

KOMA-Script für Autoren

In diesem Teil sind die Informationen für die Autoren von Artikeln, Berichten, Büchern und Briefen zu finden. Dabei wird davon ausgegangen, dass der normale Anwender sich weniger dafür interessiert, wie in KOMA-Script die Dinge implementiert sind und wo die Schwierigkeiten dabei liegen. Auch ist es für den normalen Anwender wenig interessant, welche obsoleten Optionen und Anweisungen noch enthalten sind. Er will wissen, wie er aktuell etwas erreichen kann. Eventuell ist er noch an typografischen Hintergrundinformationen interessiert.

Die wenigen Passagen, die weiterführende Informationen und Begründungen enthalten und deshalb für ungeduldige Leser weniger von Interesse sind, wurden in diesem Teil in serifenloser Schrift gesetzt und können bei Bedarf übersprungen werden. Wer hingegen noch mehr Informationen zu Hintergründen der Implementierung, Nebenwirkungen bei Verwendung anderer Pakete und zu obsoleten Optionen oder Anweisungen sucht, sei auf Teil II ab Seite 379 verwiesen. Darüber hinaus beschäftigt sich jener Teil von KOMA-Script auch mit all den Möglichkeiten, die speziell für Autoren von Paketen und Klassen geschaffen wurden.

2.

Satzspiegelberechnung mit `typearea.sty`

Viele \LaTeX -Klassen bieten dem Anwender eine weitgehend feste Aufteilung von Rändern und Textbereich. Bei den Standardklassen ist die konkrete Aufteilung in engen Grenzen von der gewählten Schriftgröße abhängig. Darüber hinaus gibt es Pakete wie `geometry` (siehe [Ume10]), die dem Anwender die volle Kontrolle, aber auch die Verantwortung für die Einstellungen des Textbereichs und der Ränder überlassen.

KOMA-Script geht mit dem Paket `typearea` einen etwas anderen Weg. Hier werden basierend auf einer in der Typografie etablierten Konstruktion Einstellmöglichkeiten und Automatismen geboten, die es dem Anwender erleichtern, eine gute Wahl zu treffen.

2.1. Grundlagen der Satzspiegelkonstruktion

Betrachtet man eine einzelne Seite eines Buches oder eines anderen Druckwerkes, so besteht diese auf den ersten Blick aus den Rändern, einem Kopfbereich, einem Textkörper und einem Fußbereich. Genauer betrachtet, kommt noch ein Abstand zwischen Kopfbereich und Textkörper sowie zwischen Textkörper und Fußbereich hinzu. Der Textkörper heißt in der Fachsprache der Typografen und Setzer *Satzspiegel*. Die Aufteilung dieser Bereiche sowie ihre Anordnung zueinander und auf dem Papier nennen wir *Satzspiegeldefinition* oder *Satzspiegelkonstruktion*.

In der Literatur werden verschiedene Algorithmen und heuristische Verfahren zur Konstruktion eines guten Satzspiegels vorgeschlagen und diskutiert. Einen Überblick über einige Möglichkeiten und der dabei angenommenen Grundsätze gibt Anhang A. Häufig findet man ein Verfahren, das mit verschiedenen Diagonalen und deren Schnittpunkten arbeitet. Das gewünschte Ergebnis ist, dass das Seitenverhältnis des Satzspiegels dem Seitenverhältnis *der Seite* entspricht. Bei einem einseitigen Dokument sollen außerdem der linke und der rechte Rand gleich breit sein, während der obere zum unteren Rand

im Verhältnis 1:2 stehen sollte. Bei einem doppelseitigen Dokument, beispielsweise einem Buch, ist hingegen zu beachten, dass der gesamte innere Rand genauso groß sein sollte wie jeder der beiden äußeren Ränder. Eine einzelne Seite steuert dabei jeweils nur die Hälfte des inneren Randes bei.

Im vorherigen Absatz wurde *die Seite* erwähnt und hervorgehoben. Irrtümlich wird oftmals angenommen, das Format der Seite wäre mit dem Format des Papiers gleichzusetzen. Betrachtet man jedoch ein gebundenes Druckerzeugnis, so ist zu erkennen, dass ein Teil des Papiers in der Bindung verschwindet und nicht mehr als Seite zu sehen ist. Für den Satzspiegel ist jedoch nicht entscheidend, welches Format das Papier hat, sondern, was der Leser für einen Eindruck vom Format der Seite bekommt. Damit ist klar, dass bei der Berechnung des Satzspiegels der Teil, der durch die Bindung versteckt wird, aus dem Papierformat herausgerechnet und dann zum inneren Rand hinzugefügt werden muss. Wir nennen diesen Teil *Bindekorrektur*. Die Bindekorrektur ist also rechnerischer Bestandteil des *Bundstegs*, nicht jedoch des sichtbaren inneren Randes.

Die Bindekorrektur ist vom jeweiligen Produktionsvorgang abhängig und kann nicht allgemein festgelegt werden. Es handelt sich dabei also um einen Parameter, der für jeden Produktionsvorgang neu festzulegen ist. Im professionellen Bereich spielt dieser Wert nur eine geringe Rolle, da ohnehin auf größere Papierbögen gedruckt und entsprechend geschnitten wird. Beim Schneiden wird dann wiederum sichergestellt, dass obige Verhältnisse für die sichtbare Doppelseite eingehalten sind.

