheadingbehavior{<auto-section-title = true>}` übernimmt die erste `\frametitle`-Folie nach `\section` oder `\subsection` deren nummerierte Überschrift automatisch als Titel – unabhängig davon, ob dasselbe Theme Abschnitte zusätzlich im Kopf zeigt (`\setltxtalktitleplacements` Schlüssel `section`). Wird an dieser Folie zusätzlich ein eigener Text angegeben, wandert er auf den Subtitel-Platz; ein daneben separat gesetzter `\framesubtitle` entfällt dann, weil der eigene Titel den Platz schon belegt:

```

1 \setltxtalkheadingbehavior{auto-section-title = true}
2 ...
3 \section{Introduction}
4 \begin{frame}
5   \frametitle{} % title becomes "1 Introduction"
6   ...
7 \end{frame}
8 \section{Related Work}
9 \begin{frame}
10  \frametitle{A Short History} % title becomes "2 Related Work",
11                                % subtitle becomes "A Short History"
12  ...
13 \end{frame}

```

Der Merker wird von genau einem `\frametitle`-Aufruf verbraucht: spätere Folien desselben Abschnitts sind unbeeinflusst, und folgen `\section` und `\subsection` ohne dazwischenliegende Folie aufeinander, gewinnt der zuletzt genannte (speziellere) Abschnitt. Ohne jeden `\frametitle`-Aufruf bleibt eine Folie wie bisher titellos; ein leeres `\frametitle{}` genügt, um allein den Abschnittstitel zu übernehmen.

4.5. Weitere Anpassungen

Einzelne vorhandene Slots können nach der Theme-Aktivierung mit `\styleltxtalkslot` umgestaltet werden. Dafür muss der vom Theme verwendete Slotname bekannt sein; er gehört deshalb zur Schnittstelle des jeweiligen Themes, nicht zur allgemeinen Anwenderschnittstelle. Ebenso kann `\decorateltxtalkarea` beispielsweise eine Linie über dem Footer ändern.

```

1 \usepackage{ltxtalk-theme-minimal}
2 \useltxtalktheme{minimal}
3 \styleltxtalkslot{min-ftitle}{
4   color = ltxtalk-primary,
5   font = \Large\bfseries,
6   align = left

```

```
7 }  
8 \decoration\texttalkarea{footer}{  
9   frame = top,  
10   rule-color = ltxtalk-accent,  
11   rule-thickness = 0.6pt  
12 }
```

Solche Anpassungen sollten unmittelbar nach `\uselxtalktheme` stehen, weil eine spätere erneute Theme-Aktivierung alle Slots und Dekorationen zurücksetzt. Für wiederverwendbare oder strukturelle Ergänzungen ist ein Teiltheme mit `\addlxtalktheme` geeigneter.

4.6. Slotnamen der mitgelieferten Themes

Die folgenden Namen können nach Aktivierung des jeweiligen Themes mit `\stylelxtalkslot` gestaltet werden. Kombinieren Sie das Präfix mit dem aufgeführten Suffix, etwa `min-ftitle`. Farb- und Schriftthemes fügen keine eigenen Slots hinzu.

minimal Präfix `min-`: title, section, page, ftitle, fsubtitle, author, footer-page.

magpie Präfix `magpie-`: frametitle, framesubtitle, author, title, page.

hawk Präfix `hawk-`: title, author, section, subsection, frametitle, framesubtitle.

sparrow Präfix `sparrow-`: section, subsection, frametitle, framesubtitle, institute, title, page.

bluebird Präfix `bluebird-`: navband, titleband, seclist, dots, progress, ftitle, fsub, footline1, footline2, fauthor, finst, ftitle-foot, fpage.

goose Präfix `goose-`: sidemeta, toc, progress, subs, ftitle, fsub.

5. Barrierefreiheit

Ein Theme verändert die visuelle Darstellung, darf aber die von `ltx-talk` erzeugte Dokumentstruktur nicht ersetzen. Folientitel sollten daher mit `\frametitle`, Untertitel mit `\framesubtitle` und Gliederungsebenen mit `\section` beziehungsweise `\subsection` ausgezeichnet werden. Rein optisch formatierter Text kann diese semantischen Befehle nicht ersetzen.

5.1. Lesbare Gestaltung

Bei eigenen Farbanpassungen müssen Vorder- und Hintergrund ausreichend kontrastieren. Farbe sollte außerdem nie das einzige Mittel sein, um eine Bedeutung auszudrücken. Kleine Schrift, sehr dünne Linien und dicht belegte Randbereiche sind auf Projektoren und bei starker Vergrößerung besonders problematisch. Entscheidend ist nicht nur die Titelfolie, sondern auch eine inhaltlich volle Standardfolie.

Text und wesentliche Grafiken sollten innerhalb der verfügbaren Bereiche bleiben. Eine Änderung von `margin-width`, `header-height` oder `footer-height` muss deshalb mit langen Titeln, mehrzeiligen Angaben und verschiedenen Seitenverhältnissen geprüft werden.

5.2. Grafiken und Logos

Für inhaltliche Grafiken gelten die Möglichkeiten von `ltx-talk` und den verwendeten Grafikpaketen; dort ist ein aussagekräftiger Alternativtext anzugeben. Wiederkehrende Logos, Farbflächen und Trennlinien sind gewöhnlich dekorativ und dürfen die Lesereihenfolge nicht mit wiederholten Informationen belasten. Die Slotrolle `artifact` kennzeichnet dekorative Elemente. Für ein inhaltliches Bild in einem Slot geben Sie einen Alternativtext mit `alt-text` an; siehe Abschnitt 8.16.

5.3. Tagging und Prüfung

Für getaggte PDF-Ausgabe sollte die Präsentation mit `\DocumentMetadata` beginnen. Das eigentliche Tagging von Frame-Inhalt und Überschriften übernimmt gegenwärtig überwiegend `ltx-talk`. `LTX-TALK-THEME` bewahrt diese Anbindung, indem Folientitel im Inhaltsbereich über die öffentlichen Klassenbefehle gesetzt werden. Die Optionen der Slot-Schnittstelle steuern zusätzlich die Rolle und den Alternativtext der Theme-Elemente im getaggten PDF (Abschnitt 8.16).

Eine Prüfung sollte deshalb sowohl visuell als auch strukturell erfolgen: mit langen und leeren Metadaten, mehreren Gliederungsebenen, fortgesetzten Folien und einem PDF-Prüfwerkzeug. Das Bundle enthält dafür `l3build-Tests`, die unter anderem die von `ltx-talk` erzeugte Überschriftenstruktur prüfen.

Teil II.

