

Das geometry Paket

Hideo Umeki

<https://github.com/davidcarlisle/geometry>

Die Übersetzung wurde angefertigt von Hans-Martin Haase (hmhaase@pclinuxosusers.de)

2020/01/02 v5.9

Zusammenfassung

Dieses Paket bietet einen flexiblen und einfachen Weg die Seitenmaße zu verändern. Sie können das Seitenlayout mit intuitiven Parametern verändern. Ein Beispiel, wenn Sie den Abstand von jedem Rand der Seite auf 2 cm setzen möchten, können Sie einfach `\usepackage[margin=2cm]{geometry}` eingeben. Das Seitenlayout kann in der Mitte des Dokuments verändert werden, mit dem Befehl `\newgeometry`.

1 Vorwort zu Version 5

- **Das Seitenlayout in der Mitte des Dokuments ändern.**

Die neuen Befehle `\newgeometry{...}` und `\restoregeometry` erlauben es dem Benutzer, die Seitenmaße in der Mitte des Dokuments zu verändern. `\newgeometry` stimmt fast mit `\geometry` überein, ausgenommen das `\newgeometry` alle Optionen, die in der Präambel spezifiziert sind, deaktiviert und überspringt die Optionen, die sich auf die Seitengröße beziehen: `landscape`, `portrait` und die Seitengrößenoptionen (solche wie `papersize`, `paper=a4paper` und andere).

- **Eine neue Reihe von Optionen, um den Layoutbereich zu spezifizieren.**

Die Optionen, die für den Bereich angegeben wurden, in dem die Seitenmaße berechnet werden, sind hinzugefügt wurden: `layout`, `layoutsize`, `layoutwidth`, `layoutheight` und weitere. Diese Möglichkeiten sind hilfreich, falls das angegebene Layout auf eine andere Papiergröße angewendet soll. Wenn beispielsweise `a4paper` und `layout=a5paper` verwendet wird, dann benutzt das `geometry` Paket das 'A5' statt 'A4' Layout, um die Ränder für die Papiergröße zu berechnen.

- **Eine neue Treiberoption `xetex`.**

Die neue Treiberoption `xetex` wurde hinzugefügt. Die Routine für die automatische Treiberdetektion ist überarbeitet worden, sodass ein Fehler mit einer nicht definierten Kontrollsequenz vermieden werden konnte. Beachte, dass 'geometry.cfg' in `TEX Live`, welches die Autodetektionsroutine deaktiviert und `pdftex` benutzt, nicht mehr nötig ist und es keine Probleme gibt, obwohl dies dennoch existiert. Um `xetex` einzustellen, ist es dringend empfohlen `XYLATEX` zu verwenden.

- **Neue Papiergröße für JIS B-Serien und ISO C-Serien.**

Die Papiergrößenvoreinstellungen `b0j` bis `b6j` für JIS (Japanese Industrial Standards) B-Serien und `c0paper` bis `c6paper` für ISO C-Serien (v5.4~) wurden hinzugefügt.

- **Veränderung des Standards für nicht angegebenen Rand.**

In der Vorversion wurde dann, wenn man nur einen Rand angab, z. B. `bottom=1cm`, von `geometry` der andere Rand mit diesem Randverhältnis (1:1 standardmäßig für die vertikalen Maße) gesetzt und man bekam in diesem Fall `top=1cm`. Die Version 5 stellt die Textkörpergröße auf den `scale` (= 0.7) ein und bestimmt den nicht angegebenen Rand. (Siehe Kapitel 6.5)

- **Die Optionen `showframe` und `showcrop` arbeiten auf jeder Seite.**

Mit der `showframe` Option werden die Seitenränder auf jeder Seite angezeigt. Des Weiteren druckt die neue `showcrop` Option Schittmarken auf jede Ecke des Layoutbereichs auf jeder Seite. Bedenken sie, dass die Marken unsichtbar sein werden, solange die Layoutgröße nicht schmaler wie die Papiergröße gemacht wird. Version 5.4 führt einen neuen `\shipout` Überladeprozess ein, der das `atbegshi` Paket benutzt, so wird das `atbegshi` Paket benötigt, wenn die `showframe` oder `showcrop` Option angegeben wird.

- **Laden von geometry.cfg vor der Verarbeitung der Klassenoptionen.**

In der Vorgängerversion wurde geometry.cfg nach dem Verarbeiten der Dokumentklassenoptionen geladen. Jetzt, da die Konfigurationsdatei vor der Abarbeitung der Klassenoptionen geladen wird, können sie das Verhalten, wie es in geometry.cfg angegeben ist, durch das Hinzufügen von Optionen in der \documentclass oder genauso gut in \usepackage und \geometry verändern.

- **Beseitigte Optionen:** compat2 **und** twosideshift. Die Version 5 ist nicht länger mit den Vorgängerversionen kompatibel. compat2 und twosideshift wurden zur Vereinfachung entfernt.

2 Einführung

Die Maße für das Seitenlayout in \TeX einzustellen ist nicht so einfach. Sie müssen dafür verschiedene ursprüngliche \TeX -Maße anpassen, um einen Textbereich dort zu platzieren, wo sie ihn angezeigt haben möchten. Wenn sie einen Textbereich ihres Dokuments zentrieren wollen, dann müssen sie beispielsweise die systemeigenen Maße wie folgt angeben:

```
\usepackage{calc}
\setlength\textwidth{7in}
\setlength\textheight{10in}
\setlength\oddsidemargin{(\paperwidth-\textwidth)/2 - 1in}
\setlength\topmargin{(\paperheight-\textheight
-\headheight-\headsep-\footskip)/2 - 1in}.
```

Ohne das Paket calc würde das obere Beispiel noch schwierigere Einstellungen benötigen. Das Paket geometry bietet einen einfachen Weg, die Parameter des Seitenlayouts zu konfigurieren. In diesem Fall müssen sie nur dies einstellen:

```
\usepackage[text={7in,10in},centering]{geometry}.
```

Neben dem Zentrierungsproblem, ist das Einstellen der Ränder von jeder Kante des Blattes aus ebenso problematisch. Aber geometry erledigt dies ebenso einfach. Wenn sie jeden Rand auf 1.5 in setzen wollen, dann können sie

```
\usepackage[margin=1.5in]{geometry}
```

eingeben. Weil das geometry Paket eine Autovervollständigung besitzt, werden nicht angegebene Maße automatisch festgelegt. Das geometry Paket wird auch nützlich sein, wenn sie das Seitenlayout beispielsweise nach den folgenden strikten Instruktionen einstellen müssen:

Die vollständig erlaubte Breite des Textbereiches ist 6.5, bei einer Höhe von 8.75 Inches (englische Zoll). Der obere Rand auf jeder Seite soll 1.2 Inches von der oberen Kante der Seite aus betragen. Der linke Rand sollte 0.9 Inch von der linken Kante entfernt sein. Die Fußzeile mit der Seitenzahl sollte am unteren Ende des Textbereiches sein.

In diesem Fall können sie bei der Benutzung von geometry

```
\usepackage[total={6.5in,8.75in},
top=1.2in, left=0.9in, includefoot]{geometry}.
```

eingeben.

Die Konfiguration eines Textbereiches in einem Schriftstück mit dem Textsatzprogramm hat einige Ähnlichkeiten mit dem Platzieren eines Fensters auf dem Hintergrund beim Fenstermanager des Betriebssystems. Der Name 'geometry' kommt von der geometry-Option, die genutzt wird, um die Größe und Lokalisierung eines Fensters beim X-Window-System anzugeben.

3 Seitengeometrie

Abbildung 1 zeigt die Seitenlayoutmaße, die im geometry Paket definiert sind. Das Seitenlayout beinhaltet einen *total body* (druckbaren Bereich) und *margins* (Ränder). Der *total body* besteht aus einem *body* (Textbereich) mit einem optionalen *header* (Kopf), *footer* (Fuß) und Randbemerkungen (*marginpar*). Es gibt vier Ränder: *left* (links), *right* (rechts), *top* (oben) und *bottom* (unten). Für zweiseitige Dokumente, sollten horizontale Ränder *inner* (innen) und *outer* (außen) genannt werden.

`paper` : `total body` und `margins`
`total body` : `body` (Textbereich) (optionaler `head`, `foot` und `marginpar`)
`margins` : `left` (*inner*), `right` (*outer*), `top` und `bottom`

Jeder Rand wird von der korrespondierenden Kante des Blattes gemessen. Zum Beispiel, linker Rand (innerer Rand) meint eine horizontale Entfernung zwischen der linken (inneren) Kante des Blattes und der des `total body` (druckbaren Bereichs). Deshalb sind der linke und obere Rand in `geometry` anders als das natürliche Maß `\leftmargin` und `\topmargin` definiert. Die Größe eines Body (Textbereichs) kann durch `\textwidth` (Textbreite) und `\textheight` (Texthöhe) angepasst werden. Die Maße für die Seite, den druckbaren Bereich und die Ränder stehen in folgenden Beziehungen.

$$\text{paperwidth (Seitenbreite)} = \text{left (links)} + \text{width (Breite)} + \text{right (rechts)} \quad (1)$$

$$\text{paperheight (Seitenhöhe)} = \text{top (oben)} + \text{height (Höhe)} + \text{bottom (unten)} \quad (2)$$

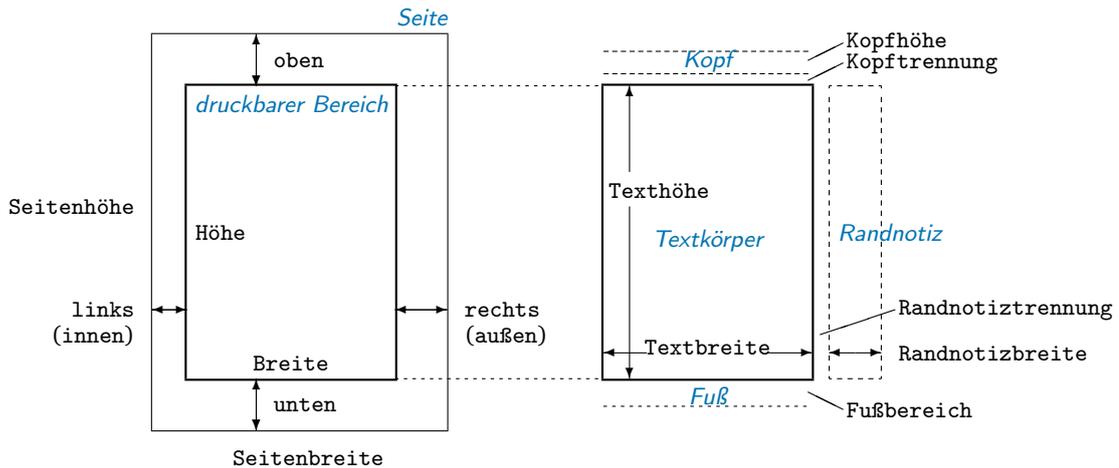


Abbildung 1: Namen von Maßen, die im `geometry` Paket genutzt werden: `width` = `textwidth` (Textbreite) und `height` = `textheight` (Texthöhe) sind Standard. `left` (`links`), `right` (`rechts`), `top` (`oben`) und `bottom` (`unten`) sind Ränder. Wenn die Ränder auf der Rückseite durch die `twoside` (Zweiseitigkeit) Option getauscht werden, dann werden die Ränder, die mit `left` und `right` angegeben wurden, für die inneren und äußeren Ränder genutzt. `inner` (`innen`) und `outer` (`außen`) sind Pseudonyme für `left` und `right`.

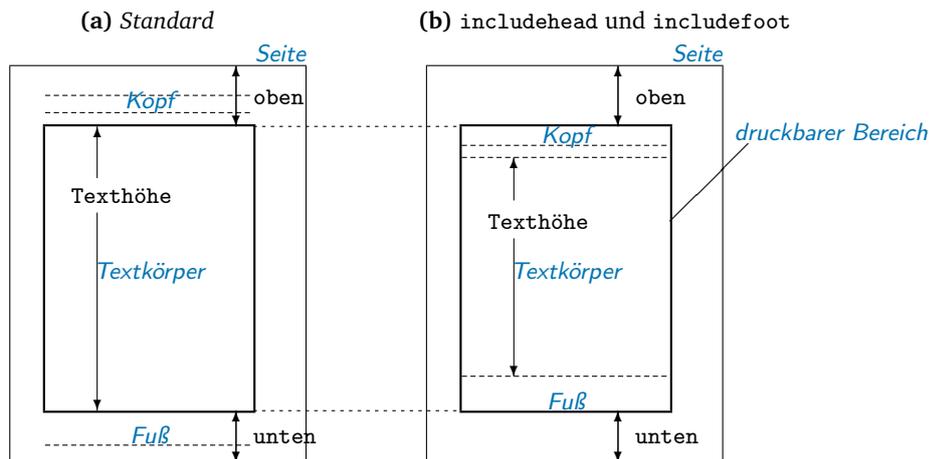


Abbildung 2: `includehead` und `includefoot` schließen den Kopf und den Fuß mit in den `total body` (druckbaren Bereich) ein. (a) `height` = `textheight` (Standard). (b) `height` = `textheight` + `headheight` + `headsep` + `footskip`, wenn `includehead` und `includefoot` aktiv. Wenn der obere und untere Rand angegeben ist führt `includehead` und `includefoot` dazu, dass die Texthöhe verringert wird.

Die Breite und Höhe des druckbaren Bereichs wird definiert:

$$\text{width} := \text{textwidth} (+ \text{marginparsep} * + \text{marginparwidth} * *) \quad (3)$$

$$\text{height} := \text{textheight} (+ \text{headheight} * * * + \text{headsep} * * * * + \text{footskip} * * * * *) \quad (4)$$

* Randnotiztrennung ** Randnotizbreite *** Kopfhöhe **** Kopftrennung ***** Fußbereich

In Gleichung (3) ist standardmäßig `width:=textwidth`, während `marginparsep` und `marginparwidth` nur in der `width` (Breite) enthalten sind, wenn die `includeemp` Option auf `true` gesetzt ist. In Gleichung (4) ist standardmäßig `height:=textheight`. Falls `includehead` auf `true` gesetzt ist, dann werden `headheight` und `headsep` als Teil der Höhe (`height`) betrachtet. Auf dem selben Weg, nimmt `includefoot` `footskip` in `height` auf. Abbildung 2 zeigt, wie diese Optionen in der vertikalen Richtung arbeiten.

Folglich besteht das Seitenlayout aus drei Teilen (Längen) in jeder Richtung: ein Textkörper und zwei Ränder. Wenn zwei der Ränder explizit angegeben sind, dann sind die anderen offensichtlich und müssen nicht angegeben werden. Abbildung 3 zeigt ein einfaches Modell der Seitenmaße. Falls eine Länge `L` gegeben ist und eingeteilt ist in den Textkörper `b`, die Ränder `a` und `c`, dann ist offensichtlich, dass

$$L = a + b + c \tag{5}$$

Die Spezifikation mit zwei von drei (`a`, `b` und `c`) festgelegten, ist explizit lösbar. Wenn zwei oder mehr der übrigen nicht angegeben sind, kann Gleichung (5) nicht ohne eine andere Beziehung zwischen ihnen gelöst werden. Falls alle Werte angegeben sind, dann muss überprüft werden, ob diese Gleichung (5) genügen, da es zu viele Angaben sind oder es ist 'überspezifiziert'.

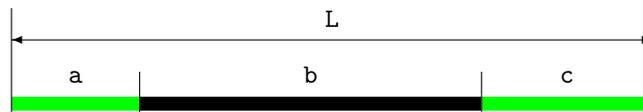


Abbildung 3: Ein einfaches Modell der Seitenmaße.