Wir wissen nun also, wie die einzelnen Teile zueinander stehen. Wir wissen aber noch nicht, wie breit und hoch der Satzspiegel ist. Kennen wir eines dieser beiden Maße, so ergeben sich zusammen mit dem Papierformat und dem Seitenformat oder der Bindekorrektur alle anderen Maße durch Lösung mehrerer mathematischer Gleichungen:

$$\text{Satzspiegelhöhe} : \text{Satzspiegelbreite} = \text{Seitenhöhe} : \text{Seitenbreite}$$

$$\text{oberer Rand} : \text{unterer Rand} = 1 : 2$$

$$\text{linker Rand} : \text{rechter Rand} = 1 : 1$$

$$\text{innerer Randanteil} : \text{äußerer Rand} = 1 : 2$$

$$\text{Seitenbreite} = \text{Papierbreite} - \text{Bindekorrektur}$$

$$\text{oberer Rand} + \text{unterer Rand} = \text{Seitenhöhe} - \text{Satzspiegelhöhe}$$

$$\text{linker Rand} + \text{rechter Rand} = \text{Seitenbreite} - \text{Satzspiegelbreite}$$

$$\text{innerer Randanteil} + \text{äußerer Rand} = \text{Seitenbreite} - \text{Satzspiegelbreite}$$

$$\text{innerer Randanteil} + \text{Bindekorrektur} = \text{Bundsteg}$$

Dabei gibt es *linker Rand* und *rechter Rand* nur im einseitigen Druck. Entsprechend gibt es *innerer Randanteil* und *äußerer Rand* nur im doppelseitigen Druck.

In den Gleichungen wird mit *innerer Randanteil* gearbeitet, weil der komplette innere Rand ein Element der vollständigen Doppelseite ist. Zu einer Seite gehört also nur die Hälfte des inneren Randes: *innerer Randanteil*.

Die Frage nach der Breite des Satzspiegels wird in der Literatur ebenfalls diskutiert. Die optimale Satzspiegelbreite ist von verschiedenen Faktoren abhängig:

- Größe, Laufweite und Art der verwendeten Schrift,
- verwendeter Durchschuss,
- Länge der Worte,
- verfügbarer Platz.

Der Einfluss der Schrift wird deutlich, wenn man sich bewusst macht, wozu Serifen dienen. Serifen sind kleine Striche an den Linienenden der Buchstaben. Buchstaben, die mit vertikalen Linien auf die Grundlinie der Textzeile treffen, lösen diese eher auf, als dass sie das Auge auf der Linie halten. Genau bei diesen Buchstaben liegen die Serifen horizontal auf der Grundlinie und verstärken damit die Zeilenwirkung der Schrift. Das Auge kann der Textzeile nicht nur beim Lesen der Worte, sondern insbesondere auch beim schnellen Zurückspringen an den Anfang der nächsten Zeile besser folgen. Damit darf die Zeile bei einer Schrift mit Serifen genau genommen länger sein als bei einer Schrift ohne Serifen.

Unter dem Durchschuss versteht man den Abstand zwischen Textzeilen. Bei \LaTeX ist ein Durchschuss von etwa 20 % der Schriftgröße voreingestellt. Mit Befehlen wie `\linespread` oder besser mit Hilfe von Paketen wie `setspace` (siehe [TF11]) kann der Durchschuss verändert werden. Ein großer Durchschuss erleichtert dem Auge die Verfolgung einer Zeile. Bei sehr großem Durchschuss wird das Lesen aber dadurch gestört, dass das Auge zwischen den Zeilen weite Wege zurücklegen muss. Daneben wird sich der Leser des entstehenden Streifeneffekts sehr deutlich und unangenehm bewusst. Der Graueindruck der Seite ist in diesem Fall gestört. Dennoch dürfen bei großem Durchschuss die Zeilen länger sein.

Durchschuss

Auf der Suche nach konkreten Werten für gute Zeilenlängen findet man in der Literatur je nach Autor unterschiedliche Angaben. Teilweise ist dies auch in der Muttersprache des Autors begründet. Das Auge springt nämlich üblicherweise von Wort zu Wort, wobei kurze Wörter diese Aufgabe erleichtern. Über alle Sprachen und Schriftarten

hinweg kann man sagen, dass eine Zeilenlänge von 60 bis 70 Zeichen, einschließlich Leer- und Satzzeichen, einen brauchbaren Kompromiss darstellt. Ein gut gewählter Durchschuss wird dabei vorausgesetzt. Bei den Voreinstellungen von \LaTeX braucht man sich über Letzteres normalerweise keine Sorgen zu machen. Größere Zeilenlängen darf man nur Gewohnheitslesern zumuten, die täglich viele Stunden lesend zubringen. Aber auch dann sind Zeilenlängen jenseits von 80 Zeichen unzumutbar. In jedem Fall ist dann der Durchschuss anzupassen. 5 % bis 10 % zusätzlich sind dabei als Faustregel empfehlenswert. Bei Schriften wie Palatino, die bereits bei einer normalen Zeilenlänge nach 5 % mehr Durchschuss verlangt, können es auch mehr sein.