Eigene Themes entwickeln

6. Das Theme-Modell

Ein Theme besteht aus Bereichen, Slots und Gestaltung. Der Slottyp bestimmt den Inhalt; Bereich und Position bestimmen den Ort; die Slotgestaltung bestimmt Schrift, Farbe und Rahmen. Verwenden Sie die vorhandenen Typen für Titel und Metadaten, und `custom` für eigenen Inhalt. Die Klasse bleibt für Frames, Overlays und die semantische Struktur des Folieninhalts zuständig. Die semantische Rolle der Theme-Elemente legen Sie mit den Optionen für Barrierefreiheit fest (Abschnitt 8.16).

6.1. Bereiche

Ein Theme ordnet Seitenelemente den fünf Bereichen `header`, `left`, `content`, `right` und `footer` zu. Header, Ränder und Footer werden durch den Seitenstil `ltxtheme` ausgegeben. Der Inhalts-

bereich bleibt Teil des normalen Frame-Inhalts; dadurch bleiben Umbruch, Overlays und die semantische Einordnung durch `ltx-talk` erhalten.

header: Titel und Navigation		
left	content: Frame-Inhalt	right
footer: Metadaten und Foliennummer		

Die Abbildung zeigt nur die räumliche Schachtelung. Ein Theme darf jeden Bereich leer lassen, und mehrere Slots dürfen denselben Bereich belegen. Überlappungen werden nicht automatisch verhindert.

6.2. Slots

Ein Slot bezeichnet ein Element mit einer Aufgabe, nicht eine fest verdrahtete Stelle. Erst die Zuordnung zu einem Bereich, seine Position und sein Darstellungs-Template ergeben das konkrete Layout. Mehrere Slots desselben Typs sind zulässig. So kann etwa ein Folientitel im Header, im Inhaltsbereich oder in beiden Bereichen vorkommen.

6.3. Slottypen

Vordefinierte Typen liefern unter anderem Dokumenttitel, Autor, Institut, Abschnitt, Unterabschnitt, Folientitel, Folienuntertitel und Foliennummer. Der Typ `custom` bezieht seinen Inhalt aus `\setltxtalkslotcontent`; damit lassen sich beispielsweise Pfad- und Baumnavigationen oder frei gestaltete Logos ergänzen.

6.4. Layout

Jeder Slot besitzt eine horizontale und vertikale Ankerposition sowie Breite und Höhe. Symbolische Positionen wie `left`, `center`, `right`, `top` und `bottom` können mit absoluten Längen kombiniert werden.

6.5. Gestaltung

Für jeden Slot existiert eine Instanz des Templates `lttheme-slot`. Themes geben geeignete Vorgaben vor; Anwender können diese Instanzen mit den üblichen Template-Mitteln weiter bearbeiten. Schrift, Farbe, Hintergrund, Ausrichtung, Innenabstand und Regeln gehören damit zum Theme, nicht zum Seitenstil.

6.6. Dekoration von Bereichen

Bereichsdekorationen sind von den Slots unabhängig. Das Template `lttheme-area` kann insbesondere Hintergrundflächen und Trennlinien zeichnen. Eine obere Linie des Footers ist daher eine Eigenschaft des Footer-Bereichs und kein künstlicher Footer-Slot.

Der Seitenstil bildet die stabile Verbindung zu `LTX-TALK`. Sockets teilen seine Ausgabe zusätzlich in Hintergrund, die vier Seitenbereiche und Vordergrund. Ein vollständiges Theme muss diese Sockets nicht anfassen; sie sind Erweiterungspunkte für austauschbare Layoutphasen, nicht der Ersatz für die additive Slot-Schnittstelle.

7. Ein Theme schrittweise entwickeln

Die Entwicklung folgt am zuverlässigsten dieser Reihenfolge: zuerst mit `\ltxtalksetup` die benötigten Bereiche dimensionieren, dann Slots definieren und mit Testinhalten positionieren, anschließend Slots und Bereiche gestalten und zuletzt innere Elemente, Titelseite und Farben abstimmen. So bleiben geometrische und rein visuelle Fehler voneinander unterscheidbar.

7.1. Theme registrieren und Geometrie wählen

Ein Theme-Paket lädt zunächst `ltxtalk-theme`, definiert eine Setup-Anweisung und registriert diese unter einem eindeutigen Namen. Die Setup-Anweisung wird nicht beim Laden, sondern erst durch `\useltxtalktheme` ausgeführt. Dadurch können mehrere Theme-Pakete geladen und später gezielt aktiviert werden.

```

1 \RequirePackage{ltxtalk-theme}
2 \ExplSyntaxOn
3 \cs_new_protected:Npn \__my_theme_setup:
4 {
5   \ltxtalksetup{header=true,header-height=1.5cm}
6   % Slots, Positionen und Gestaltung
7 }
8 \RegisterLTXTalkTheme{my-theme}{\__my_theme_setup:}
9 \ExplSyntaxOff

```

Die Paketlogik kennt den Namen des Themes nicht. Bei der Aktivierung leert sie den bisherigen Theme-Zustand, führt die registrierte Setup-Anweisung aus, passt den Textbereich an und wählt den Seitenstil `ltxtheme`. Das Theme beschreibt daher ausschließlich Bereiche, Slots und deren Erscheinungsbild.

7.2. Bereiche und Slots definieren

Ein Slot wird mit Name, Bereich, Typ und Angaben zur Barrierefreiheit definiert. Der Name muss innerhalb des aktiven Themes eindeutig sein. Als Bereiche sind `header`, `left`, `content`, `right` und `footer` vorgesehen.

```

1 \defineltxtalkslot{my-section}{header}{section}{
2   role = navigation
3 }
4 \defineltxtalkslot{my-path}{right}{custom}{
5   role = navigation
6 }

```

```

7 \setltxtalkslotcontent{my-path}{
8   \thesection\quad\thesubsection
9 }

```

Die vordefinierten Typen title, author, date, institute, section, subsection, frametitle, framesubtitle und pagenummer beziehen ihren Inhalt aus der Klasse. Der Typ custom sowie unbekannte eigene Typen verwenden den mit `\setltxtalkslotcontent` hinterlegten Inhalt. Auch Logos werden zweckmäßig auf diese Weise mit explizitem Grafikcode belegt.

7.3. Slots positionieren

Nach der Definition erhält jeder sichtbare Slot eine Position und eine Größe:

```

1 \positionltxtalkslot
2   {my-section}{left}{center}{0.55\paperwidth}{0.8cm}
3 \positionltxtalkslot
4   {my-path}{center}{1cm}{2cm}{4cm}

```

Das zweite und dritte Argument bezeichnen die horizontale und vertikale Position innerhalb des Bereichs. Horizontal sind left, center und right, vertikal top, center und bottom zulässig. Stattdessen kann jeweils eine Länge ab der linken beziehungsweise oberen Bereichskante angegeben werden. Breite und Höhe sind Längen; ein Slot wird nicht automatisch vor Überlappungen mit anderen Slots geschützt.