Das `geometry` Paket besitzt einen Autovervollständigungsmechanismus, der vor dem Ärger der Angabe von Seitenlayoutmaßen schützt. Zum Beispiel können sie das Folgende

```
\usepackage[width=14cm, left=3cm]{geometry}
```

auf A4-Papier einstellen. In diesem Fall müssen sie den rechten Rand nicht angeben. Die Details der Autovervollständigung werden in Kapitel 6.5 beschrieben.

4 Benutzerschnittstelle

4.1 Befehle

Das `geometry` Paket unterstützt die folgenden Befehle:

- `\geometry{<Optionen>}`
- `\newgeometry{<Optionen>}` und `\restoregeometry`
- `\savegeometry{<Name>}` und `\loadgeometry{<Name>}`

`\geometry{<Optionen>}` verändert das Seitenlayout entsprechend den angegebenen Optionen im Argument. Dieser Befehl, wenn überhaupt, sollte nur in der Präambel platziert werden (vor `\begin{document}`).

Das `geometry` Paket kann als Teil einer Klasse oder eines weiteren Pakets, das sie im Dokument benutzen, genutzt werden. Der Befehl `\geometry` kann einige der Einstellungen in der Präambel überschreiben. Eine mehrfache Nutzung von `\geometry` ist erlaubt und wird dann mit den verbundenen Optionen ausgeführt. Wenn `geometry` noch nicht geladen ist, können sie nur `\usepackage[<Optionen>]{geometry}` anstelle von `\geometry` nutzen.

`\newgeometry{<Optionen>}` verändert das Seitenlayout innerhalb des Dokuments. `\newgeometry` ist fast gleich mit `\geometry`, ausgenommen, dass `\newgeometry` alle Optionen, die von `\usepackage` und `\geometry` in der Präambel vorgenommen wurden, deaktiviert. `\restoregeometry` stellt das Seitenlayout, wie es in der Präambel angegeben ist, wieder her. Dieser Befehl besitzt keine Argumente. Lesen sie Kapitel 7 für Details.

`\savegeometry{<Name>}` speichert die Seitenmaße als `<Name>`, wo sie diesen Befehl nutzen. `\loadgeometry{<Name>}` lädt die Seitenmaße, die als `<Name>` gespeichert wurden. Lesen sie Kapitel 7 für Details.

4.2 Optionales Argument

Das geometry Paket übernimmt die keyval Schnittstelle ‘ $\langle\text{Schlüssel}\rangle=\langle\text{Wert}\rangle$ ’ für das optionale Argument von `\usepackage`, `\geometry` und `\newgeometry`.

Das Argument beinhaltet eine Liste von kommaseparierten keyval Optionen und besitzt folgende Grundregeln:

- Mehrfache Zeilen sind erlaubt, während es leere nicht sind.
- Jedes Leerzeichen zwischen den Wörtern wird ignoriert.
- Die Reihenfolge der Optionen ist grundlegend unabhängig. (Es gibt einige Ausnahmen. Lesen sie Kapitel 6.2 für Details.)

Zum Beispiel ist

```
\usepackage[ a5paper , hmargin = { 3cm,
      .8in } , height
      = 10in ]{geometry}
```

äquivalent zu

```
\usepackage[height=10in,a5paper,hmargin={3cm,0.8in}]{geometry}
```

Einigen Optionen ist es erlaubt, Unterlisten zu haben, z. B. `{3cm,0.8in}`. Beachten sie, dass die Reihenfolge der Werte in der Liste bedeutsam ist. Die obere Einstellung ist ebenso äquivalent zu Folgendem:

```
\usepackage{geometry}
\geometry{height=10in,a5paper,hmargin={3cm,0.8in}}
```

oder

```
\usepackage[a5paper]{geometry}
\geometry{hmargin={3cm,0.8in},height=8in}
\geometry{height=10in}.
```

Folglich hängt die mehrfache Nutzung von `\geometry` die Optionen aneinander. `geometry` unterstützt das Paket *calc*¹. Zum Beispiel,

```
\usepackage{calc}
\usepackage[ $\text{textheight}=20\backslash\text{baselineskip}+10\text{pt}$ ]{geometry}
```

4.3 Optionstypen

`geometry` Optionen sind in vier Typen eingeteilt:

1. Wahrheitswerttyp

Er nimmt einen Wahrheitswert an (`true` (wahr) oder `false` (falsch)). Falls kein Wert angegeben ist, wird standardmäßig `true` (wahr) eingesetzt.

$\langle\text{Schlüssel}\rangle=\text{true} \mid \text{false}$.
 $\langle\text{Schlüssel}\rangle$ ohne Wert ist äquivalent zu $\langle\text{Schlüssel}\rangle=\text{true}$.

Beispiele: `verbose=true, includehead, twoside=false`.

Der Papiername ist eine Ausnahme. Der bevorzugte Papiername sollte ohne Wert angegeben werden. Welcher Wert auch immer gegeben ist er wird ignoriert. Zum Beispiel ist `a4paper=XXX` äquivalent zu `a4paper`.

2. Typ mit einzelner Wert

Man benötigt einen zwingenden Wert.

$\langle\text{Schlüssel}\rangle=\langle\text{Wert}\rangle$.

Beispiele: `width=7in, left=1.25in, footskip=1cm, height=.86\paperheight`.

¹CTAN: macros/latex/required/tools

3. Typ mit zwei Werten

Man benötigt ein Paar von kommasetrennten Werten in geschweiften Klammern. Die zwei Werte können zu einem gekürzt werden, wenn beide identisch sind.

$\langle\text{Schlüssel}\rangle=\{\langle\text{Wert1}\rangle,\langle\text{Wert2}\rangle\}$.

$\langle\text{Schlüssel}\rangle=\langle\text{Wert}\rangle$ ist äquivalent zu $\langle\text{Schlüssel}\rangle=\{\langle\text{Wert}\rangle,\langle\text{Wert}\rangle\}$.

Beispiele: `hmargin={1.5in,1in}`, `scale=0.8`, `body={7in,10in}`.

4. Typ mit drei Werten

Man benötigt drei zwingende Werte, kommasetrennte Werte in geschweiften Klammern.

$\langle\text{Schlüssel}\rangle=\{\langle\text{Wert1}\rangle,\langle\text{Wert2}\rangle,\langle\text{Wert3}\rangle\}$

Jeder Wert muss ein Maß sein oder Null. Falls sie einen leeren Wert angeben oder '*', dies bedeutet Null, dann wird der übergebene Wert dem Autovervollständigungsmechanismus zugeführt. Sie müssen mindestens ein Maß angeben, typischerweise aber zwei. Sie können für alle Werte Null einsetzen, aber das macht keinen Sinn. *Beispiele:*

`hdivide={2cm,*,1cm}`, `vdivide={3cm,19cm, }`, `divide={1in,*,1in}`.

5 Details zu Optionen

Dieses Kapitel beschreibt alle Optionen, die in geometry verfügbar sind. Optionen mit einem Dolch † sind nicht vorhanden als Argumente von `\newgeometry` (Lesen sie Kapitel 7).

5.1 Papiergröße

Die Optionen unten dienen zum Einstellen der Größe des Papiers/Mediums und der Orientierung.

- † `paper` (Papier) | `papername` (Papiername)
gibt die Papiergröße anhand des Namens an. `paper=\langle\text{Papiername}\rangle`. Für besseren Komfort können sie den Papiernamen ohne `paper=` angeben. `a4paper` ist beispielsweise äquivalent zu `paper=a4paper`.
- † `a0paper`, `a1paper`, `a2paper`, `a3paper`, `a4paper`, `a5paper`, `a6paper`,
`b0paper`, `b1paper`, `b2paper`, `b3paper`, `b4paper`, `b5paper`, `b6paper`,
`c0paper`, `c1paper`, `c2paper`, `c3paper`, `c4paper`, `c5paper`, `c6paper`,
`b0j`, `b1j`, `b2j`, `b3j`, `b4j`, `b5j`, `b6j`,
`ansiapaper`, `ansibpaper`, `ansicpaper`, `ansidpaper`, `ansiepaper`,
`letterpaper`, `executivepaper`, `legalpaper`
gibt den Papiernamen an. Der Wert wird ignoriert, sogar wenn einer angegeben wird. Beispielsweise hat das Folgende den gleichen Effekt: `a5paper`, `a5paper=true`, `a5paper=false` und so weiter. `a[0-6]paper`, `b[0-6]paper` und `c[0-6]paper` sind ISO A, B und C Serien von Papiergrößen. Die JIS (Japanischer Industriestandard) A-Serien ist identisch zu den ISO A-Serien, aber die JIS B-Serien sind von den ISO B-Serien verschieden. `b[0-6]j` sollte für die JIS B-Serien genutzt werden.
- † `screen` eine spezielle Papiergröße mit (B,H) = (225 mm,180 mm). Für Präsentationen mit dem PC und einem Beamer kann "`screen,centering`", in Verbindung mit 'slide' in der Dokumentklasse nützlich sein.
- † `paperwidth` Breite des Papiers. `paperwidth=\langle\text{Länge}\rangle`.
- † `paperheight` Höhe des Papiers. `paperheight=\langle\text{Länge}\rangle`.
- † `papersize` Breite und Höhe des Papiers. `papersize=\{\langle\text{Breite}\rangle,\langle\text{Höhe}\rangle\}` oder `papersize=\langle\text{Länge}\rangle`.
- † `landscape` wechselt die Papierorientierung in den Landschaftsmodus.
- † `portrait` wechselt die Papierorientierung in den Portraitsmodus. Dies ist das Gleiche wie `landscape=false`.

Die Optionen für den Papiernamen (z. B., `a4paper`) und die Orientierung (`portrait` und `landscape`) können als Dokumentklassenoptionen benutzt werden. Zum Beispiel können sie `\documentclass[a4paper,landscape]{article}` eingeben, dann werden `a4paper` und `landscape` auch in geometry abgearbeitet. Dies ist ebenso im Fall von `twoside` (zweiseitig) und `twocolumn` (zweispaltig) (Lesen sie auch Kapitel 5.5).

5.2 Layoutgröße

Sie können den Layoutbereich mit den Optionen, die in diesem Kapitel beschrieben sind, ungeachtet der Papiergröße, angeben. Die Optionen helfen das angegebene Layout auf unterschiedlichen Papiergrößen zu drucken. Wenn sie beispielsweise `a4paper` und `layout=a5paper` nutzen, dann benutzt das Paket das 'A5' Layout, um die Ränder auf dem 'A4' Papier zu berechnen. Die Layoutgröße wird genauso groß wie die des Papiers. Die Optionen für die Layoutgröße sind mittels `\newgeometry` erreichbar, sodass sie die Layoutgröße in der Mitte des Dokuments verändern können. Die Papiergröße selbst kann so nicht geändert werden. Abbildung 4 zeigt, was die Unterschiede zwischen `layout` und `paper` sind.

- `layout` gibt die Layoutgröße vom Papiernamen an. `layout=<Papiername>`. Alle Papiernamen, die in `geometry` definiert sind, sind wählbar. Lesen sie auch Kapitel 5.1 für mehr Details.
- `layoutwidth` Breite des Layouts. `layoutwidth=<Länge>`.
- `layoutheight` Höhe des Layouts. `layoutheight=<Länge>`.
- `layoutsizes` Breite und Höhe des Layouts. `layoutsizes={<Breite>, <Höhe>}` oder `layoutsizes=<Länge>`.
- `layouthoffset` gibt den horizontalen Versatz von der linken Kante des Papiers an. `layouthoffset=<Länge>`.
- `layoutvoffset` gibt den vertikalen Versatz von der oberen Kante des Papiers an. `layoutvoffset=<Länge>`.
- `layoutoffset` gibt beides, horizontalen und vertikalen Versatz an. `layoutoffset={<hoffset>, <voffset>}` oder `layoutsizes=<Länge>`.

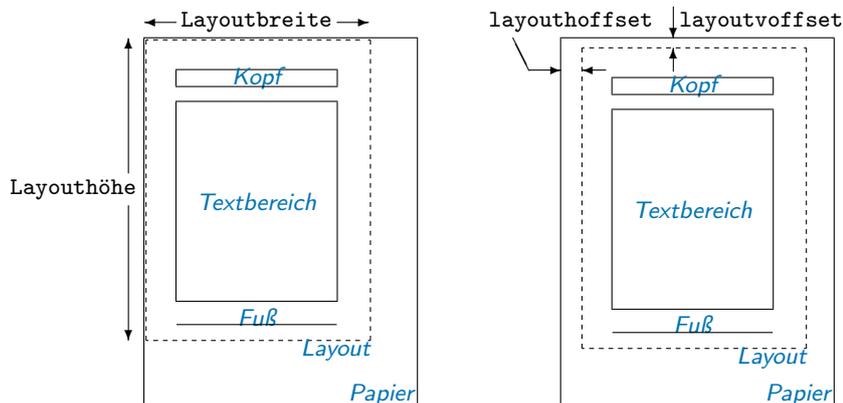


Abbildung 4: Die Maße sind mit denen der Layoutgröße verwandt. Beachte, dass die Layoutgrößenvorgaben zu der Größe des Papiers korrespondieren, sodass sie die meisten layoutbezogenen Optionen nicht explizit angeben müssen.

5.3 Größe des Textbereichs

Die Optionen, die die Größe des *druckbaren Bereichs* angeben, werden in diesem Kapitel beschrieben.

- `hscale` Verhältnis der Breite des *druckbaren Bereichs* zur `\paperwidth` (Papierbreite). `hscale=<h-scale>`, z. B. `hscale=0.8` ist äquivalent zu `width=0.8\paperwidth` (Papierbreite). (0.7 ist der Standard)
- `vscale` Verhältnis der Höhe des *druckbaren Bereichs* zur `\paperheight` (Papierhöhe), z. B. `vscale=<v-scale>`. (0.7 ist der Standard) `vscale=0.9` ist äquivalent zu `height=0.9\paperheight` (Papierhöhe).
- `scale` Verhältnis des *druckbaren Bereichs* zum Papier. `scale={<h-scale>, <v-scale>}` oder `scale=<scale>`. (0.7 ist der Standard)
- `width` | `totalwidth` Breite des *druckbaren Bereichs*. `width=<Länge>` oder `totalwidth=<Länge>`. Dieses Maß korrespondiert zur `textwidth` (Textbreite), aber wenn `includemp` auf `true` gesetzt ist, dann ist die Breite \geq Textbreite, weil die Breite die Breite der Randnotizen beinhaltet. Wenn die Textbreite und die Breite gleichzeitig angegeben werden, dann hat die Textbreite eine höhere Priorität als die Breite.