Bevor wir uns an die konkrete Konstruktion machen, fehlen jetzt nur noch Kleinigkeiten, die man wissen sollte. \LaTeX beginnt die erste Zeile des Textbereichs einer Seite nicht am oberen Rand des Textbereichs, sondern setzt die Grundlinie der Zeile mit einem definierten Mindestabstand zum oberen Rand des Textbereichs. Des Weiteren verfügt \LaTeX über die beiden Befehle `\raggedbottom` und `\flushbottom`. Der erste dieser Befehle legt fest, dass die letzte Zeile einer jeden Seite dort liegen soll, wo sie eben zu liegen kommt. Das kann dazu führen, dass sich die Position der letzten Zeile von Seite zu Seite vertikal um nahezu eine Zeile verändern kann – bei Zusammenreffen des Seitenendes mit Überschriften, Abbildungen, Tabellen oder Ähnlichem auch mehr. Im doppelseitigen Druck ist das in der Regel unerwünscht. Mit dem zweiten Befehl, `\flushbottom`, wird hingegen festgelegt, dass die letzte Zeile immer am unteren Rand des Textbereichs zu liegen kommt. Um diesen vertikalen Ausgleich zu erreichen, muss \LaTeX gegebenenfalls dehnbare vertikale Abstände über das erlaubte Maß hinaus strecken. Ein solcher Abstand ist beispielsweise der Absatzabstand. Dies gilt in der Regel auch, wenn man gar keinen Absatzabstand verwendet. Um nicht bereits auf normalen Seiten, auf denen der Absatzabstand das einzige dehnbare vertikale Maß darstellt, eine Dehnung zu erzwingen, sollte die Höhe des Textbereichs ein Vielfaches der Textzeilenhöhe zuzüglich des Abstandes der ersten Zeile vom oberen Rand des Textbereichs sein.

Soweit die Grundlagen. In den folgenden beiden Abschnitten werden die von KOMA-Script angebotenen Konstruktionen im Detail vorgestellt.

2.2. Satzspiegelkonstruktion durch Teilung

Der einfachste Weg, um zu erreichen, dass der Textbereich dasselbe Seitenverhältnis aufweist wie die Seite, ist folgender:

Abbildung 2.1.: Doppelseite mit der Rasterkonstruktion für die klassische Neunerteilung nach Abzug einer Bindekorrektur

1	2	3	4	5	6	7	8	9		9	8	7	6	5	4	3	2	1
2																		2
3																		3
4																		4
5																		5
6																		6
7																		7
8																		8
9																		9

- Zunächst zieht man an der Innenseite des Papiers den Teil *BCOR*, der für die Bindekorrektur benötigt wird, ab und teilt die restliche Seite vertikal in eine Anzahl *DIV* gleich hoher Streifen.
- Dann teilt man die Seite horizontal in die gleiche Anzahl *DIV* gleich breiter Streifen.
- Nun verwendet man den obersten horizontalen Streifen als oberen und die beiden untersten horizontalen Streifen als unteren Rand. Im doppelseitigen Druck verwendet man außerdem den innersten vertikalen Streifen als inneren und die beiden äußersten vertikalen Streifen als äußeren Rand.
- Zum inneren Rand gibt man dann noch *BCOR* hinzu.

Was nun innerhalb der Seite noch übrig bleibt, ist der Textbereich. Die Breite bzw. Höhe der Ränder und des Textbereichs resultiert damit automatisch aus der Anzahl *DIV* der Streifen. Da für die Ränder insgesamt jeweils drei Streifen benötigt werden, muss *DIV* zwingend größer als drei sein. Damit der Satzspiegel horizontal und vertikal jeweils mindestens doppelt so viel Platz wie die Ränder einnimmt, sollte *DIV* sogar mindestens 9 betragen. Mit diesem Wert ist die Konstruktion auch als *klassische Neunerteilung* bekannt (siehe Abbildung 2.1).

Bei KOMA-Script ist diese Art der Konstruktion im Paket `typearea` realisiert, wobei der untere Rand weniger als eine Textzeile kleiner

ausfallen kann, um die im vorherigen Abschnitt erwähnte Nebenbedingung für die Satzspiegelhöhe einzuhalten und damit die dort erwähnte Problematik in Bezug auf `\flushbottom` zu mindern. Dabei sind für A4-Papier je nach Schriftgröße unterschiedliche Werte für *DIV* voreingestellt, die Tabelle 2.2, Seite 41 zu entnehmen sind. Bei Verzicht auf Bindekorrektur, wenn also $BCOR = 0\text{ pt}$ gilt, ergeben sich in etwa die Satzspiegelmaße aus Tabelle 2.1, Seite 40.

Neben den voreingestellten Werten kann man *BCOR* und *DIV* direkt beim Laden des Pakets als Option angeben (siehe Abschnitt 2.6 ab Seite 38). Zusätzlich existiert ein Befehl, mit dem man einen Satzspiegel explizit berechnen kann und dem man die beiden Werte als Parameter übergibt (siehe ebenfalls Abschnitt 2.6, Seite 45).

Das *typearea*-Paket bietet außerdem die Möglichkeit, den optimalen *DIV*-Wert automatisch zu bestimmen. Dieser ist von der Schrift und dem Durchschuss abhängig, der zum Zeitpunkt der Satzspiegelberechnung eingestellt ist. Siehe hierzu ebenfalls Abschnitt 2.6, Seite 41).

2.3. Satzspiegelkonstruktion durch Kreisschlagen

Neben der zuvor beschriebenen Satzspiegelkonstruktion gibt es in der Literatur noch eine eher klassische oder sogar mittelalterliche Methode. Bei diesem Verfahren will man die gleichen Werte nicht nur in Form des Seitenverhältnisses wiederfinden; man geht außerdem davon aus, dass das Optimum dann erreicht wird, wenn die Höhe des Textbereichs der Breite der Seite entspricht. Das genaue Verfahren ist beispielsweise in [Tsc87] nachzulesen.