7.4. Slots gestalten

`\styleltxtalkslot` bearbeitet die für den Slot angelegte Instanz des Templates lttheme-slot. So bleiben Inhalt, Position und Gestaltung voneinander getrennt.

```

1 \styleltxtalkslot{my-section}{
2   font = \large\bfseries,
3   color = white,
4   align = left,
5   padding = 2mm,
6   frame = box,
7   rule-color = ltxtalk-accent,
8   rule-thick = 0.8pt
9 }

```

Neben `font`, `color`, `align` und `padding` stehen `background`, `frame`, `rule-color`, `rule-thick` und `space-after` zur Verfügung. `frame` akzeptiert `none`, `box`, `top`, `bottom`, `left` und `right`. Die Schnittstelle kennt außerdem die Optionen `rotate`, `opacity`, `shadow`, `uppercase` und `lowercase`; diese haben derzeit keine sichtbare Wirkung. Auch einzelne Rahmenkanten werden bei Slots noch nicht gezeichnet; verwenden Sie dort `box` oder dekorieren Sie den gesamten Bereich.

7.5. Bereiche dekorieren

Hintergrundflächen und Trennlinien, die einen ganzen Bereich betreffen, gehören nicht in einen Slot. Sie werden mit `\decorationtalkarea` an der Instanz des Templates `lttheme-area` eingestellt:

```
1 \decorationtalkarea{footer}{  
2   background = white,  
3   frame = top,  
4   rule-color = ltxalk-primary,  
5   rule-thickness = 0.5pt,  
6   artifact-type = decorative  
7 }
```

Für `frame` sind zusätzlich `topbottom` und die üblichen Seiten zulässig. Die Dekoration wird vor den Slots des Bereichs gesetzt. Die Optionen `background-opacity`, `shadow`, `shadow-color`, `shadow-offset`, `margin-top` und `margin-bottom` sind bereits Teil der Schnittstelle, werden aber noch nicht vollständig gezeichnet.

7.6. Vorhandene Themes erweitern

Ein registriertes Theme kann auch nur zusätzliche Slots oder eine einzelne Dekoration enthalten. Nach der Aktivierung des Basisthemes fügt `\addltxalktheme{<Name>}` ein solches Teiltheme hinzu, ohne vorhandene Slots zu löschen. Das Beispiel `example-switching.tex` ergänzt auf diese Weise eine eigene Pfadnavigation. Das erneute Aktivieren eines Themes beginnt dagegen bewusst mit einem leeren Theme-Zustand.

```
1 \useltxalktheme{minimal}  
2 \addltxalktheme{course-navigation}
```

Ein Teiltheme sollte nur die Eigenschaften definieren, die es tatsächlich ergänzt. Insbesondere darf es weder den Theme-Zustand leeren noch selbst den Seitenstil wechseln. Slotnamen sollten einen paketspezifischen Präfix tragen, damit sie nicht mit denen des Basisthemes kollidieren.

7.7. Innere Elemente abstimmen

Ein vollständiges Theme sollte nicht bei Kopf, Rändern und Fuß enden. `\styleltxtalkblock` stimmt normale, hervorgehobene und Beispielblöcke auf die semantischen Farben ab; `\setltxtalkitemize` wählt dazu passende Listenmarken. Beide Befehle ändern nur die Darstellung der von `ltx-talk` bereitgestellten Strukturen.

7.8. Vollständiges Beispiel

Dieses Dokument fasst die Schritte zu einem kleinen Theme zusammen. Es setzt den Folientitel im Inhaltsbereich und die Foliennummer im Fuß. Der Registrierungscode kann später unverändert in ein eigenes Paket verschoben werden; dieses lädt das Kernpaket mit `\RequirePackage`.

```
1 \DocumentMetadata{}
2 \documentclass{ltx-talk}
3 \usepackage{ltxtalk-theme}
4 \RegisterLXTalkTheme{my-theme}{
5   \ltxtalksetup{footer=true, footer-height=0.7cm}
6   \setltxtalkcolors{primary={model=HTML, value=005B50}}
7   \defineltxtalkslot{my-title}{content}{frametitle}{}
8   \positionltxtalkslot{my-title}{0pt}{0pt}{\textwidth}{1cm}
9   \styleltxtalkslot{my-title}{
10     font-slot=heading, font=\Large\bfseries,
11     color=ltxtalk-primary, align=left
12   }
13   \defineltxtalkslot{my-page}{footer}{pagenumber}{}
14   \DeclareLXTalkAreaGrid{footer}{*, 2cm}{1}
15   \PositionLXTalkSlotInGrid{my-page}{2}{1}{1}{1}
16   \decoratelxtalkarea{footer}{
17     frame=top, rule-color=ltxtalk-primary
18   }
19 }
20 \useltxtalktheme{my-theme}
21 \begin{document}
22 \begin{frame}
23   \frametitle{A custom theme}
24   Content belongs to the presentation.
25 \end{frame}
26 \end{document}
```

7.9. Empfehlungen

Semantische Slottypen sind eigenen Typen vorzuziehen. Dekorative Elemente sollten als Artefakt, Navigation als solche und echte Überschriften mit einer passenden Ebene beschrieben

7. Ein Theme schrittweise entwickeln

werden. Wiederkehrende Seitenelemente dürfen den eigentlichen Frame-Inhalt und dessen Lesereihenfolge nicht duplizieren.

Themes sollten kurze und lange Inhalte, fehlende Metadaten, mehrzeilige Kopfzeilen, beide Randbereiche sowie Folien mit und ohne Untertitel abdecken. Für jedes Theme empfiehlt sich ein `l3build-Sichttest`; strukturelle Tests sollten zusätzlich die getaggte Überschriftenhierarchie prüfen. Theme-spezifische Logik bleibt im Theme-Paket, während `LXTALK-THEME` nur die allgemeinen Bereiche, Slots, Templates und den Seitenstil bereitstellt.

8. Referenz

8.1. Paketoptionen

Geben Sie Optionen des Kernpakets vor dem ersten Theme-Paket an:

```
1 \usepackage[block-hairlines=true]{ltxtalk-theme}
2 \usepackage[shadow=true]{ltxtalk-theme-magpie}
3 \useltxtalktheme{magpie}
```

outer-only = `true`|`false` Voreinstellung: `false`
 Deaktiviert die Schriftpalettenverwaltung des Themes. Die Textrollen verwenden vorhandene LaTeX-Schriftfamilien. Eine Schriftpalette zu aktivieren erzeugt dann nur eine Warnung. Diese Option ist nur beim Laden verfügbar; die Anforderungen der Dokumentklasse gelten weiterhin.

block-hairlines = `true`|`false` Voreinstellung: `false`
 Ergänzt bei der nächsten Theme-Aktivierung dünne Blocklinien. Auch über `\ltxtalksetup` vor `\useltxtalktheme` einstellbar.

shadow = `true`|`false` Voreinstellung: `false`
 Option ausschließlich von `ltxtalk-theme-magpie`: aktiviert Schatten unter den Blöcken.