<code>height</code> <code>totalheight</code>	Höhe des <i>druckbaren Bereichs</i> , wobei Kopf und Fuß standardmäßig ausgeschlossen sind. Wenn <code>includehead</code> oder <code>includefoot</code> gesetzt ist, dann beinhaltet die <code>height</code> (Höhe) den Kopf oder den Fuß der Seite sowie die <code>textheight</code> (Texthöhe). <code>height=<Länge></code> oder <code>totalheight=<Länge></code> . Falls beide, <code>textheight</code> (Texthöhe) und <code>height</code> (Höhe), angegeben sind, wird die <code>height</code> (Höhe) ignoriert werden.
<code>total</code>	Breite und Höhe des <i>druckbaren Bereichs</i> . <code>total={<Breite>, <Höhe>}</code> oder <code>total=<Länge></code> .
<code>textwidth</code>	gibt die <code>\textwidth</code> (Textbreite), die Breite des <i>Körpers</i> (der Textbereich) an. <code>textwidth=<Länge></code> .
<code>textheight</code>	gibt die <code>\textheight</code> (Texthöhe), die Höhe des <i>Körpers</i> (der Textkörper) an. <code>textheight=<Länge></code> .
<code>text</code> <code>body</code>	gibt beides, die <code>\textwidth</code> (Textbreite) und <code>\textheight</code> (Texthöhe) des Körpers der Seite an. <code>body={<Breite>, <Höhe>}</code> oder <code>text=<Länge></code> .
<code>lines</code>	erlaubt dem Benutzer, die <code>\textheight</code> (Texthöhe) durch die Anzahl der Zeilen anzugeben. <code>lines=<ganzzahliger Wert></code> .
<code>includehead</code>	schließt den Kopf der Seite, die <code>\headheight</code> (Kopfhöhe) und <code>\headsep</code> (Kopftrennung) in den <i>druckbaren Bereich</i> ein. Diese Option ist in den Voreinstellungen durch <code>false</code> deaktiviert. Es ist das Gegenteil von <code>ignorehead</code> . Betrachten sie auch Abbildung 2 und Abbildung 5 .
<code>includefoot</code>	schließt den Fuß der Seite, <code>\footskip</code> (Fußbereich), in den <i>druckbaren Bereich</i> ein. Es ist das Gegenteil von <code>ignorefoot</code> . Die Voreinstellung ist <code>false</code> (deaktiviert). Betrachten sie auch Abbildung 2 und Abbildung 5 .
<code>includeheadfoot</code>	setzt beides, <code>includehead</code> und <code>includefoot</code> , auf <code>true</code> , dies ist das Gegenteil von <code>ignoreheadfoot</code> . Betrachten sie auch Abbildung 2 und Abbildung 5 .
<code>includemp</code>	schließt die Randnotizen, die <code>\marginparwidth</code> (Randnotizbreite) und <code>\marginparsep</code> (Randnotiztrennung) in den <i>Körper</i> ein, wenn die horizontalen Maße berechnet werden.
<code>includeall</code>	setzt beides, <code>includeheadfoot</code> und <code>includemp</code> auf <code>true</code> (aktiv). Betrachten sie auch Abbildung 5 .
<code>ignorehead</code>	missachtet den Kopf der Seite, <code>headheight</code> (Kopfhöhe) und <code>headsep</code> (Kopftrennung), sind ausschlaggebend für das vertikale Layout, aber diese ändern dessen Längen nichts. Es ist äquivalent zu <code>includehead=false</code> . Diese Option ist mit <code>true</code> voreingestellt. Lesen sie auch <code>includehead</code> .
<code>ignorefoot</code>	missachtet den Fuß der Seite, der <code>footskip</code> (Fußbereich), ist ausschlaggebend für das vertikale Layout, aber dies ändert die Länge nicht. Diese Voreinstellung der Option ist <code>true</code> . Lesen sie auch <code>includefoot</code> .
<code>ignoreheadfoot</code>	setzt beides, <code>ignorehead</code> und <code>ignorefoot</code> auf <code>true</code> . Lesen sie auch <code>includeheadfoot</code> .
<code>ignoremp</code>	vernachlässigt die Randnotizen bei der Berechnung der horizontalen Ränder (der Standard ist <code>true</code>). Wenn die Randnotizen die Seite überfüllen, dann wird eine Warnmeldung angezeigt werden, wenn <code>verbose=true</code> aktiviert ist. Sehen sie auch <code>includemp</code> und Abbildung 5 .
<code>ignoreall</code>	setzt beides, <code>ignoreheadfoot</code> und <code>ignoremp</code> auf <code>true</code> . Lesen sie auch <code>includeall</code> .
<code>heightrounded</code>	Diese Option rundet die <code>\textheight</code> (Texthöhe), um n -Mal mehr (n : ist eine ganze Zahl) <code>\baselineskip</code> plus <code>\topskip</code> , um "underfull vbox" in manchen Fällen zu vermeiden. Zum Beispiel, wenn die <code>\textheight</code> (Texthöhe) 486pt ist, mit <code>\baselineskip 12pt</code> und <code>\topskip 10pt</code> , dann <p style="text-align: center;">$(39 \times 12pt + 10pt =) 478pt < 486pt < 490pt (= 40 \times 12pt + 10pt),$</p> als Ergebnis wird die <code>\textheight</code> (Texthöhe) auf 490 pt aufgerundet. <code>heightrounded=false</code> ist voreingestellt.

[Abbildung 5](#) illustriert die verschiedenen Layouttypen mit den verschiedenen Layoutmodi. Die Maße für einen Kopf und einen Fuß können durch den `nohead`- oder `nofoot`-Modus kontrolliert werden, dies

setzt jede Länge auf Opt. Auf der anderen Seite verändern Optionen mit dem Präfix *ignore nicht* die korrespondierenden, ursprünglichen Maße.

Die folgenden Optionen können den Körper und die Ränder gleichzeitig angeben, mit drei kommagetrennten Werten in geschweiften Klammern.

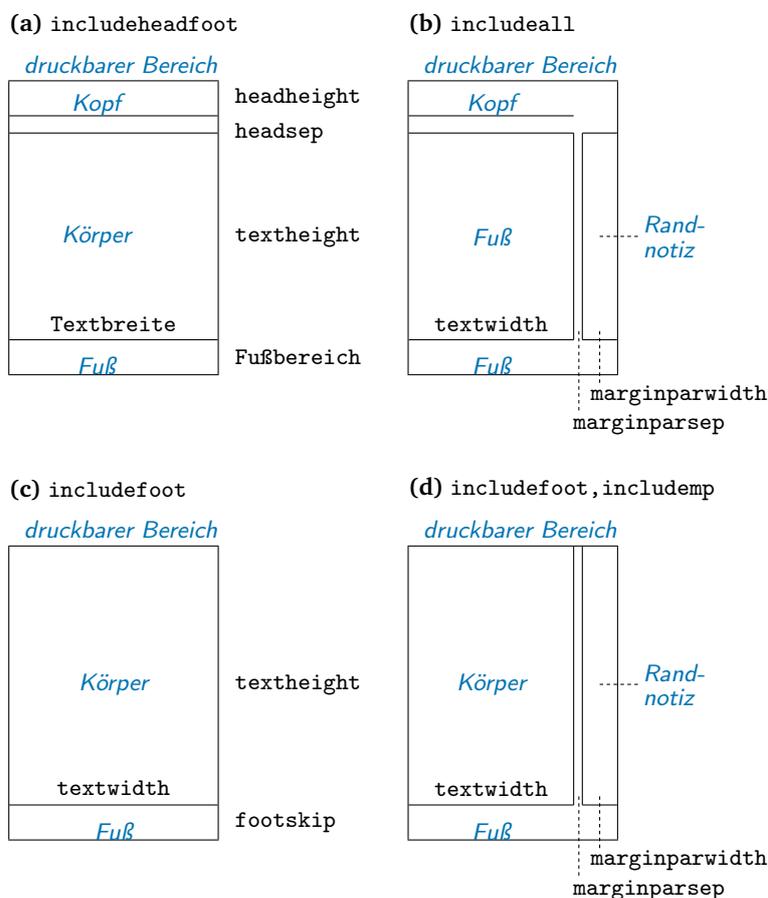


Abbildung 5: Musterlayouts für den *druckbaren Bereich* mit verschiedenen Veränderungen. (a) `includeheadfoot`, (b) `includeall`, (c) `includefoot` und (d) `includefoot,includemp`. Wenn `reversemp` auf `true` gesetzt ist, dann werden die Orte für die Randnotizen auf jeder Seite getauscht. Die Option `twoside` tauscht beides, die Ränder und die Randnotizen auf der Rückseite. Beachten sie, dass die Randnotiz gedruckt wird, falls vorhanden, trotz das `ignoremp` oder `includemp=false` und Überlauf der Seite, in einigen Fällen.

`hdivide` horizontale Aufteilung (links,Breite,rechts). `hdivide={⟨linker Rand⟩,⟨Breite⟩,⟨rechter Rand⟩}`. Beachten sie, dass sie nicht alle drei Parameter eingeben sollten. Die beste Möglichkeit der Nutzung dieser Option ist die Angabe von zwei der drei Parameter und das Freilassen des verbliebenen mit Null(nichts) oder `*`. Zum Beispiel, wenn sie `hdivide={2cm,15cm, }` einstellen, dann wird der Rand von der rechtsseitigen Kante der Seite die Rechnung bestimmen `paperwidth-2cm-15cm`.

`vdivide` vertikale Aufteilung (oben,Höhe,unten). `vdivide={⟨oberer Rand⟩,⟨Höhe⟩,⟨unterer Rand⟩}`.

`divide` `divide={A,B,C}` wird interpretiert als `hdivide={A,B,C}` und `vdivide={A,B,C}`.

5.4 Randgröße

Die Optionen zum Spezifizieren der Größe der Ränder sind unten aufgelistet.

`left` | `lmargin` | `inner`

linker Rand (für einseitige Dokumente) oder innerer Rand (für zweiseitige Dokumente) des *druckbaren Bereichs*. Mit anderen Worten, es ist die Distanz zwischen der linken (inneren) Kante des Papiers und der des *druckbaren Bereichs*. `left=⟨Länge⟩`.

	inner (innen) hat keine besondere Bedeutung, nur eine alternative für left und lmargin.
right rmargin outer	rechter oder äußerer Rand des <i>druckbaren Bereichs</i> . right= <i>Länge</i> .
top tmargin	oberer Rand der Seite. top= <i>Länge</i> . Beachten sie, dass diese Option nichts mit dem ursprünglichen Maß \topmargin zu tun hat.
bottom bmargin	unterer Rand der Seite. bottom= <i>Länge</i> .
hmargin	linker und rechter Rand. hmargin={ <i>linker Rand</i> , <i>rechter Rand</i> } oder hmargin= <i>Länge</i> .
vmargin	oberer und unterer Rand. vmargin={ <i>oberer Rand</i> , <i>unterer Rand</i> } oder vmargin= <i>Länge</i> .
margin	margin={ <i>A</i> , <i>B</i> } ist äquivalent zu hmargin={ <i>A</i> , <i>B</i> } und vmargin={ <i>A</i> , <i>B</i> }. margin= <i>A</i> wird automatisch erweitert zu hmargin= <i>A</i> und vmargin= <i>A</i> .
hmarginratio	Verhältnis des horizontalen Rands des linken (inneren) zum rechten (äußeren). Der Wert des <i>Verhältnisses</i> sollte mit zwei Werten, die durch einen Doppelpunkt getrennt sind, angegeben werden. Jeder Wert sollte eine positive ganze Zahl kleiner 100 sein, um einen arithmetischen Überlauf zu verhindern, z. B. 2:3 anstatt 1:1.5. Das Standardverhältnis ist 1:1 für einseitige Dokumente, 2:3 für zweiseitige.
vmarginratio	Verhältnis des vertikalen Randes, des oberen Randes zum unteren. Das Standardverhältnis ist 2:3.
marginratio ratio	horizontale und vertikale Randverhältnisse. marginratio={ <i>horizontales Verhältnis</i> , <i>vertikales Verhältnis</i> } oder marginratio= <i>Verhältnis</i> .
hcentering	aktiviert die automatische horizontale Zentrierung und ist äquivalent zu hmarginratio=1:1. Es ist für die Option oneside auf true gesetzt. Lesen sie auch hmarginratio.
vcentering	aktiviert die automatische vertikale Zentrierung und ist äquivalent zu vmarginratio=1:1. Diese Option ist standardmäßig auf false gesetzt. Lesen sie auch vmarginratio.
centering	aktiviert die automatische Zentrierung und ist äquivalent zu marginratio=1:1. Lesen sie auch marginratio. Der Standard ist false.
twoside	wechselt das Layout in den twoside (zweiseitigen) Modus, wobei der linke und der rechte Rand auf der Rückseite vertauscht wird. Die Option setzt einen \@twoside und \@mparswitch Schalter. Lesen sie auch asymmetric.
asymmetric	führt ein zweiseitiges Layout ein, wobei die Ränder nicht getauscht werden auf alternativen Seiten (durch die Einstellung \oddsidemargin zu \evensidemargin + bindingoffset) und die Randnotizen immer auf der selben Seite bleiben. Diese Option kann als Alternative für die twoside Option genutzt werden. Lesen sie auch twoside.
bindingoffset	entfernt einen angegeben Raum von der linken Seite bei oneside oder der inneren Seite bei twoside. bindingoffset= <i>Länge</i> . Dies ist nützlich, wenn die Seiten durch eine Pressbindung verbunden sind (glued, stitched, stapled ...). Betrachten sie auch Abbildung 6.
hdivide	Siehe Beschreibung im Kapitel 5.3.
vdivide	Siehe Beschreibung im Kapitel 5.3.
divide	Siehe Beschreibung im Kapitel 5.3.

5.5 Ursprüngliche Maße

Die unteren Optionen überschreiben die ursprünglichen Maße von \LaTeX und die Veränderungen für das Seitenlayout (Betrachten sie auch die rechte Seite in Abbildung 1).

headheight head	modifiziert die \headheight, die Höhe des Kopfes. headheight= <i>Länge</i> oder head= <i>Länge</i> .
headsep	modifiziert die \headsep, die Abtrennung zwischen dem Kopfbereich und dem Text (body). headsep= <i>Länge</i> .

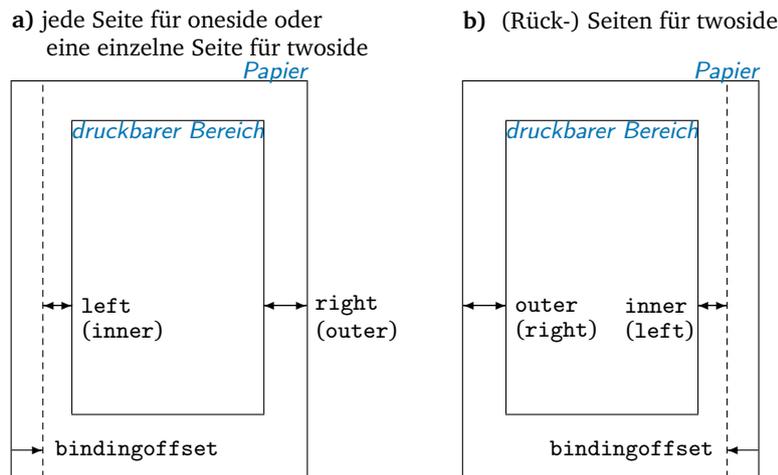


Abbildung 6: Die Option `bindingoffset` fügt eine angegebene Länge zum inneren Rand hinzu. Beachten sie, dass die `twoside` Option die horizontalen Ränder und die Randnotizen zusammen mit `bindingoffset` auf jeder Seite tauscht (siehe **b**)), aber die `asymmetric` Option unterdrücke den Tausch der Ränder und der Randnotizen (aber `bindingoffset` wird dennoch getauscht).