Als Nachteil dieses spätmittelalterlichen Buchseitenkanons ergibt sich, dass die Breite des Textbereichs nicht mehr von der Schriftart abhängt. Es wird also nicht mehr der zur Schrift passende Textbereich gewählt, stattdessen muss der Autor oder Setzer unbedingt die zum Textbereich passende Schrift wählen. Dies ist als zwingend zu betrachten.

Im *typearea*-Paket wird diese Konstruktion dahingehend abgewandelt, dass durch Auswahl eines ausgezeichneten – normalerweise unsinnigen – *DIV*-Wertes oder einer speziellen, symbolischen Angabe derjenige *DIV*-Wert ermittelt wird, bei dem der resultierende Satzspiegel dem spätmittelalterlichen Buchseitenkanon am nächsten kommt. Es wird also wiederum auf die Satzspiegelkonstruktion durch Teilung zurückgegriffen.

2.4. Frühe oder späte Optionenwahl

In diesem Abschnitt wird eine Besonderheit von KOMA-Script vorgestellt, die neben `typearea` auch andere KOMA-Script-Pakete und -Klassen betrifft. Damit die Anwender alle Informationen zu einem Paket oder einer Klasse im jeweiligen Kapitel finden, ist dieser Abschnitt nahezu gleichlautend in mehreren Kapiteln zu finden. Anwender, die nicht nur an der Anleitung zu einem Paket oder einer Klasse interessiert sind, sondern sich einen Gesamtüberblick über KOMA-Script verschaffen wollen, brauchen diesen Abschnitt nur in einem der Kapitel zu lesen und können ihn beim weiteren Studium der Anleitung dann überspringen.

```
\documentclass[Optionenliste]{KOMA-Script-Klasse}
\usepackage[Optionenliste]{Paket-Liste}
```

Bei \LaTeX ist vorgesehen, dass Anwender Klassenoptionen in Form einer durch Komma getrennten Liste einfacher Schlüsselwörter als optionales Argument von `\documentclass` angeben. Außer an die Klasse werden diese Optionen auch an alle Pakete weitergereicht, die diese Optionen verstehen. Ebenso ist vorgesehen, dass Anwender Paketoptionen in Form einer durch Komma getrennten Liste einfacher Schlüsselwörter als optionales Argument von `\usepackage` angeben. KOMA-Script erweitert den Mechanismus der Optionen für die KOMA-Script-Klassen und einige Pakete um weitere Möglichkeiten. So haben die meisten Optionen bei KOMA-Script zusätzlich einen Wert. Eine Option hat also nicht unbedingt nur die Form *Option*, sondern kann auch die Form *Option=Wert* haben. Bis auf diesen Unterschied arbeiten `\documentclass` und `\usepackage` bei KOMA-Script wie in [Tea05b] oder jeder \LaTeX -Einführung, beispielsweise [DGS⁺12], beschrieben.

Klassenoptionen

globale Optionen

Paketoptionen

Option=Wert

Bei Verwendung einer KOMA-Script-Klasse sollten im Übrigen beim Laden des Pakets `typearea` oder `scrbase` keine Optionen angegeben werden. Das ist darin begründet, dass die Klasse diese Pakete bereits ohne Optionen lädt und \LaTeX das mehrmalige Laden eines Pakets mit unterschiedlicher Angabe von Optionen verweigert.

Achtung!

Das Setzen der Optionen mit `\documentclass` oder `\usepackage` hat übrigens einen entscheidenden Nachteil: Anweisungen, Längen, Zähler und ähnliches können darin leicht zerbrechen. So führt die Verwendung einer \LaTeX -Länge im Wert einer Option bei diesen Anweisungen zu einer Fehlermeldung, noch bevor der Wert an KOMA-Script übergeben wird, es also die Kontrolle darüber übernehmen könnte. Wertzuweisungen mit \LaTeX -Längen oder \LaTeX -Zählern sollten daher nie per `\documentclass` oder `\usepackage`,

Achtung!

sondern immer mit den nachfolgend dokumentierten Anweisungen `\KOMAOPTIONS` oder `\KOMAOPTION` vorgenommen werden.

```
\KOMAOPTIONS{Optionenliste}
\KOMAOPTION{Option}{Werteliste}
```

KOMA-Script bietet bei den meisten Klassen- und Paketoptionen auch die Möglichkeit, den Wert der Optionen noch nach dem Laden der Klasse beziehungsweise des Pakets zu ändern. Mit der Anweisung `\KOMAOPTIONS` kann man wie bei `\documentclass` oder `\usepackage` die Werte einer Reihe von Optionen ändern. Jede Option der *Optionenliste* hat dabei die Form *Option=Wert*.

Einige Optionen besitzen auch einen Säumniswert (engl. *default value*). Versäumt man die Angabe eines Wertes, verwendet man die Option also einfach in der Form *Option*, so wird automatisch dieser Säumniswert angenommen.

Manche Optionen können gleichzeitig mehrere Werte besitzen. Für solche Optionen besteht die Möglichkeit, mit `\KOMAOPTION` der einen *Option* nacheinander eine Reihe von Werten zuzuweisen. Die einzelnen Werte sind dabei in der *Werteliste* durch Komma voneinander getrennt.