8.2. Theme-Verwaltung

`\useltxtalktheme{⟨Name⟩}`
 Aktiviert das registrierte Theme `⟨Name⟩`. Der bisherige Theme-Zustand wird vollständig geleert. `default` stellt den Seitenstil der Klasse ohne `lththeme`-Slots wieder her.

`\addltxtalktheme{⟨Name⟩}`
 Führt die Setup-Anweisung eines registrierten Themes zusätzlich zum aktiven Theme aus, ohne vorhandene Slots zu löschen. Der Befehl dient vor allem zum Hinzufügen von Teilthemes.

`\RegisterLTXTalkTheme{⟨Name⟩}{⟨Setup-Code⟩}`
 Registriert `⟨Setup-Code⟩` unter `⟨Name⟩`. Üblicherweise wird hier eine geschützte interne Anweisung des Theme-Pakets übergeben. Eine vorhandene Registrierung gleichen Namens wird ersetzt.

8.3. Layout-Konfiguration

`\ltxtalksetup{⟨Optionen⟩}`
 Ändert nur die angegebenen Eigenschaften und passt den Textbereich an. Die folgenden Vorgaben gelten vor der Theme-Aktivierung; ein Theme setzt eigene Werte.

`\DeclareLTXTalkLayout{⟨Name⟩}{⟨Optionen⟩}[⟨Aufbau-Code⟩]`

`\UseLTXTalkLayout{⟨Name⟩}[⟨Overrides⟩]`

`\DeclareLTXTalkLayout` registriert die Layoutkonfiguration.

`\UseLTXTalkLayout` aktiviert sie. *⟨Optionen⟩* sind Schlüssel aus `ltxtalk / layout`; der optionale *⟨Aufbau-Code⟩* kann Slots definieren, Flächen dekorieren oder eine Navigation binden. `\UseLTXTalkLayout` wendet erst die gespeicherten Optionen an, dann die *⟨Overrides⟩* desselben Schlüsselraums, dann den Aufbau-Code – der Code sieht also die endgültigen Werte (etwa `\LTXTalkHeaderRows`). Nicht genannte Schlüssel behalten ihren Wert, Layouts sind also additiv. Ein Theme darf mehrere Layouts deklarieren und pro Frame zwischen ihnen umschalten; `\UseLTXTalkLayout` darf unmittelbar vor einem frame oder als erste Anweisung darin stehen. Enthält der Aufbau-Code unbalancierte eckige Klammern, ist er zusätzlich in geschweifte Klammern einzuschließen.

`\LTXTalkBaseLeftMargin`

`\LTXTalkBaseRightMargin`

`\LTXTalkBaseTopMargin`

`\LTXTalkBaseBottomMargin`

`\LTXTalkCurrentLeftMargin`

`\LTXTalkCurrentRightMargin`

`\LTXTalkCurrentTopMargin`

`\LTXTalkCurrentBottomMargin`

Expandierbare Längen für die ursprünglichen beziehungsweise aktuell wirksamen Seitenränder. Theme-Code kann damit ohne Zugriff auf interne Variablen an der Textkante ausrichten.

`header = true | false`

Voreinstellung: false

Kopfbereich ein- oder ausschalten.

`left-margin = true | false`

Voreinstellung: false

Linker Randbereich ein- oder ausschalten.

`right-margin = true | false`

Voreinstellung: false

Rechter Randbereich ein- oder ausschalten.

`footer = true | false`

Voreinstellung: false

Fußbereich ein- oder ausschalten.

8. Referenz

<code>header-height = {⟨Länge⟩}</code> Höhe des Kopfbereichs.	Voreinstellung: 1.5cm
<code>footer-height = {⟨Länge⟩}</code> Höhe des Fußbereichs.	Voreinstellung: 1cm
<code>margin-width = {⟨Länge⟩}</code> Setzt beide Randbreiten und beide Inhaltseinzüge gleichzeitig. Nachfolgende Einzelschlüssel überschreiben diese Werte.	Voreinstellung: 2.5cm
<code>left-area-width = {⟨Länge⟩}</code> Breite des linken Randbereichs.	Voreinstellung: 2.5cm
<code>right-area-width = {⟨Länge⟩}</code> Breite des rechten Randbereichs.	Voreinstellung: 2.5cm
<code>content-inset-left = {⟨Länge⟩}</code> Linker Inhaltseinzug bei aktivem Randbereich.	Voreinstellung: 2.5cm
<code>content-inset-right = {⟨Länge⟩}</code> Rechter Inhaltseinzug bei aktivem Randbereich.	Voreinstellung: 2.5cm
<code>header-rows = {⟨Ganzzahl⟩}</code> Über <code>\LTXTalkHeaderRows</code> abrufbare Zeilenzahl für Layout-Aufbaucode. Ändert ein Bereichsraster nicht von selbst.	Voreinstellung: 1
<code>header-columns = {⟨Liste⟩}</code> Ältere Schnittstelle; ohne Wirkung auf frei positionierte Slots.	Voreinstellung: {0.2,0.6,0.2}
<code>footer-columns = {⟨Liste⟩}</code> Ältere Schnittstelle; verwenden Sie für neue Layouts Bereichsraster.	Voreinstellung: {0.33,0.34,0.33}
<code>block-hairlines = true false</code> Globale Voreinstellung für die nächste Theme-Aktivierung, keine Geometrieangabe.	Voreinstellung: false

8.4. Metadaten

`\LTXTalkMetadata{⟨Feld⟩}`

`\UseLTXTalkMetadata[⟨Optionen⟩]{⟨Feld⟩}`

Liefert Präsentationsmetadaten über eine öffentliche Schnittstelle. Verfügbare Felder sind `title`, `short-title`, `subtitle`, `short-subtitle`, `author`, `short-author`, `institute`, `short-institute`, `date`, `short-date`, `section`, `subsection`, `frametitle`, `framesubtitle`, `framenumber` und `totalframes`. Die zweite Form akzeptiert `lines=1`, um erzwungene Zeilenumbrüche durch `linebreak` (standardmäßig ein Leerzeichen) zu ersetzen.

`\LTXTalkMetadata` ist expandierbar. Verwenden Sie `\UseLTXTalkMetadata` zum Setzen mit folgenden Ausgabeoptionen:

`lines = all|1`

Voreinstellung: all

Behält Zeilenumbrüche bei oder ersetzt sie für einzeilige Ausgabe.