<code>footskip</code> <code>foot</code>	modifiziert den <code>\footskip</code> , die Distanzabtrennung zwischen der Grundlinie der letzten Zeile des Textes und der Grundlinie des Fußes. <code>footskip=<Länge></code> oder <code>foot=<Länge></code> .
<code>nohead</code>	entfernt den Platz für den Kopf der Seite, dies ist äquivalent zu <code>\headheight=0pt</code> und <code>\headsep=0pt</code> .
<code>nofoot</code>	entfernt den Platz für den Fuß der Seite, dies ist äquivalent zu <code>\footskip=0\,pt</code> .
<code>noheadfoot</code>	ist äquivalent zu <code>nohead</code> und <code>nofoot</code> , dies bedeutet, dass <code>\headheight</code> , <code>\headsep</code> und <code>\footskip</code> alle auf <code>0pt</code> eingestellt werden.
<code>footnotesep</code>	verändert die Maße <code>\skip\footins</code> , die Abtrennung zwischen dem unteren Ende des Textkörpers und dem oberen Ende des Fußnotentextes.
<code>marginparwidth</code> <code>marginpar</code>	modifiziert die <code>\marginparwidth</code> , die Breite der Randnotizen. <code>marginparwidth=<Länge></code> .
<code>marginparsep</code>	modifiziert die <code>\marginparsep</code> , die Abtrennung zwischen dem Körper und den Randnotizen. <code>marginparsep=<Länge></code> .
<code>nomarginpar</code>	schrumpft den Platz für die Randnotizen auf <code>0pt</code> , dies ist äquivalent zu <code>\marginparwidth=0pt</code> und <code>\marginparsep=0pt</code> .
<code>columnsep</code>	modifiziert die <code>\columnsep</code> , die Abtrennung zwischen zwei Spalten im <code>twocolumn</code> Modus.
<code>hoffset</code>	modifiziert <code>\hoffset</code> . <code>hoffset=<Länge></code> .
<code>voffset</code>	modifiziert <code>\voffset</code> . <code>voffset=<Länge></code> .
<code>offset</code>	horizontaler und vertikaler Versatz. <code>offset={<hoffset>, <voffset>}</code> oder <code>offset=<Länge></code> .
<code>twocolumn</code>	aktiviert den <code>twocolumn</code> Modus mit <code>\@twocolumntrue</code> . <code>twocolumn=false</code> bezeichnet den <code>onecolumn</code> Modus (Modus mit einer Spalte) mit <code>\@twocolumnfalse</code> . Anstatt <code>twocolumn=false</code> , können sie <code>onecolumn</code> angeben (die Vorgabe ist <code>true</code>)
<code>onecolumn</code>	arbeitet als <code>twocolumn=false</code> . Auf der anderen Seite ist <code>onecolumn=false</code> äquivalent zu <code>twocolumn</code> .
<code>twoside</code>	aktiviert beides, <code>\@twosidetrue</code> und <code>\@mparswitchtrue</code> . Lesen sie auch Kapitel 5.4.
<code>textwidth</code>	lässt sie die <code>\textwidth</code> (Textbreite) direkt einstellen. Lesen sie auch Kapitel 5.3.
<code>textheight</code>	lässt sie die <code>\textheight</code> (Texthöhe) direkt einstellen. Lesen sie auch Kapitel 5.3.
<code>reversemp</code> <code>reversemarginpar</code>	sorgt dafür, dass die Randnotizen am (inneren) linken Rand erscheinen, mit <code>\@reversemargintrue</code> . Die Option verändert nichts am <code>includemp</code> Modus. Diese Option ist mit <code>false</code> voreingestellt.

5.6 Treiber

Dieses Paket unterstützt die Treiber `dvips`, `dvipdfm`, `pdftex`, `luatex`, `xetex` und `vtex`. Sie können auch `dvipdfm` für `dvipdfmx` einstellen und `xdvipdfm`. `pdftex` für `pdflatex`, und `vtex` für die $\text{V}\text{T}\text{E}\text{X}$ Umgebung. Die Treiberoptionen sind exklusiv. Der Treiber kann durch beides eingestellt werden, `driver=<Treibername>` oder die direkte Angabe eines Treibers wie `pdftex`. Standardmäßig nimmt `geometry` den Treiber, der am besten zum benutzten System passt. Deshalb müssen sie in den meisten Fällen keinen Treiber angeben. Wie auch immer, wenn sie `dvipdfm` benutzen wollen, dann sollten sie dies explizit angeben.

- † `driver` gibt den Treibernamen mit `driver=<Treibername>` an. `dvips`, `dvipdfm`, `pdftex`, `vtex`, `xetex`, `auto` und `none` sind als Treibername einsetzbar. Außer den Namen für `auto` und `none` können alle Treiber direkt mit dem Namen, ohne `driver=`, angegeben werden. `driver=auto` schaltet die automatische Detektion ein, egal was die vorherigen Einstellungen waren. `driver=none` schaltet die automatische Detektion ab und es wird kein Treiber gewählt. Dies könnte nützlich sein, wenn sie wollen, dass andere Pakete die Treibereinstellungen herausfinden sollen. Ein Beispiel: Wenn sie das `crop` Paket mit `geometry` nutzen wollen, dann sollten sie `\usepackage[driver=none]{geometry}` aufrufen, bevor sie das `crop` Paket aufrufen.
- † `dvips` schreibt die Papiergröße in die dvi Ausgabe, mit dem `\special` Makro. Wenn sie `dvips` als DVI-zu-PS Treiber nutzen, dann brauchen sie, wenn sie beispielsweise ein Dokument mit `\geometry{a3paper,landscape}` auf A3-Papier mit der `landscape` Orientierung drucken wollen, die Optionen `“-t a3 -t landscape”` für `dvips` nicht.
- † `dvipdfm` arbeitet wie `dvips`, ausgenommen die `landscape` Korrektur. Sie können diese Option einschalten, wenn sie `dvipdfmx` nutzen und `xdvipdfm`, um die dvi Ausgabe zu erzeugen.
- † `pdftex` setzt `\pdfpagewidth` und `\pdfpageheight` intern.
- † `xetex` ist dasselbe wie `pdftex`, ausgenommen das Ignorieren von `\pdf{h,v}origin`, das in $\text{X}\text{E}\text{L}\text{A}\text{T}\text{E}\text{X}$ nicht definiert ist. Diese Option wurde in 5. Version eingeführt. Beachten sie, dass die `‘geometry.cfg’` in TEX Live, welche die automatische Detektionsroutine deaktiviert und `pdftex` einstellt, nicht länger notwendig ist. Trotzdem gibt es keine Probleme, sogar wenn die Datei nicht gelöscht wird. Anstatt `xetex`, können sie `dvipdfm` angeben, mit $\text{X}\text{E}\text{L}\text{A}\text{T}\text{E}\text{X}$, wenn sie die Besonderheiten von `dvipdfm` nutzen wollen, die $\text{X}\text{E}\text{L}\text{A}\text{T}\text{E}\text{X}$ unterstützt.
- † `vtex` setzt die Maße `\mediawidth` und `\mediaheight` für $\text{V}\text{T}\text{E}\text{X}$. Wenn dieser Treiber ausgewählt ist (explizit oder automatisch), wird `geometry` den Ausgabemodus (DVI, PDF oder PS), der in $\text{V}\text{T}\text{E}\text{X}$ gewählt ist, selbstständig herausfinden und die richtigen Einstellungen für ihn machen.

Wenn der explizit angegebene Treiber nicht zum benutzten Textsatzprogramm passt, dann wird der Standardtreiber `dvips` ausgewählt.

5.7 Sonstige Optionen

Die sonstigen nützlichen Optionen werden jetzt beschrieben.

- † `verbose` zeigt die Parameterergebnisse im Terminal an. `verbose=false` (Standard) schreibt diese nur in die Log-Datei.
- † `reset` setzt die Layoutmaße zurück und nimmt die Einstellungen an, die vor dem Laden von `geometry` vorgegeben waren. Die Optionen, die in `geometry.cfg` angegeben sind, werden auch entfernt. Beachten sie, dass dies nicht pass und mag mit `truedimen` zurücksetzen kann. `reset=false` hat keine Auswirkungen und kann ein vorheriges `reset(=true)`, falls es dies gibt, nicht aufheben. Zum Beispiel, wenn sie

```
\documentclass[landscape]{article}
\usepackage[twoside,reset,left=2cm]{geometry}
```

eingeben, mit `\ExecuteOptions{scale=0.9}` in `geometry.cfg`, dann bleiben `landscape` und `left=2cm` wirksam und `scale=0.9` und `twoside` sind unwirksam.
- † `mag` setzt den Vergrößerungswert (`\mag`) und modifiziert automatisch `\hoffset` und `\voffset` entsprechend der Vergrößerung. `mag=<Wert>`. Beachten sie, dass `<Wert>` ein ganzzahliger Wert sein sollte, wobei 1000 die Normalgröße ist. Ein Beispiel: `mag=1414` mit `a4paper` bietet eine erweiterte Druckgröße in `a3paper`, welches $1.414 (= \sqrt{2})$ mal

größer als `a4paper` ist. Die Vergrößerung der Schriftart benötigt extra Speicherplatz. **Beachten sie, dass die Einstellung über jeder anderen Einstellung mit ‘true’ Maßen stehen sollte, solchen Maßen wie `1.5truein`, `2truecm` und so weiter.** Lesen sie auch die `trueedimen` Option.

- † `trueedimen` verändert alle internen expliziten Maßwerte in *true* Maße, z. B. wird `1in` zu `1truein`. Typischerweise wird diese Option zusammen mit der `mag` Option genutzt. Beachten sie, dass dies unwirksam gegenüber extern angegebenen Maßen ist. Ein Beispiel: Wenn sie “`mag=1440, margin=10pt, trueedimen`” einstellen, dann sind die Ränder nicht ‘true’ aber vergrößert. Wenn sie exakte Ränder wollen, dann sollten sie stattdessen wie folgt definieren “`mag=1440, margin=10truept, trueedimen`”.
- † `pass` deaktiviert alle `geometry` Optionen und Berechnungen, ausgenommen `verbose` und `showframe`. Es ist unabhängig von der Reihenfolge und kann für das Überprüfen des Seitenlayouts der Dokumentklasse, sonstiger Pakete und manuellen Einstellungen ohne `geometry` genutzt werden.
- † `showframe` zeigt sichtbare Ränder um den Textbereich und die Seite und die Linien für den Kopf und den Fuß auf der ersten Seite.
- † `showcrop` druckt Schnittmarken auf jede Ecke des benutzerdefinierten Layoutbereichs.

6 Verarbeitungsoptionen

6.1 Reihenfolge des Ladens

Wenn es eine `geometry.cfg` gibt, irgendwo wo es \TeX finden kann, dann lädt `geometry` diese zuerst. Ein Beispiel: In `geometry.cfg` können sie `\ExecuteOptions{a4paper}` schreiben, welches A4-Format als das Standardpapier angibt. Grundlegend können sie alle Optionen, die in `geometry` definiert sind, mit `\ExecuteOptions{}` nutzen.

Die Reihenfolge des Ladens in der Präambel ihres Dokuments ist wie folgt:

1. `geometry.cfg`, wenn sie existiert.
2. Optionen, die mit `\documentclass[(Optionen)]{...}` angegeben werden.
3. Optionen, die mit `\usepackage[(Optionen)]{geometry}` angegeben werden.
4. Optionen, die mit `\geometry{(Optionen)}` angegeben werden. Die jeweilige Option kann mehrfach aufgerufen werden. (Die `reset` Option wird alle Optionen, die in `\usepackage{geometry}` oder `\geometry` angegeben wurden, zurücksetzen.)

6.2 Reihenfolge der Optionen

Die Spezifikation der `geometry` Optionen ist unabhängig von der Reihenfolge und überschreibt die vorherigen gleichartigen Einstellungen. Ein Beispiel:

`[left=2cm, right=3cm]` ist äquivalent zu `[right=3cm, left=2cm]`.

Der mehrfache Aufruf der Optionen überschreibt die jeweiligen vorherigen Einstellungen. Auch hier ein Beispiel:

`[verbose=true, verbose=false]` das Ergebnis ist `verbose=false`.

`[hmargin={3cm,2cm}, left=1cm]` ist das Gleiche wie `hmargin={1cm,2cm}`, wobei der linke (oder innere) Rand durch `left=1cm` überschrieben wird.

`reset` und `mag` sind Ausnahmen. Die `reset` Option entfernt alle `geometry` Optionen (ausgenommen `pass`) davor. Wenn sie

```
\documentclass[landscape]{article}
\usepackage[margin=1cm,twoside]{geometry}
\geometry{a5paper, reset, left=2cm}
```

einstellen, dann werden `margin=1cm` (Rand), `twoside` und `a5paper` entfernt und ist gelegentlich äquivalent zu

```
\documentclass[landscape]{article}
\usepackage[left=2cm]{geometry}
```

Die `mag` Option sollte vor jeder anderen Einstellungen mit ‘true’ Länge, solche wie `left=1.5truecm`, `width=5truein` und so weiter, stehen. Die `\mag` Option kann vor dem Aufruf dieses Pakets stehen.

6.3 Priorität

Es gibt verschiedene Wege die Maße des *Körpers* einzustellen: `scale`, `total`, `text` und `lines`. Das `geometry` Paket gibt den konkreteren Spezifikationen eine höhere Priorität. Hier ist die Prioritätsregel für den *Textkörper*.

$$\begin{array}{c} \text{Priorität:} \quad \text{niedrig} \quad \longrightarrow \quad \text{hoch} \\ \left\{ \begin{array}{l} \text{hscale} \\ \text{vscale} \\ \text{scale} \end{array} \right\} < \left\{ \begin{array}{l} \text{width} \\ \text{height} \\ \text{total} \end{array} \right\} < \left\{ \begin{array}{l} \text{textwidth} \\ \text{textheight} \\ \text{text} \end{array} \right\} < \text{lines.} \end{array}$$

Ein Beispiel:

```
\usepackage[hscale=0.8, textwidth=7in, width=18cm]{geometry}
```

ist das Gleiche wie `\usepackage[textwidth=7in]{geometry}`. Ein weiteres Beispiel:

```
\usepackage[lines=30, scale=0.8, text=7in]{geometry}
```

resultiert in `[lines=30, textwidth=7in]`.

6.4 Standards

Dieses Kapitel summiert die Standardeinstellungen für die automatische Vervollständigung, die später beschrieben wird.

Das Standardverhältnis für den vertikalen Rand ist $2/3$, namentlich,

$$\text{top} : \text{bottom} = 2 : 3 \quad \textit{Standard.} \quad (6)$$

Was den horizontalen Rand betrifft, hängen die Standardwerte davon ab, ob das Dokument einseitig oder zweiseitig ist,

$$\text{left (inner)} : \text{right (outer)} = \begin{cases} 1 : 1 & \textit{Standard für oneside,} \\ 2 : 3 & \textit{Standard für twoside.} \end{cases} \quad (7)$$

Offensichtlich ist, dass der Standard für den horizontalen Rand für einseitig (*oneside*) ‘centering’ ist.

Das `geometry` Paket besitzt folgende Standardeinstellungen für *onesided* (*einseitige*) Dokumente:

- `scale=0.7` (*Textkörper* ist $0.7 \times$ *Papier*)
- `marginratio={1:1, 2:3}` (1:1 für horizontal und 2:3 für vertikale Ränder)
- `ignoreall` (der Kopf, der Fuß, die Randnotizen sind ausgeschlossen, wenn die Größe des *Körpers* (*body*) berechnet wird.)

Für *twosided* (*zweiseitige*) Dokumente mit der `twoside` Option, ist die Standardeinstellung die gleiche, wie bei *onesided* (*einseitigen*), ausgenommen, dass das horizontale Randverhältnis auf 2:3 gesetzt wird.

Zusätzliche Optionen überschreiben die vorher angegebenen Maße.

6.5 Autovervollständigung

Abbildung 7 zeigt schematisch, wie viele Spezifikationsmuster existieren und wie sie die Zweideutigkeiten der Spezifikationen lösen können. Jede Achse zeigt die Anzahl der Längen, die explizit für den Körper und die Ränder angegeben werden müssen. $S(m,b)$ zeigt die Spezifikation mit einer Menge an Zahlen (*Rand, Körper*) = (m, b) .

Ein Beispiel: Die Spezifikation `width=14cm, left=3cm` ist kategorisiert in $S(1,1)$, welches eine adäquate Spezifikation ist. Wenn sie `right=4cm` hinzufügen, dann wird es $S(2,1)$ sein und überspezifiziert. Wenn nur `width=14cm` gegeben ist, dann ist es in $S(0,1)$ unterspezifiziert.

Das `geometry` Paket besitzt einen Autovervollständigungsmechanismus, der, wenn die Layoutparameter unter- oder überspezifiziert sind, die Unstimmigkeiten herausarbeitet und dann die Standards nutzt und andere Verhältnisse. Hier sind die Spezifikationen und die Vervollständigungsregeln.