Achtung! Soll ein *Wert* ein Gleichheitszeichen oder ein Komma enthalten, so ist der *Wert* in geschweifte Klammern zu setzen.

KOMA-Script bedient sich für die Realisierung dieser Möglichkeit der Anweisungen `\FamilyOptions` und `\FamilyOption` mit der Familie »KOMA«. Näheres zu diesen Anweisungen finden fortgeschrittene Anwender in Abschnitt 12.2 ab Seite 388.

Mit `\KOMAOPTIONS` oder `\KOMAOPTION` gesetzte Optionen erreichen sowohl die KOMA-Script-Klasse als auch alle bereits geladenen KOMA-Script-Pakete, die diese Optionen kennen. Ist eine Option oder ein Wert insgesamt unbekannt, so wird die Option einschließlich des Wertes von `scrbase` als fehlerhaft gemeldet.

2.5. Kompatibilität zu früheren Versionen von KOMA-Script

Wer seine Dokumente im Quellcode archiviert, legt in der Regel allergrößten Wert darauf, dass bei zukünftigen L^AT_EX-Läufen immer wieder exakt dasselbe Ergebnis erzielt wird. In einigen Fällen führen aber Verbesserungen und Korrekturen am Paket zu Änderungen im Verhalten, insbesondere beim Umbruch. Dies ist jedoch manchmal eher unerwünscht.

```
version=Wert
version=first
version=last
```

Seit Version 3.01b besteht bei `typearea` die Wahl, ob eine Quelldatei, soweit irgend möglich, auch zukünftig bei einem \LaTeX -Lauf zu exakt demselben Ergebnis führen soll oder ob er jeweils entsprechend der Anpassungen der neusten Version zu setzen ist. Zu welcher Version Kompatibilität herzustellen ist, wird dabei über die Option `version` festgelegt. Kompatibilität zur ältesten unterstützten KOMA-Script-Version kann mit `version=first` oder `version=2.9` oder `version=2.9t` erreicht werden. Bei Angabe einer unbekannten Version als *Wert* wird eine Warnung ausgegeben und sicherheitshalber `version=first` angenommen.

`version=first`

Mit `version=last` kann die jeweils neueste Version ausgewählt werden. In diesem Fall wird also auf rückwirkende Kompatibilität verzichtet. Wird die Option ohne Wertangabe verwendet, so wird ebenfalls `last` angenommen. Dies entspricht auch der Voreinstellung, solange keine obsolet Option verwendet wird.

`version=last`
Voreinstellung

Bei der Verwendung einer obsoleten Option von KOMA-Script 2 setzt KOMA-Script 3 automatisch `version=first`. In der dabei ausgegebenen Warnung wird erklärt, wie man diese Kompatibilitätsumschaltung verhindern kann. Alternativ kann man auch nach der obsoleten Option selbst eine abweichende Einstellung für Option `version` wählen.

Achtung!

Die Frage der Kompatibilität betrifft in erster Linie Fragen des Umbruchs. Neue Möglichkeiten, die sich nicht auf den Umbruch auswirken, sind auch dann verfügbar, wenn man per Option die Kompatibilität zu einer älteren Version ausgewählt hat. Die Option hat keine Auswirkungen auf Umbruchänderungen, die bei Verwendung einer neueren Version durch Beseitigung eindeutiger Fehler entstehen. Wer auch im Fehlerfall unbedingte Umbruchkompatibilität benötigt, sollte stattdessen mit dem Dokument auch die verwendete KOMA-Script-Version archivieren.

Tipp!

Es ist zu beachten, dass die Option `version` nach dem Laden des Pakets `typearea` nicht mehr verändert werden kann. Das Setzen mit `\KOMAOPTIONS` oder `\KOMAOPTION` ist daher nicht vorgesehen.

Achtung!

2.6. Einstellung des Satzspiegels und der Seitenaufteilung

Das Paket *typearea* bietet zwei unterschiedliche Benutzerschnittstellen, um auf die Satzspiegelkonstruktion Einfluss zu nehmen. Die wichtigste Möglichkeit ist die Angabe von Optionen. Wie im vorherigen Abschnitt erwähnt, können die Optionen dabei auf unterschiedlichen Wegen gesetzt werden.

Hinweis! In diesem Abschnitt wird die Klasse *protokol* verwendet werden. Es handelt sich dabei nicht um eine KOMA-Script-Klasse, sondern um eine hypothetische Klasse. Dieses Buch geht von dem Idealfall aus, dass für jede Aufgabe eine dafür passende Klasse zur Verfügung steht.

BCOR=*Korrektur*

Bindekorrektur

Mit Hilfe der Option *BCOR=Korrektur* geben Sie den absoluten Wert der Bindekorrektur an, also die Breite des Bereichs, der durch die Bindung von der Papierbreite verloren geht. Dieser Wert wird in der Satzspiegelkonstruktion automatisch berücksichtigt und bei der Ausgabe wieder dem inneren beziehungsweise linken Rand zugeschlagen. Als *Korrektur* können Sie jede von $\text{T}_{\text{E}}\text{X}$ verstandene Maßeinheit angeben.