`linebreak = {⟨Ersatz⟩}`

Voreinstellung: Leerzeichen

Ersatz für erzwungene Umbrüche bei `lines=1`.

```
\UseLTXTalkMetadata[lines=1,linebreak={ -- }]{short-title}
```

8.5. Navigation

`\LTXTalkNavigation{⟨Name⟩}`

`\DeclareLTXTalkNavigation{⟨Name⟩}{⟨Code⟩}`

`\LTXTalkNavigation` setzt einen benannten Navigations-Renderer – gedacht als Slot-Inhalt im Aufbau-Code eines Layouts oder Themes. `\DeclareLTXTalkNavigation` registriert einen eigenen Renderer; `⟨Code⟩` wird bei jedem Aufruf ausgeführt und kann beispielsweise `\LTXTalkMetadata` verwenden.

Mitgeliefert:

- `current-section`, `current-subsection` – Titel des aktuellen Abschnitts bzw. Unterabschnitts; ein Durchlauf genügt.
- `section-list` – alle Abschnitte in Dokumentreihenfolge, nebeneinander.
- `subsection-list` – die Unterabschnitte des aktuellen Abschnitts, nebeneinander.
- `toc` – Abschnitte untereinander, unter dem aktuellen seine Unterabschnitte eingerückt.
- `section-dots` – ein Punkt je Abschnitt (aktueller gefüllt, zurückliegende gedämpft, folgende offen).
- `section-progress` – „`⟨Position⟩ / ⟨Anzahl⟩`“.

Der jeweils aktuelle Eintrag steht in `ltxtalk-primary`, die übrigen gedämpft.

Die Navigation verwendet die Abschnittsstruktur aus dem vorherigen LaTeX-Lauf. Alle Renderer außer `current-section` und `current-subsection` brauchen also zwei Läufe; ändert sich die Struktur, mahnt `lthème` einen weiteren Lauf an. Abschnittstitel mit Auszeichnung erscheinen als Klartext, da sie ein bewegliches Argument sind.

8.6. Inhaltselemente

`\InsertLTXTalkElement{⟨Name⟩}`

Setzt eine benannte Instanz des Template-Typs `ltxtalk-element` – gedacht für einzelne, formatierbare Inhaltswerte wie Seitenzahlen oder Abschnittsüberschriften, im Unterschied zu `\LTXTalkNavigations` Renderern für die *gesamte* Abschnittsfolge. Die getaggte Rolle kommt,

wie bei jedem anderen Slotinhalt, vom vierten Argument des umgebenden `\defineltxttalkslot` (Abschnitt 8.16 behandelt entsprechend die Tagging-Optionen der Slots).

Mitgeliefert:

- `pagenumber`, `totalframes` – aktuelle Foliennummer beziehungsweise Gesamtzahl.
- `pagenumber-of-total` – „ $\langle \text{Foliennummer} \rangle / \langle \text{Gesamtzahl} \rangle$ “.
- `section-pagenumber` – „ $\langle \text{Abschnittstitel} \rangle - \langle \text{Foliennummer} \rangle$ “.
- `sectionheading`, `subsectionheading` – nummerierter Abschnitts- beziehungsweise Unterabschnittstitel.

Jede eingebaute Instanz verwendet eine von drei Schablonen und lässt sich mit `\EditInstance{ $\text{ltxtalk-element}$ }{Name}{Schlüssel}` anpassen:

- `counter`: Schlüssel `source` (der auszugebende Wert).
- `fraction`: Schlüssel `left`, `right`, `separator`.
- `heading`: Schlüssel `numbered` (boolesch), `level`, `number`, `text`, `separator`. Mit `level` (`section`, `subsection` oder `subsubsection`) entfällt die Nummer, solange die aktuelle Überschrift dieser Ebene unnummeriert ist (`\section*`).

Neue Instanzen – auch mit einer dieser drei Schablonen, nur mit anderen Werten – entstehen wie bei jedem anderen Template-Typ über `\DeclareInstance{ $\text{ltxtalk-element}$ }{Name}{Schablone}{Schlüssel}`.

```

1 \defineltxttalkslot{my-page}{footer}{custom}{role=navigation}
2 \setltxttalkslotcontent{my-page}{\InsertLTXTalkElement{pagenumber-of-total}}
3 \EditInstance{ltxtalk-element}{pagenumber-of-total}{separator = { of }}
```

8.7. Farbkonfiguration

`\setltxttalkcolors{ $\langle \text{Farben} \rangle$ }`

Definiert die semantischen Farben `primary`, `secondary`, `accent`, `text`, `background`, `structure`, `alert` und `example`. Jeder Wert hat die Form `model=...`, `value=...`, beispielsweise `primary={model=HTML, value=005B50}`. Die zugehörigen Farbnamen lauten `ltxtalk-primary`, `ltxtalk-secondary`, `ltxtalk-accent`, `ltxtalk-text`, `ltxtalk-bg`, `ltxtalk-structure`, `ltxtalk-alert` und `ltxtalk-example`. Die letzten drei werden zugleich auf die Klassennamen `structure`, `alert` und `example` abgebildet.

8.8. Schriftkonfiguration

`\DeclareLTXTalkFontPalette{ $\langle \text{Name} \rangle$ }{ $\langle \text{Schriften} \rangle$ }`

\UseLTXTalkFontPalette{<Name>}

Deklariert beziehungsweise aktiviert eine Schriftpalette. Die portablen Slots heißen main, companion, display, accent, mono, math und symbol. Eine Ressource hat die Form font={...} mit optionalen features; alternativ verweist alias=... auf einen anderen Slot. main ist Pflicht. Die mitgelieferte Palette ltxtalk-default verwendet New Computer Modern Sans und Latin Modern Mono.

\UseLTXTalkFont{<Rolle>}

Wählt eine Textschrift über ihre semantische Rolle. Verfügbar sind body, support, heading, emphasis, code und symbol. Die technischen Palettennamen main, companion, display, accent und mono werden ebenfalls akzeptiert. math bezeichnet nur eine Ressource und kann nicht als Textschrift gewählt werden.

Die Rollen sind feste Namen für die Paletteneinträge:

Rolle	Paletteneintrag
body	main
support	companion
heading	display
emphasis	accent
code	mono
symbol	symbol

```

1 \DeclareLTXTalkFontPalette{my-fonts}{
2   main = {font={Latin Modern Sans}},
3   companion = {alias=main},
4   display = {alias=companion},
5   accent = {alias=display},
6   mono = {font={Latin Modern Mono}},
7   symbol = {alias=main}
8 }
9 \UseLTXTalkFontPalette{my-fonts}

```

Schriftnamen müssen in der TeX-Installation verfügbar sein. features enthält Schriftoptionen für [fontspec](#). Vollständige Angaben zu fehlenden Einträgen, Aliasauflösung und Mathematiksschriften stehen in der Schriftpalettenreferenz von [osgstyler](#); die beiden Palettenbefehle verwenden diese Schnittstelle. Mit der Ladeoption outer-only=true ist die Palettenauswahl deaktiviert.