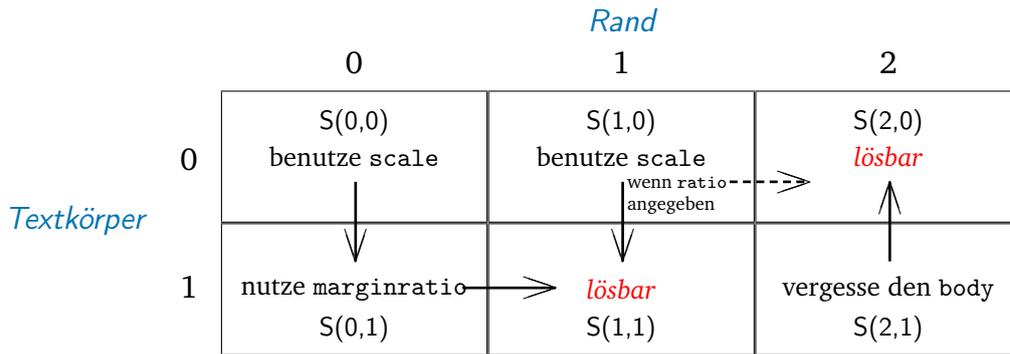


Abbildung 7: Die Spezifikationen S(0,0) zu S(2,1) und die Komplettierungsregeln (Pfeile). Spalten- und Zeilennummern bezeichnen die Anzahl der explizit angegebenen Längen für den Rand und den Körper. S(m,b) bezeichnet eine Spezifikation mit einer Menge an Zahlen (*Rand, Körper*) = (m, b).

S(0,0) Es ist nichts angegeben. Das geometry Paket gibt für den *body (Körper)* den Standard scale (= 0.7) vor.
Ein Beispiel: width (Breite) wird auf $0.7 \times \text{layoutwidth}$ eingestellt. Beachten sie, dass standardmäßig die layoutwidth und layoutheight (Layoutbreite und -höhe) gleich der \paperwidth und \paperheight (Papierbreite und -höhe) sein werden. Demnach wird S(0,0) zu S(0,1). Siehe auch S(0,1).

S(0,1) Nur der *body* ist angegeben, wie z. B. width=7in, lines=20, body={20cm, 24cm}, scale=0.9 und so weiter. Dann setzt geometry die Ränder mithilfe des Randverhältnisses. Falls kein Verhältnis angegeben ist, wird das Standardverhältnis genutzt. Das Standardverhältnis für das vertikale Randverhältnis ist wie folgt definiert:

$$\text{top} : \text{bottom} = 2 : 3 \quad \textit{Standard}. \quad (8)$$

Was das horizontale Verhältnis betrifft, hängt das Standardverhältnis davon ab, ob das Dokument einseitig oder zweiseitig ist.

$$\text{left (inner)} : \text{right (outer)} = \begin{cases} 1 : 1 & \textit{Der Standard bei einseitig,} \\ 2 : 3 & \textit{Der Standard bei zweiseitig.} \end{cases} \quad (9)$$

Wenn zum Beispiel auf A4-Papier die height=22cm angegeben ist, dann berechnet geometry den top (oberen) Rand wie folgt:

$$\begin{aligned} \text{top} &= (\text{layoutheight} - \text{height}) \times 2/5 \\ &= (29.7 - 22) \times 2/5 = 3.08(\text{cm}) \end{aligned} \quad (10)$$

Demnach sind der top (obere) Rand und die height (Höhe) des Körpers bestimmt wurden, die Spezifikation für die Vertikale wird also zu S(1,1) und alle Parameter können gelöst werden.

S(1,0) Nur ein Rand ist angegeben, wie bottom=2cm, left=1in, top=3cm und so weiter.

- Wenn das Randverhältnis *nicht* angegeben ist, dann nimmt geometry den Standard für scale (= 0.7) für den *body* an. Wenn beispielsweise top=2.4cm angegeben ist, dann setzt geometry

$$\text{height} = 0.7 \times \text{layoutheight} \quad (= 0.7 \backslash \text{paperheight by default}),$$

dann wird S(1,0) zu S(1,1), wobei der bottom mit $\text{layoutheight} - (\text{height} + \text{top})$ berechnet wird und in 6.51 cm auf A4-Papier resultiert, wenn die Layoutgröße gleich der Papiergröße ist.

- **Wenn das Randverhältnis angegeben ist**, wie `hmarginratio={1:2}`, `vratio={3:4}` und so weiter, dann setzt `geometry` die anderen Ränder mit dem gegebenen Randverhältnis. Falls z. B. eine Menge an Optionen “`top=2.4cm, vratio={3:4}`” spezifiziert wurde, dann setzt `geometry` den `bottom` auf den berechneten Wert von 3.2cm

$$\text{bottom} = \text{top}/3 \times 4 = 3.2\text{cm}$$

Demnach wird S(1,0) zu S(2,0).

Beachten sie, dass in der Version 4 oder früher das Berechnungsverfahren ein anderes war. Deshalb wird, wenn sie in Version 5 und 4 die gleichen Werte eingeben, das Ergebnis nicht gleich sein. Wenn nur `top=2.4cm` gegeben ist, dann bekommen sie in Version 4 oder früher `bottom=2.4cm`, aber sie werden `bottom=6.51cm` bei Version 5 erhalten.

S(2,1) Der *body (Körper)* und zwei *margins (Ränder)* sind angegeben, wie z. B. `vdivide={1in,8in,1.5in}`, “`left=3cm,width=13cm,right=4cm`” und so weiter. Seit `geometry` grundsätzlich die Priorität an die *margins* gibt, wenn die Maße überspezifiziert sind, vergisst `geometry` den *body* und setzt diesen zurück. Ein Beispiel: Wenn sie

```
\usepackage[a4paper,left=3cm,width=13cm,right=4cm]{geometry},
```

angeben, dann wird die `width` (Breite) auf 14 cm zurückgesetzt, da die Breite von A4-Papier 21 cm ist.

7 Verändern des Layouts innerhalb des Dokuments

Die Version 5 bietet die neuen Befehle `\newgeometry{...}` und `\restoregeometry` an, welche es ihnen erlauben, das Seitenlayout im Dokument zu verändern. Nicht wie `geometry` in der Präambel, ist `\newgeometry` nur nach `\begin{document}` verfügbar. Es setzt alle Optionen zurück, ausgenommen die, die sich auf die Papiergröße beziehen – `landscape`, `portrait` – und Papiergrößenoptionen (wie `papersize`, `paper=a4paper` und so weiter), die nicht durch `\newgeometry` geändert werden können.

Der Befehl `\restoregeometry` sichert das in der Präambel (vor `\begin{document}`) angegebene Seitenlayout, welches mit `\usepackage{geometry}`, `geometry` und deren Optionen erzeugt wurde.

Beachten sie, dass beide, `\newgeometry` und `\restoregeometry`, ein `\clearpage` einfügen, da wo sie aufgerufen werden.

Es folgt ein Beispiel für das Verändern des Layouts im Dokument. Das Layout L1 wird wie folgt angegeben: `hmargin=3cm` (left (linker) und right (rechter) Rand sind 3cm lang), dies wird in L2 in `left=3cm, right=1cm` und `bottom=0.1cm` geändert. Das Layout L1 wird mit `\restoregeometry` gesichert.

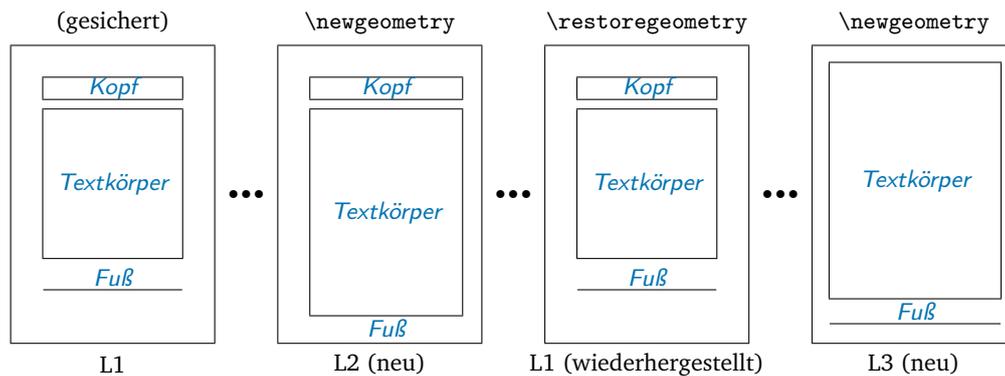
```
\usepackage[hmargin=3cm]{geometry}
\begin{document}
  Layout L1

\newgeometry{left=3cm,right=1cm,bottom=0.1cm}
  Layout L2 (neu)

\restoregeometry
  Layout L1 (wiederhergestellt)

\newgeometry{margin=1cm,includefoot}
  Layout L3 (neu)

\end{document}
```



Eine Menge von Befehlen `\savegeometry{<Name>}` und `\loadgeometry{<Name>}` ist nützlich, wenn sie mehrere verschiedene Layouts in ihrem Dokument nutzen wollen. Ein Beispiel:

```
\usepackage[hmargin=3cm]{geometry}
\begin{document}
  L1
  \newgeometry{left=3cm,right=1cm,bottom=0.1cm}
  \savegeometry{L2}
  L2 (neu, gespeichert)
  \restoregeometry
  L1 (wiederhergestellt)
  \newgeometry{margin=1cm,includefoot}
  L3 (neu)
  \loadgeometry{L2}
  L2 (geladen)
\end{document}
```

8 Beispiele

- Ein einseitiges Seitenlayout mit einem zentrierten Textbereich auf dem Papier:
Das sich anschließende Beispiel hat das gleiche Ergebnis, weil das horizontale Randverhältnis auf 1:1 gesetzt ist, wie es bei einseitigen Dokumenten der Standard ist.
 - `centering`
 - `marginratio=1:1`
 - `vcentering`
- Ein zweiseitiges Layout mit einem Versatz von 1 cm für das Binden auf der Innenseite:
 - `twoside, bindingoffset=1cm`

In diesem Fall ist die `textwidth` (Textbreite) kürzer als der Standard bei zweiseitigen Dokumenten $0.7 \times 1\text{cm} (= 0.7\text{cm})$, weil die Standardbreite des *body (Körpers)* mit `scale=0.7` eingestellt ist, was bedeutet, `width = 0.7 \times layoutwidth (= 0.7 \backslash paperwidth` standardmäßig).
- Ein Layout mit linkem, rechtem und oberem Rand (3 cm, 2 cm und 2.5 in), mit einer Texthöhe von 40 Zeilen und der Einbeziehung des Kopfes und des Fußes auf der Seite in den *druckbaren Bereich*: Die zwei Beispiele haben das gleiche Ergebnis.
 - `left=3cm, right=2cm, lines=40, top=2.5in, includeheadfoot`
 - `hmargin={3cm,2cm}, tmargin=2.5in, lines=40, includeheadfoot`
- Ein Layout mit einer Höhe des *druckbaren Bereichs* von 10 in, einem unteren Rand von 2 cm und der Standardbreite:
Der obere Rand wird automatisch berechnet. Jede der unteren Lösungen erzeugt das gleiche Layout.
 - `vdivide={*, 10in, 2cm}`

- `bmargin=2cm, height=10in`
- `bottom=2cm, textheight=10in`

Beachten sie, dass die Maße für *Kopf* und *Fuß* von der Höhe des *druckbaren Bereichs* ausgeschlossen sind. Ein zusätzliches `includefoot` sorgt dafür, dass der `\footskip` (Fußbereich) in die `totalheight` (Gesamthöhe) einbezogen wird. Deshalb ist in den unteren zwei Fällen die `textheight` (Texthöhe) im vorhergehenden Layout kürzer als in letzterem (mit exakt 10 in) über dem `\footskip`. In anderen Worten, `height = textheight + footskip` falls `includefoot=true` ist, in diesem Falle.

- `bmargin=2cm, height=10in, includefoot`
- `bottom=2cm, textheight=10in, includefoot`

5. Ein Layout mit einer `textwidth` (Textbreite) und `textheight` (Texthöhe) von 90% des Papiers und einem zentrierten *body*: Jede der unteren Lösungen erzeugt das gleiche Seitenlayout, sofern die `layoutwidth` (Layoutbreite) und die `layoutheight` (Layouthöhe) nicht geändert werden, also der Standard beibehalten wird.

- `scale=0.9, centering`
- `text={.9\paperwidth,.9\paperheight}, ratio=1:1`
- `width=.9\paperwidth, vmargin=.05\paperheight, marginratio=1:1`
- `hdivide={*,0.9\paperwidth,*}, vdivide={*,0.9\paperheight,*}` (für einseitige Dokumente)
- `margin={0.05\paperwidth,0.05\paperheight}`

Sie können ein `heightrounded` hinzufügen, um eine “underfull vbox warning” (Warnmeldung) wie

```
Underfull \vbox (badness 10000) has occurred while \output is active.
```

zu vermeiden. Lesen sie auch Kapitel 5.3 für eine detaillierte Beschreibung zu `heightrounded`.

6. Ein Layout, bei dem die Breite für die Randnotizen auf 3 cm eingestellt ist und in die Breite des *druckbaren Bereichs* einbezogen ist:

Die folgenden Beispiele erzeugen das Gleiche.

- `marginparwidth=3cm, includemp`
- `marginpar=3cm, ignoremp=false`

7. Ein Layout, bei dem der *body* das ganze Papier (A5-Papier im landscape Modus) einnimmt:

Die folgenden Beispiele sind gleich.

- `a5paper, landscape, scale=1.0`
- `landscape=TRUE, paper=a5paper, margin=0pt`

8. Ein Bildschirmgrößenlayout, verwendbar für Präsentationen mit dem PC oder dem Beamer:

```
\documentclass{slide}
\usepackage[screen,margin=0.8in]{geometry}
...
\begin{slide}
...
\end{slide}
```

9. Ein Layout mit vergrößerter Schrift und Zwischenräumen, von A4 auf A3:

Im unteren Fall ist die sich ergebende Papiergröße A3.

- `a4paper, mag=1414.`

Wenn sie ein Layout mit doppelt so großer Schriftgröße, aber ohne das Verändern der Papiergröße wollen, dann tippen sie

- `letterpaper, mag=2000, truedimen.`

Sie können die `dvips` Option hinzufügen, das ist nützlich, um eine Vorschau der eigenen Papiergröße zu betrachten, die durch `dviout` oder `xdvi` erzeugt wird.

10. Verändern des Layouts auf der ersten Seite und Beibehalten der anderen Einstellungen vor dem Laden von `geometry`:

Benutzen sie die `pass` Option, `\newgeometry` und `\restoregeometry`.

```
\documentclass{book}
\usepackage[pass]{geometry}
% 'pass' vernachlässigt das Paketlayout,
% so wird das originale 'book' Layout gespeichert.
\begin{document}
\newgeometry{margin=1cm}% Verändert die Maße der ersten Seite.
  Seite 1
\restoregeometry % stellt das 'book' Layout wieder her.
  Seite 2 und folgende
\end{document}
```

11. Ein komplexes Seitenlayout:

```
\usepackage[a5paper, landscape, twocolumn, twoside,
  left=2cm, hmarginratio=2:1, includemp, marginparwidth=43pt,
  bottom=1cm, foot=.7cm, includefoot, textheight=11cm, heightrounded,
  columnsep=1cm, dvips, verbose]{geometry}
```

Probieren sie das Layoutsetzen selbst und überprüfen sie, wie das Ergebnis ist. :-)

9 Bekannte Probleme

- Mit `mag` \neq 1000 und `truedimen`, `paperwidth` und `paperheight` ist die Anzeige im `verbose`-Modus verschieden von der wirklichen Größe der resultierenden PDF. Die PDF ist aber korrekt.
- Mit `mag` \neq 1000, *no* `truedimen` und `hyperref` sollte `hyperref` vor `geometry` geladen werden. Andernfalls wird die PDF-Größe fehlerhaft.
- Mit dem `crop` Paket und `mag` \neq 1000 arbeitet die `center`-Option von `crop` nicht gut.