Beispiel: Angenommen, Sie erstellen einen Finanzbericht. Das Ganze soll einseitig in A4 gedruckt und anschließend in eine Klemmmappe geheftet werden. Die Klemme der Mappe verdeckt 7,5 mm. Der Papierstapel ist sehr dünn, deshalb gehen beim Knicken und Blättern durchschnittlich höchstens weitere 0,75 mm verloren. Sie schreiben dann also:

```
\documentclass[a4paper]{report}
\usepackage[BCOR=8.25mm]{typearea}
```

mit *BCOR=8.25mm* als Option für *typearea* oder

```
\documentclass[a4paper,BCOR=8.25mm]{report}
\usepackage{typearea}
```

zur Angabe von *BCOR=8.25mm* als globale Option.

Bei Verwendung einer KOMA-Script-Klasse kann das explizite Laden von *typearea* entfallen:

```
\documentclass[BCOR=8.25mm]{scrreprt}
```

Die Option *a4paper* konnte bei *scrreprt* entfallen, da diese der Voreinstellung bei allen KOMA-Script-Klassen entspricht.

Setzt man die Option erst später auf einen neuen Wert, verwendet man also beispielsweise

```
\documentclass{scrreprt}
\KOMAoptions{BCOR=8.25mm}
```

so werden bereits beim Laden der Klasse `scrreprt` Standardeinstellungen vorgenommen. Beim Ändern der Einstellung mit Hilfe einer der Anweisung `\KOMAoptions` oder `\KOMAoption` wird dann automatisch ein neuer Satzspiegel mit neuen Randeinstellungen berechnet.

Bitte beachten Sie unbedingt, dass diese Option bei Verwendung einer der KOMA-Script-Klassen wie im Beispiel als Klassenoption oder per `\KOMAoptions` beziehungsweise `\KOMAoption` nach dem Laden der Klasse übergeben werden muss. Weder sollte das Paket `typearea` bei Verwendung einer KOMA-Script-Klasse explizit per `\usepackage` geladen, noch die Option dabei als optionales Argument angegeben werden. Wird die Option per `\KOMAoptions` oder `\KOMAoption` nach dem Laden des Pakets geändert, so werden Satzspiegel und Ränder automatisch neu berechnet.

Achtung!

*automatische
Neuberechnung*

DIV=Faktor

Mit Hilfe der Option `DIV=Faktor` wird festgelegt, in wie viele Streifen die Seite horizontal und vertikal bei der Satzspiegelkonstruktion eingeteilt wird. Die genaue Konstruktion ist Abschnitt 2.2 zu entnehmen. Wichtig zu wissen ist, dass gilt: Je größer der *Faktor*, desto größer wird der Textbereich und desto kleiner die Ränder. Als *Faktor* kann jeder ganzzahlige Wert ab 4 verwendet werden. Bitte beachten Sie jedoch, dass sehr große Werte dazu führen können, dass Randbedingungen der Satzspiegelkonstruktion, je nach Wahl der weiteren Optionen, verletzt werden. So kann die Kopfzeile im Extremfall auch außerhalb der Seite liegen. Bei Verwendung der Option `DIV=Faktor` sind Sie für die Einhaltung der Randbedingungen sowie eine nach typografischen Gesichtspunkten günstige Zeilenlänge selbst verantwortlich.

In Tabelle 2.1 finden Sie für das Seitenformat A4 ohne Bindekorrektur die aus einigen *DIV*-Faktoren resultierenden Satzspiegelgrößen. Dabei werden die weiteren von der Schriftgröße abhängigen Nebenbedingungen nicht berücksichtigt.

Beispiel: Angenommen, Sie schreiben ein Sitzungsprotokoll. Sie verwenden dafür die Klasse `protokol`. Das Ganze soll doppelseitig werden. In Ihrer Firma wird die Schriftart Bookman in 12 pt verwendet. Diese Schriftart, die zu den

Tabelle 2.1.: Satzspiegelmaße in Abhängigkeit von *DIV* bei A4 ohne Berücksichtigung von `\topskip` oder *BCOR*

<i>DIV</i>	Satzspiegel		Ränder	
	Breite	Höhe	oben	innen
6	105,00	148,50	49,50	35,00
7	120,00	169,71	42,43	30,00
8	131,25	185,63	37,13	26,25
9	140,00	198,00	33,00	23,33
10	147,00	207,90	29,70	21,00
11	152,73	216,00	27,00	19,09
12	157,50	222,75	24,75	17,50
13	161,54	228,46	22,85	16,15
14	165,00	233,36	21,21	15,00
15	168,00	237,60	19,80	14,00

(alle Längen in mm)

Standard-PostScript-Schriften gehört, wird in \LaTeX mit der Anweisung `\usepackage{bookman}` aktiviert. Die Schriftart Bookman läuft sehr weit, das heißt, die einzelnen Zeichen sind im Verhältnis zur Höhe relativ breit. Deshalb ist Ihnen die Voreinstellung für den *DIV*-Wert in *typearea* zu gering. Statt einem Wert von 12 sind Sie nach gründlichem Studium dieses Kapitels einschließlich der weiterführenden Abschnitte überzeugt, dass der Wert 15 angebracht ist. Das Protokoll wird nicht gebunden, sondern gelocht und in einen Ordner abgeheftet. Eine Bindekorrektur ist deshalb nicht notwendig. Sie schreiben also:

```
\documentclass[a4paper,twoside]{protokol}
\usepackage{bookman}
\usepackage[DIV=15]{typearea}
```