8.9. Inneres Theme

\styleltxtalkblock{<Typ>}{<Optionen>}

Bearbeitet die Instanz block, alertblock oder exampleblock des Template-Typs osgstyler-block (bereitgestellt von [osgstyler](#)). Die Standard-Schnittstelle umfasst background, rule-color,

rule-thickness, padding, corner-radius, shadow, shadow-offset, before und after. Fertige Optiken liegen als `osgstyler`-Traits vor: `card` (leicht gerahmte, zart getönte Box mit fettem Titel), `titlebar` (Beamers Madrid: voller Farbbalken über getönter Body-Fläche mit Unterkante), `heading` (nur farbiger Titel, fließend), `soft` (Titel mit dünnen Linien, fließend), `plain` (rahmenlos) sowie `framed/filled` (Rahmen ohne bzw. mit Tönung). `heading`, `soft` und `plain` lassen das Tagging der Klasse unangetastet; die übrigen sind echte Kästen und fügen im getaggten PDF genau ein Div je Block hinzu. Ohne Aufruf verwendet `lttheme` den Trait `card`, sodass ein Block stets erkennbar ist; `magpie` und `bluebird` setzen `titlebar`, `hawk`, `sparrow` und `goose` `heading` – jeweils in der Themefarbe. Die Ladeoption `block-hairlines` (oder `\ltxtalksetup`) schaltet bei allen Themes dünne Trennlinien dazu (Rahmen beim Kasten, Kopf-/Fußlinie beim reinen Titel). Ein Blockstil wird beim Theme-Wechsel auf den `card`-Standard zurückgesetzt.

`\setltxtalkitemize{<Optionen>}`

Setzt mit level-1 bis level-4 das Aufzählungszeichen der jeweiligen `itemize`-Ebene. Es wird nur die visuelle Marke geändert; Listen- und Listenelement-Tags der Klasse bleiben unverändert.

8.10. Slot-Definition

`\defineltxtalkslot{<Name>}{<Bereich>}{<Typ>}{<Optionen für Barrierefreiheit>}`

Definiert einen Slot und fügt ihn am Ende des angegebenen Bereichs ein. Eine erneute Definition ersetzt seine Zuordnung und ordnet ihn dort erneut am Ende ein; sie erzeugt keinen zweiten Slot gleichen Namens. Gültige Standardbereiche sind `header`, `left`, `content`, `right` und `footer`. Vordefinierte Typen sind `title`, `subtitle`, `author`, `date`, `institute`, `section`, `subsection`, `frametitle`, `framesubtitle`, `pagenumber`, `logo` und `custom`.

`\setltxtalkslotcontent{<Name>}{<Inhalt>}`

Hinterlegt den auszuführenden Inhalt eines freien Slots. Er wird für `custom`, `logo` und nicht vordefinierte eigene Typen verwendet.

Typ	Beschreibung	Mögliche Rolle
<code>frametitle</code>	Folientitel	heading (H2)
<code>framesubtitle</code>	Folienuntertitel	heading (H3)
<code>title</code>	Präsentationstitel	heading (H1)
<code>section</code>	Aktueller Abschnitt	heading (H2)
<code>subsection</code>	Aktueller Unterabschnitt	heading (H3)
<code>author</code>	Autor	note
<code>date</code>	Datum	note
<code>institute</code>	Institut	note
<code>pagenumber</code>	Foliennummer	navigation
<code>logo</code>	Logo	artifact
<code>custom</code>	Frei definiert	artifact

TABELLE 1: Vordefinierte Slottypen

Die Rollen in der Tabelle sind Beispiele, keine Vorgaben des Slottyps. Ohne explizite Rolle gilt `artifact`; eine Ausnahme ist `frametitle`, dessen Tagging die Klasse übernimmt.

8.11. Slot-Positionierung

`\positionltxtalkslot{⟨Name⟩}{⟨x⟩}{⟨y⟩}{⟨Breite⟩}{⟨Höhe⟩}`

Positioniert den Slot relativ zu seinem Bereich. Für `⟨x⟩` sind `left`, `center`, `right` oder eine Länge zulässig; für `⟨y⟩` entsprechend `top`, `center`, `bottom` oder eine Länge. Positive Längen werden von der linken beziehungsweise oberen Kante gemessen.

`\DeclareLTXTalkAreaGrid{⟨Bereich⟩}{⟨Spalten⟩}{⟨Zeilen⟩}`

`\PositionLTXTalkSlotInGrid{⟨Name⟩}{⟨Spalte⟩}{⟨Zeile⟩}{⟨Spaltenanzahl⟩}{⟨Zeilenanzahl⟩}`

`\SetLTXTalkAreaGridRows{⟨Bereich⟩}{⟨Zeilen⟩}`

Definiert ein Raster für einen Bereich und positioniert Slots darin. Spalten sind kommasetrennte Längen; `*` verteilt den verbleibenden Platz gleichmäßig auf alle Sternspalten. Spalten und Zeilen werden ab eins gezählt. Die letzten beiden Argumente der Positionierung sind die Anzahl überspannter Spalten und Zeilen, keine Längen. Die Rasterdefinition und Änderungen der Zeilenzahl sind global; eine TeX-Gruppe oder das Frame-Ende nimmt sie nicht zurück. Stellen Sie für die nächste Folie die gewünschte Zeilenzahl explizit ein.

8.12. Titelseiten

`\DeclareLTXTalkTitleLayout{⟨Name⟩}{⟨Layoutoptionen⟩}{⟨Inhalt⟩}`

`\UseLTXTalkTitleLayout{⟨Name⟩}`

`\MakeLTXTalkTitle[⟨Name⟩]`

`\DeclareLTXTalkTitleLayout` registriert eine Titelseitenvariante aus temporären Layoutoptionen und ihrem Inhalt. `\UseLTXTalkTitleLayout` wählt die Vorgabe; `\MakeLTXTalkTitle` erzeugt den Frame und kann optional eine Variante nur für diesen Aufruf wählen. Nach der Titelseite wird die vorherige Geometrie wiederhergestellt. Das Theme `bluebird` bringt mit `bluebird/band` eine solche Variante mit; `ltx-talks \maketitle` bleibt davon unberührt.

`\LTXTalkHeaderRows`

`\IfLTXTalkTitlePageTF{⟨Titelseitencode⟩}{⟨Framecode⟩}`

Liefert Theme-Autoren die aktuell wirksame Zahl der Kopfzeilen bzw. unterscheidet eine von `\MakeLTXTalkTitle` erzeugte Titelseite von einem gewöhnlichen Frame. Beide Schnittstellen bleiben auch dann gültig, wenn die Basisklasse den Seitenkopf erst beim Ausschiffen aufbaut.