10 Danksagungen

Der Autor würdigt hiermit die hilfreichen Vorschläge und Kommentare Vieler (in alphabetischer Reihenfolge ihrer Familiennamen): Jean-Bernard Addor, Frank Bennett, Alexis Dimitriadis, Friedrich Flander, Adrian Heathcote, Stephan Hennig, Morten Høgholm, Jonathan Kew, James Kilfiger, Yusuke Kuroki, Jean-Marc Lasgouttes, Włodzimierz Macewicz, Frank Mittelbach, Eckhard Neber, Rolf Niepraschk, Hans Fr. Nordhaug, Heiko Oberdiek, Keith Reckdahl, Peter Riocreux, Will Robertson, Pablo Rodriguez, Nico Schlöemer, Perry C. Stearns, Frank Stengel, Plamen Tanovski, Petr Uher, Piet van Oostrum, Vladimir Volovich, Michael Vulis.

11 Implementierung

1 \langle *package \rangle

This package requires the following packages: keyval, ifvtex.

2 \backslash RequirePackage{keyval}%

3 \backslash RequirePackage{ifvtex}%

Internal switches are declared here.

4 \backslash newif\ifGm@verbose

5 \backslash newif\ifGm@landscape

6 \backslash newif\ifGm@swap@papersize

7 \backslash newif\ifGm@includehead

8 \backslash newif\ifGm@includefoot

9 \backslash newif\ifGm@includemp

10 \backslash newif\ifGm@hbody

11 \backslash newif\ifGm@vbody

12 \backslash newif\ifGm@heightrounded

13 \backslash newif\ifGm@showframe

14 \backslash newif\ifGm@showcrop

15 \backslash newif\ifGm@pass

16 \backslash newif\ifGm@resetpaper

17 \backslash newif\ifGm@layout

18 \backslash newif\ifGm@newgm

\backslash Gm@cnth The counters for horizontal and vertical partitioning patterns.

\backslash Gm@cntv 19 \backslash newcount\Gm@cnth

20 \backslash newcount\Gm@cntv

\backslash c@Gm@tempcnt The counter is used to set number with calc.

21 \backslash newcount\c@Gm@tempcnt

\backslash Gm@bindingoffset The binding offset for the inner margin.

22 \backslash newdimen\Gm@bindingoffset

\backslash Gm@wd@mp Correction lengths for \backslash textwidth, \backslash oddsidemargin and \backslash evensidemargin in includemp mode.

\backslash Gm@odd@mp 23 \backslash newdimen\Gm@wd@mp

\backslash Gm@even@mp 24 \backslash newdimen\Gm@odd@mp

25 \backslash newdimen\Gm@even@mp

\backslash Gm@layoutwidth The dimensions for the layout area.

\backslash Gm@layoutheight 26 \backslash newdimen\Gm@layoutwidth

\backslash Gm@layouthoffset 27 \backslash newdimen\Gm@layoutheight

\backslash Gm@layoutvoffset 28 \backslash newdimen\Gm@layouthoffset

29 \backslash newdimen\Gm@layoutvoffset

\backslash Gm@dimlist The token in which \LaTeX native dimensions can be stored.

30 \backslash newtoks\Gm@dimlist

\backslash Gm@warning The macro to print warning messages.

31 \backslash def\Gm@warning#1{\PackageWarningNoLine{geometry}{#1}}%

\backslash ifGm@preamble The macro executes the option given as an argument only if it's specified in the preamble, as the options of \backslash usepackage and/or the argument of \backslash geometry. Otherwise, the macro would print the warning message and ignores the option setting.

32 \backslash def\ifGm@preamble#1{%

33 \backslash ifGm@newgm

34 \backslash Gm@warning{‘#1’: not available in ‘\string\newgeometry’; skipped}%

35 \backslash else

36 \backslash expandafter\@firstofone

37 \backslash fi}%

\backslash Gm@Dhratio The default values for the horizontal and vertical *marginalratio* are defined. \backslash Gm@Dhratio two denotes the default value of horizontal *marginratio* for twoside page layout with left and right margins swapped on verso pages, which is set by twoside.

```

38 \def\Gm@Dhratio{1:1}% = left:right default for oneside
39 \def\Gm@Dhratio{2:3}% = inner:outer default for twoside.
40 \def\Gm@Dvratio{2:3}% = top:bottom default

\Gm@Dhscale The default values for the horizontal and vertical scale are defined with 0.7.
\Gm@Dvscale 41 \def\Gm@Dhscale{0.7}%
42 \def\Gm@Dvscale{0.7}%

\Gm@dvinfos The driver names.
\Gm@dvipdfm 43 \def\Gm@dvinfos{dvips}%
\Gm@pdftex 44 \def\Gm@dvipdfm{dvipdfm}%
\Gm@luatex 45 \def\Gm@pdftex{pdftex}%
\Gm@xetex 46 \def\Gm@luatex{luatex}%
\Gm@vtex 47 \def\Gm@xetex{xetex}%
48 \def\Gm@vtex{vtex}%

\Gm@true The macros for true and false.
\Gm@false 49 \def\Gm@true{true}%
50 \def\Gm@false{false}%

\Gm@orgpw These macros keep original paper (media) size intact.
\Gm@orgph 51 \edef\Gm@orgpw{\the\paperwidth}%
52 \edef\Gm@orgph{\the\paperheight}%

\Gm@savelength The macro saves the specified length to \Gm@restore.
53 \def\Gm@savelength#1{%
54 \g@addto@macro\Gm@restore{\expandafter\noexpand\expandafter\csname
55 #1\endcsname\expandafter=\expandafter\the\csname #1\endcsname\relax}}%

\Gm@saveboolean The macro saves the specified boolean to \Gm@restore.
56 \def\Gm@saveboolean#1{%
57 \csname if#1\endcsname
58 \g@addto@macro\Gm@restore{\expandafter\noexpand\csname #1true\endcsname}%
59 \else
60 \g@addto@macro\Gm@restore{\expandafter\noexpand\csname #1false\endcsname}%
61 \fi}%

\Gm@restore The initialization for \Gm@restore.
62 \def\Gm@restore{}%

\Gm@save The definition of the macro saving the real lengths  $\TeX$  options.
63 \def\Gm@save{%
64 \Gm@savelength{paperwidth}%
65 \Gm@savelength{paperheight}%
66 \Gm@savelength{textwidth}%
67 \Gm@savelength{textheight}%
68 \Gm@savelength{evensidemargin}%
69 \Gm@savelength{oddsidemargin}%
70 \Gm@savelength{topmargin}%
71 \Gm@savelength{headheight}%
72 \Gm@savelength{headsep}%
73 \Gm@savelength{topskip}%
74 \Gm@savelength{footskip}%
75 \Gm@savelength{baselineskip}%
76 \Gm@savelength{marginparwidth}%
77 \Gm@savelength{marginparsep}%
78 \Gm@savelength{columnsep}%
79 \Gm@savelength{hoffset}%
80 \Gm@savelength{voffset}
81 \Gm@savelength{Gm@layoutwidth}%
82 \Gm@savelength{Gm@layoutheight}%
83 \Gm@savelength{Gm@layoutoffset}%
84 \Gm@savelength{Gm@layoutvoffset}%
85 \Gm@saveboolean{@twocolumn}%

```

```

86 \Gm@saveboolean{@twoside}%
87 \Gm@saveboolean{@mparswitch}%
88 \Gm@saveboolean{@reversemargin}}%

```

`\Gm@initnewgm` The macro initializes the parameters for layout in `\newgeometry`.

```

89 \def\Gm@initnewgm{%
90 \Gm@passfalse
91 \Gm@swap@papersizefalse
92 \Gm@dimlist={}
93 \Gm@hbodyfalse
94 \Gm@vbodyfalse
95 \Gm@heightroundedfalse
96 \Gm@includeheadfalse
97 \Gm@includefootfalse
98 \Gm@includempfalse
99 \let\Gm@width\@undefined
100 \let\Gm@height\@undefined
101 \let\Gm@textwidth\@undefined
102 \let\Gm@textheight\@undefined
103 \let\Gm@lines\@undefined
104 \let\Gm@hscale\@undefined
105 \let\Gm@vscale\@undefined
106 \let\Gm@hmarginratio\@undefined
107 \let\Gm@vmarginratio\@undefined
108 \let\Gm@lmargin\@undefined
109 \let\Gm@rmargin\@undefined
110 \let\Gm@tmargin\@undefined
111 \let\Gm@bmargin\@undefined
112 \Gm@layoutfalse
113 \Gm@layouthoffset\z@
114 \Gm@layoutvoffset\z@
115 \Gm@bindingoffset\z@}%

```

`\Gm@initall` This initialization is called as soon as the package is load It's also called as soon as reset option is specified.

```

116 \def\Gm@initall{%
117 \let\Gm@driver\@empty
118 \let\Gm@truedimen\@empty
119 \let\Gm@paper\@undefined
120 \Gm@resetpaperfalse
121 \Gm@landscapefalse
122 \Gm@verbosefalse
123 \Gm@showframefalse
124 \Gm@showcropfalse
125 \Gm@newgmfalse
126 \Gm@initnewgm}%

```

`\Gm@setdriver` The macro sets the specified driver.

```

127 \def\Gm@setdriver#1{%
128 \expandafter\let\expandafter\Gm@driver\csname Gm@#1\endcsname}%

```

`\Gm@unsetdriver` The macro unsets the specified driver if it has been set.

```

129 \def\Gm@unsetdriver#1{%
130 \expandafter\ifx\csname Gm@#1\endcsname\Gm@driver\let\Gm@driver\@empty\fi}%

```

`\Gm@setbool` The macros for boolean option processing.

```

\Gm@setboolrev 131 \def\Gm@setbool{\@dblarg\Gm@@setbool}%
132 \def\Gm@setboolrev{\@dblarg\Gm@@setboolrev}%
133 \def\Gm@@setbool[#1]#2#3{\Gm@doif{#1}{#3}{\csname Gm@#2\Gm@bool\endcsname}}%
134 \def\Gm@@setboolrev[#1]#2#3{\Gm@doifelse{#1}{#3}%
135 {\csname Gm@#2\Gm@false\endcsname}{\csname Gm@#2\Gm@true\endcsname}}%

```

`\Gm@doif` `\Gm@doif` excutes the third argument #3 using a boolean value #2 of a option #1. `\Gm@doifelse`

`\Gm@doifelse` executes the third argument #3 if a boolean option #1 with its value #2 true, and executes the fourth argument #4 if false.

```

136 \def\Gm@doif#1#2#3{%
137   \lowercase{\def\Gm@bool{#2}}%
138   \ifx\Gm@bool\@empty
139     \let\Gm@bool\Gm@true
140   \fi
141   \ifx\Gm@bool\Gm@true
142   \else
143     \ifx\Gm@bool\Gm@false
144     \else
145       \let\Gm@bool\relax
146     \fi
147   \fi
148   \ifx\Gm@bool\relax
149     \Gm@warning{‘#1’ should be set to ‘true’ or ‘false’}%
150   \else
151     #3
152   \fi}%
153 \def\Gm@doifelse#1#2#3#4{%
154   \Gm@doif{#1}{#2}{\ifx\Gm@bool\Gm@true #3\else #4\fi}}%

```

`\Gm@reverse` The macro reverses a bool value.

```

155 \def\Gm@reverse#1{%
156   \csname ifGm@#1\endcsname
157   \csname Gm@#1false\endcsname\else\csname Gm@#1true\endcsname\fi}%

```

`\Gm@defbylen` Macros `\Gm@defbylen` and `\Gm@defbycnt` can be used to define `\Gm@xxxx` variables by length and counter respectively with calc package.

```

158 \def\Gm@defbylen#1#2{%
159   \begingroup\setlength\@tempdima{#2}%
160   \expandafter\xdef\csname Gm@#1\endcsname{\the\@tempdima}\endgroup}%
161 \def\Gm@defbycnt#1#2{%
162   \begingroup\setcounter{Gm@tempcnt}{#2}%
163   \expandafter\xdef\csname Gm@#1\endcsname{\the\value{Gm@tempcnt}}\endgroup}%

```

`\Gm@set@ratio` The macro parses the value of options specifying marginal ratios, which is used in `\Gm@setbyratio` macro.

```

164 \def\Gm@sep@ratio#1:#2{\@tempcnta=#1\@tempcntb=#2}%

```

`\Gm@setbyratio` The macro determines the dimension specified by #4 calculating $\#3 \times a/b$, where a and b are given by `\Gm@ratio` with $a : b$ value. If #1 in brackets is `b`, a and b are swapped. The second argument with `h` or `v` denoting horizontal or vertical is not used in this macro.

```

165 \def\Gm@setbyratio[#1]#2#3#4{% determine #4 by ratio
166   \expandafter\Gm@sep@ratio\Gm@ratio\relax
167   \if#1b
168     \edef\@tempa{\the\@tempcnta}%
169     \@tempcnta=\@tempcntb
170     \@tempcntb=\@tempa\relax
171   \fi
172   \expandafter\setlength\expandafter\@tempdimb\expandafter
173     {\csname Gm@#3\endcsname}%
174   \ifnum\@tempcntb>\z@
175     \multiply\@tempdimb\@tempcnta
176     \divide\@tempdimb\@tempcntb
177   \fi
178   \expandafter\edef\csname Gm@#4\endcsname{\the\@tempdimb}}%

```

`\Gm@detiv` This macro determines the fourth length(#4) from #1(layoutwidth or layoutheight), #2 and #3. It is used in `\Gm@detail` macro.

```

179 \def\Gm@detiv#1#2#3#4{% determine #4.
180   \expandafter\setlength\expandafter\@tempdima\expandafter
181     {\csname Gm@layout#1\endcsname}%
182   \expandafter\setlength\expandafter\@tempdimb\expandafter
183     {\csname Gm@#2\endcsname}%
184   \addtolength\@tempdima{-\@tempdimb}%

```

```

185 \expandafter\setlength\expandafter\@tempdimb\expandafter
186   {\csname Gm@#3\endcsname}%
187 \addtolength\@tempdima{-\@tempdimb}%
188 \ifdim\@tempdima<z@
189   \Gm@warning{‘#4’ results in NEGATIVE (\the\@tempdima).%
190     ^^J\@spaces ‘#2’ or ‘#3’ should be shortened in length}%
191 \fi
192 \expandafter\edef\csname Gm@#4\endcsname{\the\@tempdima}%

```

`\Gm@detiandiii` This macro determines #2 and #3 from #1 with the first argument (#1) can be width or height, which is expanded into dimensions of paper and total body. It is used in `\Gm@detail1` macro.

```

193 \def\Gm@detiandiii#1#2#3{% determine #2 and #3.
194   \expandafter\setlength\expandafter\@tempdima\expandafter
195     {\csname Gm@layout#1\endcsname}%
196   \expandafter\setlength\expandafter\@tempdimb\expandafter
197     {\csname Gm@#1\endcsname}%
198   \addtolength\@tempdima{-\@tempdimb}%
199   \ifdim\@tempdima<z@
200     \Gm@warning{‘#2’ and ‘#3’ result in NEGATIVE (\the\@tempdima).%
201       ^^J\@spaces ‘#1’ should be shortened in length}%
202   \fi
203   \ifx\Gm@ratio\@undefined
204     \expandafter\Gm@sep@ratio\Gm@Dratio\relax
205   \else
206     \expandafter\Gm@sep@ratio\Gm@ratio\relax
207     \ifnum\@tempcntb>z@\else
208       \Gm@warning{margin ratio a:b should be non-zero; default used}%
209       \expandafter\Gm@sep@ratio\Gm@Dratio\relax
210     \fi
211   \fi
212   \@tempdimb=\@tempdima
213   \advance\@tempcntb\@tempcnta
214   \divide\@tempdima\@tempcntb
215   \multiply\@tempdima\@tempcnta
216   \advance\@tempdimb-\@tempdima
217   \expandafter\edef\csname Gm@#2\endcsname{\the\@tempdima}%
218   \expandafter\edef\csname Gm@#3\endcsname{\the\@tempdimb}%