Als Sie fertig sind, macht man Sie darauf aufmerksam, dass die Protokolle neuerdings gesammelt und am Quartalsende alle zusammen als Buch gebunden werden. Die Bindung erfolgt als einfache Leimbindung, weil den Band ohnehin nie wieder jemand anschaut und er nur wegen ISO 9000 angefertigt wird. Für die Bindung einschließlich Biegefalz werden durchschnittlich 12 mm benötigt. Sie ändern die Optionen von *typearea* also entsprechend ab und verwenden die Klasse für Protokolle nach ISO 9000:

```
\documentclass[a4paper,twoside]{iso9000p}
\usepackage{bookman}
\usepackage[DIV=15,BCOR=12mm]{typearea}
```

Tabelle 2.2.: DIV-Voreinstellungen für A4

Grundschriftgröße:	10 pt	11 pt	12 pt
DIV:	8	10	12

Natürlich können Sie auch hier wieder eine KOMA-Script-Klasse verwenden:

```
\documentclass[twoside,DIV=15,BCOR=12mm]{scrartcl}
\usepackage{bookman}
```

Option `a4paper` entspricht der Voreinstellung und konnte daher entfallen.

Bitte beachten Sie unbedingt, dass die Option `DIV` bei Verwendung einer der KOMA-Script-Klassen wie im Beispiel als Klassenoption oder per `\KOMAOPTIONS` beziehungsweise `\KOMAOPTION` nach dem Laden der Klasse übergeben werden muss. Weder sollte das Paket `typearea` bei Verwendung einer KOMA-Script-Klasse explizit per `\usepackage` geladen, noch die Option dabei als optionales Argument angegeben werden. Wird die Option per `\KOMAOPTIONS` oder `\KOMAOPTION` nach dem Laden des Pakets geändert, so werden Satzspiegel und Ränder automatisch neu berechnet.

Achtung!

*automatische
Neuberechnung*

```
DIV=calc
DIV=classic
```

Wie bereits in Abschnitt 2.2 erwähnt, gibt es nur für das Papierformat A4 feste Voreinstellungen für den `DIV`-Wert. Diese sind Tabelle 2.2 zu entnehmen. Solche festen Werte haben allerdings den Nachteil, dass sie die Laufweite der verwendeten Schrift nicht berücksichtigen. Das kann bei A4 und recht schmalen Schriften sehr rasch zu unangenehm hoher Zeichenzahl je Zeile führen. Siehe hierzu die Überlegungen in Abschnitt 2.1. Wird ein anderes Papierformat gewählt, so berechnet `typearea` selbst einen guten `DIV`-Wert. Natürlich können Sie diese Berechnung auch für A4 wählen. Hierzu verwenden Sie `DIV=calc` anstelle von `DIV=Faktor`. Selbstverständlich können Sie diese Option auch explizit bei allen anderen Papierformaten angeben. Wenn Sie die automatische Berechnung wünschen, ist diese Angabe sogar sinnvoll, da die Möglichkeit besteht, in einer Konfigurationsdatei andere Voreinstellungen zu setzen (siehe Abschnitt 20.3). Eine explizit angegebene Option `DIV=calc` überschreibt diese Vorkonfiguration aber.

Achtung!

`DIV=calc`

Die in Abschnitt 2.3 erwähnte klassische Konstruktion, der mittelalterliche Buchseitenkanon, ist ebenfalls auswählbar. Verwenden Sie in diesem Fall anstelle von `DIV=Faktor` oder `DIV=calc` einfach `DIV=`

`DIV=classic` `classic`. Es wird dann ein *DIV*-Wert ermittelt, der eine möglichst gute Näherung an den mittelalterlichen Buchseitenkanon darstellt.

Beispiel: In dem bei der Option *DIV=Faktor* aufgeführten Beispiel mit der Schriftart Bookman gab es ja genau das Problem, dass man einen zur Schriftart besser passenden *DIV*-Wert haben wollte. Man könnte also in Abwandlung des ersten Beispiels auch einfach die Ermittlung dieses Wertes *typearea* überlassen:

```
\documentclass[a4paper,twoside]{protokol}
\usepackage{bookman}
\usepackage[DIV=calc]{typearea}
```

`DIV=current`
`DIV=last`

Wenn Sie bis hier die Beispiele aufmerksam verfolgt haben, wissen Sie eigentlich bereits, wie man die Berechnung eines *DIV*-Wertes in Abhängigkeit von der gewählten Schrift erreicht, wenn eine KOMA-Script-Klasse zusammen mit einem Schriftpaket verwendet wird.

Das Problem dabei ist, dass die KOMA-Script-Klasse das Paket *typearea* bereits selbst lädt. Die Übergabe der Optionen als optionale Argumente von `\usepackage` ist also nicht möglich. Es würde auch nichts nützen, die Option *DIV=calc* als optionales Argument von `\documentclass` anzugeben. Diese Option würde direkt beim Laden des Pakets *typearea* ausgewertet. Damit würden Satzspiegel und Ränder für die L^AT_EX-Standardschrift und nicht für die später geladene Schrift berechnet.

Selbstverständlich ist es möglich, mit `\KOMAOPTIONS{DIV=calc}` oder `\KOMAoption{DIV}{calc}` nach dem Laden des Schriftpakets Satzspiegel und Ränder neu berechnen zu lassen. Dabei wird dann über den Wert *calc* direkt ein *DIV*-Wert für eine gute Zeilenlänge eingefordert.

Da es aber häufig praktischer ist, die Einstellung für die Option *DIV* nicht erst nach dem Laden der Schrift vorzunehmen, sondern an herausgehobener Stelle, beispielsweise beim Laden der Klasse, bietet *typearea* zwei weitere symbolische Werte für diese Option.