8.13. Slot-Gestaltung

`\styleltxtalkslot{⟨Name⟩}{⟨Optionen⟩}`

Ändert die angegebenen Eigenschaften der Slot-Instanz. Nicht genannte Eigenschaften behalten ihren bisherigen Wert. Die Vorgaben unten gelten für eine neue Instanz; Themes setzen eigene Werte.

`font-slot = {⟨Rolle⟩}` Voreinstellung: companion
Schriftrolle: body, support, heading, emphasis, code, symbol oder ein entsprechender Palettenname. math ist keine Textschrift.

`font = {⟨Befehle⟩}` Voreinstellung: leer
Zusätzliche Schriftbefehle, etwa `\Large\bfseries`; werden nach der Schriftrolle angewandt.

`color = {⟨Farbe⟩}` Voreinstellung: ltxtalk-text
Textfarbe.

`background = {⟨Farbe⟩}` Voreinstellung: none
Hintergrundfarbe oder none. Bei `frame=box` ergibt none einen weißen Hintergrund.

`background-fill = true|false` Voreinstellung: false
Füllt bei gesetztem Hintergrund die deklarierte Slotfläche statt nur die Textfläche. Bei einem Boxrahmen gilt dessen Darstellung.

`align = left|center|right` Voreinstellung: center
Horizontale Textausrichtung.

`frame = none|box|top|bottom|left|right` Voreinstellung: none
box zeichnet einen Rahmen. Die einzelnen Kantenwerte werden angenommen, zeichnen im aktuellen Slotrenderer aber keine Linie.

`padding = {⟨Länge⟩}` Voreinstellung: 0pt
Innenabstand bei Farbflächen und Boxrahmen.

`rule-color = {⟨Farbe⟩}` Voreinstellung: ltxtalk-primary
Rahmenfarbe.

`rule-thick = {⟨Länge⟩}` Voreinstellung: 0.4pt
Linienstärke des Boxrahmens. Anders als bei Bereichen heißt dieser Schlüssel nicht `rule-thickness`.

`space-after = {⟨Länge⟩}` Voreinstellung: 0pt
Abstand unter einem Titel-Slot im Inhaltsbereich. Positive Werte setzen den Abstand; sonst gilt `0.4\baselineskip` in der Slotschrift. Ohne Wirkung auf absolut positionierte Slots.

Zusätzlich angenommen, aber derzeit ohne sichtbare Wirkung: `rotate` (Vorgabe 0), `opacity` (1), `shadow`, `uppercase` und `lowercase` (jeweils false). Verwenden Sie diese Schlüssel nicht als Voraussetzung für Ihr Layout.

8.14. Bereichsdekoration

`\decorateltxtalkarea{⟨Bereich⟩}{⟨Optionen⟩}`

Ändert die angegebenen Eigenschaften der Bereichsinstanz. Die übrigen bleiben erhalten.

`background = {⟨Farbe⟩}` Voreinstellung: none

Einfarbiger Hintergrund oder keine Füllung.

`background-split = {⟨Farbliste⟩}` Voreinstellung: leer

Nebeneinanderliegende Farbflächen; Farbe*2 gibt doppeltes Gewicht. Verwenden Sie gültige Farbnamen wie `ltxtalk-primary`.

`background-gradient = {⟨Farben⟩}` Voreinstellung: leer

Zwei Farben {oben, unten} für einen vertikalen Verlauf.

`frame = none|box|top|bottom|left|right|topbottom` Voreinstellung: none

top, bottom und topbottom zeichnen horizontale Linien. left und right zeichnen die jeweilige Seitenkante. box zeichnet derzeit nur die obere und untere Linie.

`rule-color = {⟨Farbe⟩}` Voreinstellung: `ltxtalk-primary`

Farbe der Bereichslinien.

`rule-thickness = {⟨Länge⟩}` Voreinstellung: 0.4pt

Stärke der Bereichslinien.

`padding = {⟨Länge⟩}` Voreinstellung: 0pt

Abstand nach einer linken Bereichslinie; verschiebt keine Slotpositionen.

`artifact-type = {⟨Name⟩}` Voreinstellung: layout

Beschreibende Angabe; die Standardausgabe wertet den Wert derzeit nicht aus.

Die Schlüssel `background-opacity` (Vorgabe 1), `shadow` (false), `shadow-color` (black!30), `shadow-offset` (3pt), `margin-top` und `margin-bottom` (jeweils 0pt) werden angenommen, sind aber derzeit ohne sichtbare Wirkung. Verwenden Sie für die Geometrie `\ltxtalksetup`.

8.15. Titelplatzierung

`\setltxtalktitleplacement{⟨Optionen⟩}`

Legt mit `frametitle=content|header|both` und `section=header|none` die beabsichtigte Titelplatzierung fest. Das aktive Theme muss Slots für die gewählten Bereiche definieren.

`\setltxtalkheadingbehavior{⟨Optionen⟩}`

Konfiguriert mit dem booleschen Schlüssel `auto-section-title` (Vorgabe: false), ob die erste `\frametitle`-Folie nach `\section`/`\subsection` deren nummerierte Überschrift als Titel übernimmt. Ein an derselben Folie gegebener Text wird dabei zum Subtitel; ein zusätzlich gesetzter `\framesubtitle` entfällt.

Bekanntes Problem: Mit `tagging=on` kann diese Option bei bestimmten Folieninhalten zum Fehler „TeX capacity exceeded“ führen. Prüfen Sie das vollständige Dokument und deaktivieren Sie `auto-section-title`, falls dieser Fehler auftritt.

`\contframetitle[⟨Untertitel⟩]`

Wiederholt die zuletzt mit `\section` oder `\subsection` gesetzte, nummerierte Überschrift mit einem Fortsetzungshinweis. Verwenden Sie den Befehl innerhalb eines neuen frame, wenn ein Thema mehrere separate Frames umfasst. Er ist unabhängig von `auto-section-title` und übernimmt nicht den letzten manuell gesetzten Folientitel. Das optionale Argument setzt einen Untertitel; ein Pflichtargument gibt es nicht. Vor dem optionalen Argument ist wie bei `\frametitle` eine Overlay-Spezifikation in spitzen Klammern möglich. Für Overlays innerhalb desselben Frames genügt `\frametitle`.

8.16. Optionen für Barrierefreiheit

Das vierte Argument von `\defineltxtalkslot` legt die semantische Rolle und gegebenenfalls einen Alternativtext fest. Diese Angaben wirken bei aktiviertem Tagging. Leere Slots erzeugen kein Strukturelement. Beim Typ `frametitle` übernimmt die Klasse das Tagging; die folgenden Optionen ändern dessen Struktur nicht.