```

`\Gm@detail` This macro determines partition of each direction. The first argument (#1) should be h or v, the second (#2) width or height, the third (#3) lmargin or top, and the last (#4) rmargin or bottom.

```

219 \def\Gm@detail#1#2#3#4{%
220   \@tempcnta\z@
221   \if#1h
222     \let\Gm@ratio\Gm@hmarginratio
223     \edef\Gm@Dratio{\if@twoside\Gm@Dhratio\else\Gm@Dratio\fi}%
224   \else
225     \let\Gm@ratio\Gm@vmarginratio
226     \edef\Gm@Dratio{\Gm@Dvratio}%
227   \fi

```

`\@tempcnta` is treated as a three-digit binary value with top, middle and bottom denoted left(top), width(height) and right(bottom) margins user specified respectively.

```

228   \if#1h
229     \ifx\Gm@lmargin\@undefined\else\advance\@tempcnta4\relax\fi
230     \ifGm@hbody\advance\@tempcnta2\relax\fi
231     \ifx\Gm@rmargin\@undefined\else\advance\@tempcnta1\relax\fi
232     \Gm@cnth\@tempcnta
233   \else
234     \ifx\Gm@tmargin\@undefined\else\advance\@tempcnta4\relax\fi
235     \ifGm@vbody\advance\@tempcnta2\relax\fi
236     \ifx\Gm@bmargin\@undefined\else\advance\@tempcnta1\relax\fi
237     \Gm@cntv\@tempcnta
238   \fi

```

Case the value is 000 (=0) with nothing fixed (default):

```

239   \ifcase\@tempcnta

```

```

240 \if#1h
241 \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
242 \else
243 \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
244 \fi
245 \Gm@detiandiii{#2}{#3}{#4}%
Case 001 (=1) with right(bottom) fixed:
246 \or
247 \ifx\Gm@mratio\@undefined
248 \if#1h
249 \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
250 \else
251 \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
252 \fi
253 \setlength\@tempdimc{\@nameuse{Gm@#4}}%
254 \Gm@detiandiii{#2}{#3}{#4}%
255 \expandafter\let\csname Gm@#2\endcsname\@undefined
256 \Gm@defbylen{#4}{\@tempdimc}%
257 \else
258 \Gm@setbyratio[f]{#1}{#4}{#3}%
259 \fi
260 \Gm@detiv{#2}{#3}{#4}{#2}%
Case 010 (=2) with width(height) fixed:
261 \or\Gm@detiandiii{#2}{#3}{#4}%
Case 011 (=3) with both width(height) and right(bottom) fixed:
262 \or\Gm@detiv{#2}{#2}{#4}{#3}%
Case 100 (=4) with left(top) fixed:
263 \or
264 \ifx\Gm@mratio\@undefined
265 \if#1h
266 \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
267 \else
268 \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
269 \fi
270 \setlength\@tempdimc{\@nameuse{Gm@#3}}%
271 \Gm@detiandiii{#2}{#4}{#3}%
272 \expandafter\let\csname Gm@#2\endcsname\@undefined
273 \Gm@defbylen{#3}{\@tempdimc}%
274 \else
275 \Gm@setbyratio[b]{#1}{#3}{#4}%
276 \fi
277 \Gm@detiv{#2}{#3}{#4}{#2}%
Case 101 (=5) with both left(top) and right(bottom) fixed:
278 \or\Gm@detiv{#2}{#3}{#4}{#2}%
Case 110 (=6) with both left(top) and width(height) fixed:
279 \or\Gm@detiv{#2}{#2}{#3}{#4}%
Case 111 (=7) with all fixed though it is over-specified:
280 \or\Gm@warning{Over-specification in ‘#1’-direction.%
281 \~J\@spaces ‘#2’ (\@nameuse{Gm@#2}) is ignored}%
282 \Gm@detiv{#2}{#3}{#4}{#2}%
283 \else\fi}%

```

\Gm@clean The macro for setting unspecified dimensions to be \@undefined. This is used by \geometry macro.

```

284 \def\Gm@clean{%
285 \ifnum\Gm@cnth<4\let\Gm@lmargin\@undefined\fi
286 \ifodd\Gm@cnth\else\let\Gm@rmargin\@undefined\fi
287 \ifnum\Gm@cntv<4\let\Gm@tmargin\@undefined\fi
288 \ifodd\Gm@cntv\else\let\Gm@bmargin\@undefined\fi
289 \ifGm@hbody\else
290 \let\Gm@hscale\@undefined
291 \let\Gm@width\@undefined

```

```

292 \let\Gm@textwidth\@undefined
293 \fi
294 \ifGm@vbody\else
295 \let\Gm@vscale\@undefined
296 \let\Gm@height\@undefined
297 \let\Gm@textheight\@undefined
298 \fi
299 }%

```

`\Gm@parse@divide` The macro parses (h,v)divide options.

```

300 \def\Gm@parse@divide#1#2#3#4{%
301 \def\Gm@star{*}%
302 \@tempcnta\z@
303 \@for\Gm@tmp:=#1\do{%
304 \expandafter\KV@sp@def\expandafter\Gm@frag\expandafter{\Gm@tmp}%
305 \edef\Gm@value{\Gm@frag}%
306 \ifcase\@tempcnta\relax\edef\Gm@key{#2}%
307 \or\edef\Gm@key{#3}%
308 \else\edef\Gm@key{#4}%
309 \fi
310 \@nameuse{Gm@set\Gm@key false}%
311 \ifx\empty\Gm@value\else
312 \ifx\Gm@star\Gm@value\else
313 \setkeys{Gm}{\Gm@key=\Gm@value}%
314 \fi\fi
315 \advance\@tempcnta\@ne}%
316 \let\Gm@star\relax}%

```

`\Gm@branch` The macro splits a value into the same two values.

```

317 \def\Gm@branch#1#2#3{%
318 \@tempcnta\z@
319 \@for\Gm@tmp:=#1\do{%
320 \KV@sp@def\Gm@frag{\Gm@tmp}%
321 \edef\Gm@value{\Gm@frag}%
322 \ifcase\@tempcnta\relax% cnta == 0
323 \setkeys{Gm}{#2=\Gm@value}%
324 \or% cnta == 1
325 \setkeys{Gm}{#3=\Gm@value}%
326 \else\fi
327 \advance\@tempcnta\@ne}%
328 \ifnum\@tempcnta=\@ne
329 \setkeys{Gm}{#3=\Gm@value}%
330 \fi}%

```

`\Gm@magtooffset` This macro is used to adjust offsets by `\mag`.

```

331 \def\Gm@magtooffset{%
332 \@tempdima=\mag\Gm@truedimen sp%
333 \@tempdimb=1\Gm@truedimen in%
334 \divide\@tempdimb\@tempdima
335 \multiply\@tempdimb\@m
336 \addtolength{\hoffset}{1\Gm@truedimen in}%
337 \addtolength{\voffset}{1\Gm@truedimen in}%
338 \addtolength{\hoffset}{-\the\@tempdimb}%
339 \addtolength{\voffset}{-\the\@tempdimb}}%

```

`\Gm@setlength` This macro stores L^AT_EX native dimensions, which are stored and set afterwards.

```

340 \def\Gm@setlength#1#2{%
341 \let\Gm@len=\relax\let\Gm@td=\relax
342 \edef\addtolist{\noexpand\Gm@dimlist=%
343 {\the\Gm@dimlist \Gm@len{#1}{#2}}\addtolist}%

```

`\Gm@expandlengths` This macro processes `\Gm@dimlist`.

```

344 \def\Gm@expandlengths{%
345 \def\Gm@td{\Gm@truedimen}%
346 \def\Gm@len##1##2{\setlength{##1}{##2}}%
347 \the\Gm@dimlist}%

```

`\Gm@setsize` The macro sets paperwidth and paperheight dimensions using `\Gm@setlength` macro.

```
348 \def\Gm@setsize#1(#2,#3)#4{%
349   \let\Gm@td\relax
350   \expandafter\Gm@setlength\csname #1width\endcsname{#2\Gm@td #4}%
351   \expandafter\Gm@setlength\csname #1height\endcsname{#3\Gm@td #4}%
352   \ifGm@landscape\Gm@swap@papersizetrue\else\Gm@swap@papersizefalse\fi}%
```

`\Gm@setpaper@ifpre` The macro changes the paper size.

```
353 \def\Gm@setpaper@ifpre#1{%
354   \ifGm@preamble{#1}{\def\Gm@paper{#1}\@nameuse{Gm@#1}{paper}}}%
```

Various paper size are defined here.

```
355 \@namedef{Gm@a0paper}#1{\Gm@setsize{#1}(841,1189){mm}}% ISO A0
356 \@namedef{Gm@a1paper}#1{\Gm@setsize{#1}(594,841){mm}}% ISO A1
357 \@namedef{Gm@a2paper}#1{\Gm@setsize{#1}(420,594){mm}}% ISO A2
358 \@namedef{Gm@a3paper}#1{\Gm@setsize{#1}(297,420){mm}}% ISO A3
359 \@namedef{Gm@a4paper}#1{\Gm@setsize{#1}(210,297){mm}}% ISO A4
360 \@namedef{Gm@a5paper}#1{\Gm@setsize{#1}(148,210){mm}}% ISO A5
361 \@namedef{Gm@a6paper}#1{\Gm@setsize{#1}(105,148){mm}}% ISO A6
362 \@namedef{Gm@b0paper}#1{\Gm@setsize{#1}(1000,1414){mm}}% ISO B0
363 \@namedef{Gm@b1paper}#1{\Gm@setsize{#1}(707,1000){mm}}% ISO B1
364 \@namedef{Gm@b2paper}#1{\Gm@setsize{#1}(500,707){mm}}% ISO B2
365 \@namedef{Gm@b3paper}#1{\Gm@setsize{#1}(353,500){mm}}% ISO B3
366 \@namedef{Gm@b4paper}#1{\Gm@setsize{#1}(250,353){mm}}% ISO B4
367 \@namedef{Gm@b5paper}#1{\Gm@setsize{#1}(176,250){mm}}% ISO B5
368 \@namedef{Gm@b6paper}#1{\Gm@setsize{#1}(125,176){mm}}% ISO B6
369 \@namedef{Gm@c0paper}#1{\Gm@setsize{#1}(917,1297){mm}}% ISO C0
370 \@namedef{Gm@c1paper}#1{\Gm@setsize{#1}(648,917){mm}}% ISO C1
371 \@namedef{Gm@c2paper}#1{\Gm@setsize{#1}(458,648){mm}}% ISO C2
372 \@namedef{Gm@c3paper}#1{\Gm@setsize{#1}(324,458){mm}}% ISO C3
373 \@namedef{Gm@c4paper}#1{\Gm@setsize{#1}(229,324){mm}}% ISO C4
374 \@namedef{Gm@c5paper}#1{\Gm@setsize{#1}(162,229){mm}}% ISO C5
375 \@namedef{Gm@c6paper}#1{\Gm@setsize{#1}(114,162){mm}}% ISO C6
376 \@namedef{Gm@b0j}#1{\Gm@setsize{#1}(1030,1456){mm}}% JIS B0
377 \@namedef{Gm@b1j}#1{\Gm@setsize{#1}(728,1030){mm}}% JIS B1
378 \@namedef{Gm@b2j}#1{\Gm@setsize{#1}(515,728){mm}}% JIS B2
379 \@namedef{Gm@b3j}#1{\Gm@setsize{#1}(364,515){mm}}% JIS B3
380 \@namedef{Gm@b4j}#1{\Gm@setsize{#1}(257,364){mm}}% JIS B4
381 \@namedef{Gm@b5j}#1{\Gm@setsize{#1}(182,257){mm}}% JIS B5
382 \@namedef{Gm@b6j}#1{\Gm@setsize{#1}(128,182){mm}}% JIS B6
383 \@namedef{Gm@ansipaper}#1{\Gm@setsize{#1}(8.5,11){in}}%
384 \@namedef{Gm@ansibpaper}#1{\Gm@setsize{#1}(11,17){in}}%
385 \@namedef{Gm@ansicpaper}#1{\Gm@setsize{#1}(17,22){in}}%
386 \@namedef{Gm@ansidpaper}#1{\Gm@setsize{#1}(22,34){in}}%
387 \@namedef{Gm@ansiepaper}#1{\Gm@setsize{#1}(34,44){in}}%
388 \@namedef{Gm@letterpaper}#1{\Gm@setsize{#1}(8.5,11){in}}%
389 \@namedef{Gm@legalpaper}#1{\Gm@setsize{#1}(8.5,14){in}}%
390 \@namedef{Gm@executivepaper}#1{\Gm@setsize{#1}(7.25,10.5){in}}%
391 \@namedef{Gm@screen}#1{\Gm@setsize{#1}(225,180){mm}}%
```

‘paper’ paper takes a paper name as its value.

```
392 \define@key{Gm}{paper}{\setkeys{Gm}{#1}}%
393 \let\KV@Gm@papername\KV@Gm@paper
```

‘a[0-6]paper’ The following paper names are available.

```
‘b[0-6]paper’ 394 \define@key{Gm}{a0paper}[true]{\Gm@setpaper@ifpre{a0paper}}%
‘b[0-6]j’ 395 \define@key{Gm}{a1paper}[true]{\Gm@setpaper@ifpre{a1paper}}%
‘ansi[a-e]paper’ 396 \define@key{Gm}{a2paper}[true]{\Gm@setpaper@ifpre{a2paper}}%
‘letterpaper’ 397 \define@key{Gm}{a3paper}[true]{\Gm@setpaper@ifpre{a3paper}}%
‘legalpaper’ 398 \define@key{Gm}{a4paper}[true]{\Gm@setpaper@ifpre{a4paper}}%
‘executivepaper’ 399 \define@key{Gm}{a5paper}[true]{\Gm@setpaper@ifpre{a5paper}}%
‘screen’ 400 \define@key{Gm}{a6paper}[true]{\Gm@setpaper@ifpre{a6paper}}%
401 \define@key{Gm}{b0paper}[true]{\Gm@setpaper@ifpre{b0paper}}%
402 \define@key{Gm}{b1paper}[true]{\Gm@setpaper@ifpre{b1paper}}%
403 \define@key{Gm}{b2paper}[true]{\Gm@setpaper@ifpre{b2paper}}%
```

```

404 \define@key{Gm}{b3paper}[true]{\Gm@setpaper@ifpre{b3paper}}%
405 \define@key{Gm}{b4paper}[true]{\Gm@setpaper@ifpre{b4paper}}%
406 \define@key{Gm}{b5paper}[true]{\Gm@setpaper@ifpre{b5paper}}%
407 \define@key{Gm}{b6paper}[true]{\Gm@setpaper@ifpre{b6paper}}%
408 \define@key{Gm}{c0paper}[true]{\Gm@setpaper@ifpre{c0paper}}%
409 \define@key{Gm}{c1paper}[true]{\Gm@setpaper@ifpre{c1paper}}%
410 \define@key{Gm}{c2paper}[true]{\Gm@setpaper@ifpre{c2paper}}%
411 \define@key{Gm}{c3paper}[true]{\Gm@setpaper@ifpre{c3paper}}%
412 \define@key{Gm}{c4paper}[true]{\Gm@setpaper@ifpre{c4paper}}%
413 \define@key{Gm}{c5paper}[true]{\Gm@setpaper@ifpre{c5paper}}%
414 \define@key{Gm}{c6paper}[true]{\Gm@setpaper@ifpre{c6paper}}%
415 \define@key{Gm}{b0j}[true]{\Gm@setpaper@ifpre{b0j}}%
416 \define@key{Gm}{b1j}[true]{\Gm@setpaper@ifpre{b1j}}%
417 \define@key{Gm}{b2j}[true]{\Gm@setpaper@ifpre{b2j}}%
418 \define@key{Gm}{b3j}[true]{\Gm@setpaper@ifpre{b3j}}%
419 \define@key{Gm}{b4j}[true]{\Gm@setpaper@ifpre{b4j}}%
420 \define@key{Gm}{b5j}[true]{\Gm@setpaper@ifpre{b5j}}%
421 \define@key{Gm}{b6j}[true]{\Gm@setpaper@ifpre{b6j}}%
422 \define@key{Gm}{ansipaper}[true]{\Gm@setpaper@ifpre{ansipaper}}%
423 \define@key{Gm}{ansibpaper}[true]{\Gm@setpaper@ifpre{ansibpaper}}%
424 \define@key{Gm}{ansicpaper}[true]{\Gm@setpaper@ifpre{ansicpaper}}%
425 \define@key{Gm}{ansidpaper}[true]{\Gm@setpaper@ifpre{ansidpaper}}%
426 \define@key{Gm}{ansiepaper}[true]{\Gm@setpaper@ifpre{ansiepaper}}%
427 \define@key{Gm}{letterpaper}[true]{\Gm@setpaper@ifpre{letterpaper}}%
428 \define@key{Gm}{legalpaper}[true]{\Gm@setpaper@ifpre{legalpaper}}%
429 \define@key{Gm}{executivepaper}[true]{\Gm@setpaper@ifpre{executivepaper}}%
430 \define@key{Gm}{screen}[true]{\Gm@setpaper@ifpre{screen}}%