`DIV=current` Mit *DIV=current* wird eine Neuberechnung von Satzspiegel und Rändern angestoßen, wobei genau der *DIV*-Wert verwendet wird, der aktuell eingestellt ist. Dies ist weniger für die Neuberechnung des Satzspiegels nach Wahl einer anderen Grundschrift von Interesse. Vielmehr ist das dann nützlich, wenn man etwa nach Änderung des Durchschusses bei Beibehaltung des Teilers *DIV* die Randbedin-

gung sicherstellen will, dass `\textheight` abzüglich `\topskip` ein Vielfaches von `\baselineskip` sein sollte.

Mit `DIV=last` wird eine Neuberechnung von Satzspiegel und Rändern angestoßen, wobei genau dieselbe Einstellung wie bei der letzten Berechnung verwendet wird.

`DIV=last`

Beispiel: Gehen wir wieder davon aus, dass für die Schriftart Bookman ein Satzspiegel mit guter Zeilenlänge berechnet werden soll. Gleichzeitig wird eine KOMA-Script-Klasse verwendet. Dies ist mit dem symbolischen Wert `last` und der Anweisung `\KOMAOPTIONS` sehr einfach möglich:

```
\documentclass[BCOR=12mm,DIV=calc,twoside]
{scrartcl}
\usepackage{bookman}
\KOMAOPTIONS{DIV=last}
```

Wird später entschieden, dass ein anderer *DIV*-Wert verwendet werden soll, so muss nur die Einstellung im optionalen Argument von `\documentclass` geändert werden.

Eine Zusammenfassung aller möglichen symbolischen Werte für die Option `DIV` finden Sie in Tabelle 2.3. Es wird an dieser Stelle darauf hingewiesen, dass auch die Verwendung des Pakets `fontenc` dazu führen kann, dass \LaTeX eine andere Schrift lädt.

Häufig wird die Satzspiegelneuberechnung im Zusammenhang mit der Veränderung des Zeilenabstandes (*Durchschuss*) benötigt. Da der Satzspiegel unbedingt so berechnet werden sollte, dass eine ganze Anzahl an Zeilen in den Textbereich passt, muss bei Verwendung eines anderen Durchschusses als des normalen der Satzspiegel für diesen Zeilenabstand neu berechnet werden.

Achtung!

Beispiel: Angenommen, für eine Diplomarbeit wird die Schriftgröße 10 pt bei eineinhalbzeiligem Satz zwingend gefordert. \LaTeX setzt normalerweise bei 10 pt mit 2 pt Durchschuss, also 1,2-zeilig. Deshalb muss als zusätzlicher Dehnfaktor der Wert 1,25 verwendet werden. Gehen wir außerdem davon aus, dass eine Bindekorrektur von 12 mm benötigt wird. Dann könnte die Lösung dieses Problems wie folgt aussehen:

```
\documentclass[10pt,twoside,BCOR=12mm,DIV=calc]
{scrreprt}
\linespread{1.25}
\KOMAOPTIONS{DIV=last}
```

Da `typearea` selbst immer die Anweisung `\normalsize` bei Berechnung eines neuen Satzspiegels ausführt, ist es nicht

Tabelle 2.3.: Mögliche symbolische Werte für die Option DIV oder das *DIV*-Argument der Anweisung `\typearea[BCOR]{DIV}`

<code>areaset</code>	Satzspiegel neu anordnen.
<code>calc</code>	Satzspiegelberechnung einschließlich Ermittlung eines guten <i>DIV</i> -Wertes erneut durchführen.
<code>classic</code>	Satzspiegelberechnung nach dem mittelalterlichen Buchseitenkanon (Kreisberechnung) erneut durchführen.
<code>current</code>	Satzspiegelberechnung mit dem aktuell gültigen <i>DIV</i> -Wert erneut durchführen.
<code>default</code>	Satzspiegelberechnung mit dem Standardwert für das aktuelle Seitenformat und die aktuelle Schriftgröße erneut durchführen. Falls kein Standardwert existiert, <code>calc</code> anwenden.
<code>last</code>	Satzspiegelberechnung mit demselben <i>DIV</i> -Argument, das beim letzten Aufruf angegeben wurde, erneut durchführen.

`setspace` zwingend notwendig, nach `\linespread` den gewählten Durchschuss mit `\selectfont` zu aktivieren, damit dieser auch tatsächlich für die Neuberechnung verwendet wird.

Das gleiche Beispiel sähe unter Verwendung des `setspace`-Pakets (siehe [TF11]) wie folgt aus:

```
\documentclass[10pt,twoside,BCOR=12mm,DIV=calc]{scrreprt}
\usepackage[onehalfspacing]{setspace}
\KOMAoptions{DIV=last}
```

Tipp! Wie man an dem Beispiel sieht, spart man sich mit dem `setspace`-Paket das Wissen um den korrekten Dehnungswert. Dies gilt allerdings nur für die Standardschriftgrößen 10 pt, 11 pt und 12 pt. Für alle anderen Schriftgrößen verwendet das Paket einen näherungsweise passenden Dehnungswert.

Achtung! An dieser Stelle erscheint es mir angebracht, darauf hinzuweisen, dass der Zeilenabstand für die Titelseite wieder auf den normalen Wert zurückgesetzt werden sollte und außerdem auch die Verzeichnisse mit dem normalen Zeilenabstand gesetzt werden sollten.