`role = {⟨Rolle⟩}` Voreinstellung: artifact
`heading` kennzeichnet eine Überschrift auf der mit `heading-level` gewählten Ebene; `heading-1`, `heading-2` und `heading-3` wählen direkt die jeweilige Ebene. `note` kennzeichnet eine Anmerkung. `artifact` und `navigation` nehmen dekorative beziehungsweise wiederkehrende Navigations-elemente aus der Lesereihenfolge aus. `content` verwendet das automatische Absatz-Tagging für normalen Inhalt.

`heading-level = {⟨Ganzzahl⟩}` Voreinstellung: 1
 Überschriftenebene für `role=heading`, begrenzt auf 1 bis 6.

`alt-text = {⟨Text⟩}` Voreinstellung: leer
 Alternativtext für den Slotinhalt. Ein nicht leerer Wert erhält bei Überschriften und Anmerkungen deren Rolle; für alle anderen Rollen wird der Slot als Abbildung (Figure) mit Alternativtext ausgegeben. Er wird dadurch auch bei `artifact` oder `navigation` Teil der Dokumentstruktur. Für rein dekorative Elemente lassen Sie den Wert leer.

`continuation-text = {⟨Text⟩}`
 Beabsichtigter Fortsetzungstext; derzeit ohne Ausgabewirkung.

`continuation-mode = explicit|implicit`
 Beabsichtigte Fortsetzungsart; derzeit ohne Ausgabewirkung.

Bereichshintergründe mit Farbaufteilung oder Verlauf werden unabhängig davon immer ausdrücklich als Artefakte ausgegeben.

8.17. Templates und Sockets

Für jeden Slot existiert eine Instanz des Template-Typs `lttheme-slot`, für jeden Bereich eine Instanz von `lttheme-area`. Zum Bearbeiten dieser Instanzen dienen die beiden zuvor dokumentierten Befehle für Slotgestaltung und Bereichsdekoration.

8. Referenz

Der Seitenstil stellt Sockets für den Seitenhintergrund, die Bereiche `header`, `left`, `right` und `footer` sowie den Seitenvordergrund bereit. Ihre Namen folgen dem Schema `lttheme/area/⟨Bereich⟩` beziehungsweise `lttheme/page/⟨Ebene⟩`. Der Plug `standard` rendert den zugehörigen Bereich. Eigene Plugs sind für grundlegende, austauschbare Ausgabephasen gedacht; zusätzliche Seitenelemente sollten gewöhnlich als Slots eines Teilthemes implementiert werden.

Index

A

<code>\addltxtalktheme</code>	17, 20
<code>align</code>	28
<code>alt-text</code>	30
<code>artifact-type</code>	29

B

<code>background</code>	28 f.
<code>background-fill</code>	28
<code>background-gradient</code>	29
<code>background-split</code>	29
<code>block-hairlines</code>	20, 22, 26

C

<code>color</code>	28
<code>content-inset-left</code>	22
<code>content-inset-right</code>	22
<code>\contframetitle</code>	30
<code>continuation-mode</code>	30
<code>continuation-text</code>	30

D

<code>\DeclareInstance</code>	24
<code>\DeclareLTXTalkAreaGrid</code>	27
<code>\DeclareLTXTalkFontPalette</code>	24
<code>\DeclareLTXTalkLayout</code>	20
<code>\DeclareLTXTalkNavigation</code>	23
<code>\DeclareLTXTalkTitleLayout</code>	27
<code>\decorateltxtalkarea</code>	29
<code>\defineltxttalkslot</code>	26

E

<code>\EditInstance</code>	24
----------------------------------	----

F

<code>font</code>	28
<code>font-slot</code>	28
<code>footer</code>	21
<code>footer-columns</code>	22
<code>footer-height</code>	22
<code>frame (Umgebung)</code>	30
<code>frame</code>	28 f.
<code>\framesubtitle</code>	11, 29
<code>\frametitle</code>	11, 29

H

<code>header</code>	21
<code>header-columns</code>	22
<code>header-height</code>	22
<code>header-rows</code>	22
<code>heading-level</code>	30

I

<code>\IfLTXTalkTitlePageTF</code>	27
<code>\InsertLTXTalkElement</code>	23

L

<code>left-area-width</code>	22
<code>left-margin</code>	21
<code>linebreak</code>	23
<code>lines</code>	23
<code>ltx-talk</code> (Klasse)	2, 4, 12 f., 18, 27
<code>ltx-talk</code> (Paket)	14
<code>ltxtalk-theme</code> (Paket)	15
<code>\LTXTalkBaseBottomMargin</code>	21
<code>\LTXTalkBaseLeftMargin</code>	21
<code>\LTXTalkBaseRightMargin</code>	21
<code>\LTXTalkBaseTopMargin</code>	21
<code>\LTXTalkCurrentBottomMargin</code>	21
<code>\LTXTalkCurrentLeftMargin</code>	21
<code>\LTXTalkCurrentRightMargin</code>	21
<code>\LTXTalkCurrentTopMargin</code>	21
<code>\LTXTalkHeaderRows</code>	27
<code>\LTXTalkMetadata</code>	22
<code>\LTXTalkNavigation</code>	23
<code>\ltxtalksetup</code>	20

M

<code>\MakeLTXTalkTitle</code>	27
<code>margin-width</code>	22

O

<code>outer-only</code>	20
-------------------------------	----

P

<code>padding</code>	28 f.
<code>\positionltxttalkslot</code>	27
<code>\PositionLTXTalkSlotInGrid</code>	27

R

<code>\RegisterLTXTalkTheme</code>	20
<code>right-area-width</code>	22
<code>right-margin</code>	21
<code>role</code>	30
<code>rule-color</code>	28 f.
<code>rule-thick</code>	28
<code>rule-thickness</code>	29

S

<code>\section</code>	11, 29
<code>\SetLTXTalkAreaGridRows</code>	27
<code>\setltxtalkcolors</code>	24
<code>\setltxtalkheadingbehavior</code>	29

INDEX

<code>\setltxtalkitemize</code>	26	U	
<code>\setltxtalkslotcontent</code>	14, 26	<code>\UseLXTalkFont</code>	25
<code>\setltxtalktitleplacement</code>	29	<code>\UseLXTalkFontPalette</code>	25
<code>shadow</code>	20	<code>\UseLXTalkLayout</code>	21
<code>space-after</code>	28	<code>\UseLXTalkMetadata</code>	22
<code>\styleltxtalkblock</code>	25	<code>\uselxtalktheme</code>	20
<code>\styleltxtalkslot</code>	28	<code>\UseLXTalkTitleLayout</code>	27
<code>\subsection</code>	11, 29		