```

‘paperwidth’ Direct specification for paper size is also possible.

```

‘paperheight’ 431 \define@key{Gm}{paperwidth}{\ifGm@preamble{paperwidth}{%
‘papersize’ 432 \def\Gm@paper{custom}\Gm@setlength\paperwidth{#1}}%
433 \define@key{Gm}{paperheight}{\ifGm@preamble{paperheight}{%
434 \def\Gm@paper{custom}\Gm@setlength\paperheight{#1}}%
435 \define@key{Gm}{papersize}{\ifGm@preamble{papersize}{%
436 \def\Gm@paper{custom}\Gm@branch{#1}{paperwidth}{paperheight}}%

```

‘layout’ Direct specification for layout size is also possible.

```

‘layoutwidth’ 437 \define@key{Gm}{layout}{\Gm@layouttrue\@nameuse{Gm@#1}{Gm@layout}}%
‘layoutheight’ 438 \let\KV@Gm@layoutname\KV@Gm@layout
‘layoutsize’ 439 \define@key{Gm}{layoutwidth}{\Gm@layouttrue\Gm@setlength\Gm@layoutwidth{#1}}%
440 \define@key{Gm}{layoutheight}{\Gm@layouttrue\Gm@setlength\Gm@layoutheight{#1}}%
441 \define@key{Gm}{layoutsize}{\Gm@branch{#1}{layoutwidth}{layoutheight}}%

```

‘landscape’ Paper orientation setting.

```

‘portrait’ 442 \define@key{Gm}{landscape}[true]{\ifGm@preamble{landscape}{%
443 \Gm@doifelse{landscape}{#1}%
444 {\ifGm@landscape\else\Gm@landscapetrue\Gm@reverse{swap@papersize}\fi}%
445 {\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}}%
446 \define@key{Gm}{portrait}[true]{\ifGm@preamble{portrait}{%
447 \Gm@doifelse{portrait}{#1}%
448 {\ifGm@landscape\Gm@landscapefalse\Gm@reverse{swap@papersize}\fi}%
449 {\ifGm@landscape\else\Gm@landscapetrue\Gm@reverse{swap@papersize}\fi}}%

```

‘hscale’ These options can determine the length(s) of *total body* giving *scale(s)* against the paper size.

```

‘vscale’ 450 \define@key{Gm}{hscale}{\Gm@hbodytrue\edef\Gm@hscale{#1}}%
‘scale’ 451 \define@key{Gm}{vscale}{\Gm@vbodytrue\edef\Gm@vscale{#1}}%
452 \define@key{Gm}{scale}{\Gm@branch{#1}{hscale}{vscale}}%

```

‘width’ These options give concrete dimension(s) of *total body*. totalwidth and totalheight are aliases of width and height respectively.

```

‘total’ 453 \define@key{Gm}{width}{\Gm@hbodytrue\Gm@defbylen{width}{#1}}%
‘totalwidth’ 454 \define@key{Gm}{height}{\Gm@vbodytrue\Gm@defbylen{height}{#1}}%
‘totalheight’ 455 \define@key{Gm}{total}{\Gm@branch{#1}{width}{height}}%
456 \let\KV@Gm@totalwidth\KV@Gm@width
457 \let\KV@Gm@totalheight\KV@Gm@height

```

‘textwidth’ These options directly sets the dimensions `\textwidth` and `\textheight`. `body` is an alias of `text`.

‘textheight’ 458 `\define@key{Gm}{textwidth}{\Gm@hbodytrue\Gm@defbylen{textwidth}{#1}}%`
‘text’ 459 `\define@key{Gm}{textheight}{\Gm@vbodytrue\Gm@defbylen{textheight}{#1}}%`
‘body’ 460 `\define@key{Gm}{text}{\Gm@branch{#1}{textwidth}{textheight}}%`
461 `\let\KV@Gm@body\KV@Gm@text`

‘lines’ The option sets `\textheight` with the number of lines.

462 `\define@key{Gm}{lines}{\Gm@vbodytrue\Gm@defbycnt{lines}{#1}}%`

‘includehead’ The options take the corresponding dimensions as part of *body*.

‘includefoot’ 463 `\define@key{Gm}{includehead}[true]{\Gm@setbool{includehead}{#1}}%`

‘includeheadfoot’ 464 `\define@key{Gm}{includefoot}[true]{\Gm@setbool{includefoot}{#1}}%`

‘includemp’ 465 `\define@key{Gm}{includeheadfoot}[true]{\Gm@doifelse{includeheadfoot}{#1}}%`

‘includeall’ 466 `{\Gm@includeheadtrue\Gm@includefoottrue}%`
467 `{\Gm@includeheadfalse\Gm@includefootfalse}%`
468 `\define@key{Gm}{includemp}[true]{\Gm@setbool{includemp}{#1}}%`
469 `\define@key{Gm}{includeall}[true]{\Gm@doifelse{includeall}{#1}}%`
470 `{\Gm@includeheadtrue\Gm@includefoottrue\Gm@includemptrue}%`
471 `{\Gm@includeheadfalse\Gm@includefootfalse\Gm@includempfalse}%`

‘ignorehead’ These options exclude *head*, *foot* and *marginpars* when determining *body*.

‘ignorefoot’ 472 `\define@key{Gm}{ignorehead}[true]{%`

‘ignoreheadfoot’ 473 `\Gm@setboolrev[ignorehead]{includehead}{#1}}%`

‘ignoremp’ 474 `\define@key{Gm}{ignorefoot}[true]{%`

‘ignoreall’ 475 `\Gm@setboolrev[ignorefoot]{includefoot}{#1}}%`
476 `\define@key{Gm}{ignoreheadfoot}[true]{\Gm@doifelse{ignoreheadfoot}{#1}}%`
477 `{\Gm@includeheadfalse\Gm@includefootfalse}%`
478 `{\Gm@includeheadtrue\Gm@includefoottrue}%`
479 `\define@key{Gm}{ignoremp}[true]{%`
480 `\Gm@setboolrev[ignoremp]{includemp}{#1}}%`
481 `\define@key{Gm}{ignoreall}[true]{\Gm@doifelse{ignoreall}{#1}}%`
482 `{\Gm@includeheadfalse\Gm@includefootfalse\Gm@includempfalse}%`
483 `{\Gm@includeheadtrue\Gm@includefoottrue\Gm@includemptrue}}%`

‘heightrounded’ The option rounds `\textheight` to n-times of `\baselineskip` plus `\topskip`.

484 `\define@key{Gm}{heightrounded}[true]{\Gm@setbool{heightrounded}{#1}}%`

‘hdivide’ The options are useful to specify partitioning in each direction of the paper.

‘vdivide’ 485 `\define@key{Gm}{hdivide}{\Gm@parse@divide{#1}{lmargin}{width}{rmargin}}%`

‘divide’ 486 `\define@key{Gm}{vdivide}{\Gm@parse@divide{#1}{tmargin}{height}{bmargin}}%`
487 `\define@key{Gm}{divide}{\Gm@parse@divide{#1}{lmargin}{width}{rmargin}}%`
488 `\Gm@parse@divide{#1}{tmargin}{height}{bmargin}}%`

‘lmargin’ These options set *margins*. `left`, `inner`, `innermargin` are aliases of `lmargin`. `right`, `outer`,
‘rmargin’ `outermargin` are aliases of `rmargin`. `top` and `bottom` are aliases of `tmargin` and `bmargin` respec-
‘tmargin’ tively.

‘bmargin’ 489 `\define@key{Gm}{lmargin}{\Gm@defbylen{lmargin}{#1}}%`

‘left’ 490 `\define@key{Gm}{rmargin}{\Gm@defbylen{rmargin}{#1}}%`

‘inner’ 491 `\let\KV@Gm@left\KV@Gm@lmargin`

‘innermargin’ 492 `\let\KV@Gm@inner\KV@Gm@lmargin`

‘right’ 493 `\let\KV@Gm@innermargin\KV@Gm@lmargin`

‘outer’ 494 `\let\KV@Gm@right\KV@Gm@rmargin`

‘outermargin’ 495 `\let\KV@Gm@outer\KV@Gm@rmargin`

‘top’ 496 `\let\KV@Gm@outermargin\KV@Gm@rmargin`

‘bottom’ 497 `\define@key{Gm}{tmargin}{\Gm@defbylen{tmargin}{#1}}%`
498 `\define@key{Gm}{bmargin}{\Gm@defbylen{bmargin}{#1}}%`
499 `\let\KV@Gm@top\KV@Gm@tmargin`
500 `\let\KV@Gm@bottom\KV@Gm@bmargin`

‘hmargin’ These options are shorthands for setting *margins*.

‘vmargin’ 501 `\define@key{Gm}{hmargin}{\Gm@branch{#1}{lmargin}{rmargin}}%`

‘margin’ 502 `\define@key{Gm}{vmargin}{\Gm@branch{#1}{tmargin}{bmargin}}%`
503 `\define@key{Gm}{margin}{\Gm@branch{#1}{lmargin}{tmargin}}%`
504 `\Gm@branch{#1}{rmargin}{bmargin}}%`

‘hmarginratio’ Options specifying the margin ratios.

‘vmarginratio’ 505 \define@key{Gm}{hmarginratio}{\edef\Gm@hmarginratio{#1}}%

‘marginratio’ 506 \define@key{Gm}{vmarginratio}{\edef\Gm@vmarginratio{#1}}%

‘hratio’ 507 \define@key{Gm}{marginratio}{\Gm@branch{#1}{hmarginratio}{vmarginratio}}%

‘vratio’ 508 \let\KV@Gm@hratio\KV@Gm@hmarginratio

‘ratio’ 509 \let\KV@Gm@vratio\KV@Gm@vmarginratio

510 \let\KV@Gm@ratio\KV@Gm@marginratio

‘hcentering’ Useful shorthands to place *body* centered.

‘vcentering’ 511 \define@key{Gm}{hcentering}[true]{\Gm@doifelse{hcentering}{#1}}%

‘centering’ 512 {\def\Gm@hmarginratio{1:1}}{}}%

513 \define@key{Gm}{vcentering}[true]{\Gm@doifelse{vcentering}{#1}}%

514 {\def\Gm@vmarginratio{1:1}}{}}%

515 \define@key{Gm}{centering}[true]{\Gm@doifelse{centering}{#1}}%

516 {\def\Gm@hmarginratio{1:1}\def\Gm@vmarginratio{1:1}}{}}%

‘twoside’ If twoside=true, \@twoside and \@mparswitch is set to true.

517 \define@key{Gm}{twoside}[true]{\Gm@doifelse{twoside}{#1}}%

518 {\@twosidetrue\@mparswitchtrue}{\@twosidefalse\@mparswitchfalse}}%

‘asymmetric’ asymmetric sets \@mparswitchfalse and \@twosidetrue A asymmetric=false has no effect.

519 \define@key{Gm}{asymmetric}[true]{\Gm@doifelse{asymmetric}{#1}}%

520 {\@twosidetrue\@mparswitchfalse}}{}}%

‘bindingoffset’ The macro adds the specified space to the inner margin.

521 \define@key{Gm}{bindingoffset}{\Gm@setlength\Gm@bindingoffset{#1}}%

‘headheight’ The direct settings of *head* and/or *foot* dimensions.

‘headsep’ 522 \define@key{Gm}{headheight}{\Gm@setlength\headheight{#1}}%

‘footskip’ 523 \define@key{Gm}{headsep}{\Gm@setlength\headsep{#1}}%

‘head’ 524 \define@key{Gm}{footskip}{\Gm@setlength\footskip{#1}}%

‘foot’ 525 \let\KV@Gm@head\KV@Gm@headheight

526 \let\KV@Gm@foot\KV@Gm@footskip

‘nohead’ They are only shorthands to set *head* and/or *foot* to be 0pt.

‘nofoot’ 527 \define@key{Gm}{nohead}[true]{\Gm@doifelse{nohead}{#1}}%

‘noheadfoot’ 528 {\Gm@setlength\headheight\z@\Gm@setlength\headsep\z@}}{}}%

529 \define@key{Gm}{nofoot}[true]{\Gm@doifelse{nofoot}{#1}}%

530 {\Gm@setlength\footskip\z@}}{}}%

531 \define@key{Gm}{noheadfoot}[true]{\Gm@doifelse{noheadfoot}{#1}}%

532 {\Gm@setlength\headheight\z@\Gm@setlength\headsep

533 \z@\Gm@setlength\footskip\z@}}{}}%

‘footnotesep’ The option directly sets a native dimension \footnotesep.

534 \define@key{Gm}{footnotesep}{\Gm@setlength{\skip\footins}{#1}}%

‘marginparwidth’ They directly set native dimensions \marginparwidth and \marginparsep.

‘marginpar’ 535 \define@key{Gm}{marginparwidth}{\Gm@setlength\marginparwidth{#1}}%

‘marginparsep’ 536 \let\KV@Gm@marginpar\KV@Gm@marginparwidth

537 \define@key{Gm}{marginparsep}{\Gm@setlength\marginparsep{#1}}%

‘nomarginpar’ The macro is a shorthand for \marginparwidth=0pt and \marginparsep=0pt.

538 \define@key{Gm}{nomarginpar}[true]{\Gm@doifelse{nomarginpar}{#1}}%

539 {\Gm@setlength\marginparwidth\z@\Gm@setlength\marginparsep\z@}}{}}%

‘columnsep’ The option sets a native dimension \columnsep.

540 \define@key{Gm}{columnsep}{\Gm@setlength\columnsep{#1}}%

‘hoffset’ The former two options set native dimensions \hoffset and \voffset. offset can set both of

‘voffset’ them with the same value.

‘offset’ 541 \define@key{Gm}{hoffset}{\Gm@setlength\hoffset{#1}}%

542 \define@key{Gm}{voffset}{\Gm@setlength\voffset{#1}}%

543 \define@key{Gm}{offset}{\Gm@branch{#1}{hoffset}{voffset}}%

```

‘layoutoffset’
‘layoutvoffset’ 544 \define@key{Gm}{layoutoffset}{\Gm@setlength\Gm@layoutoffset{#1}}%
‘layoutoffset’ 545 \define@key{Gm}{layoutvoffset}{\Gm@setlength\Gm@layoutvoffset{#1}}%
546 \define@key{Gm}{layoutoffset}{\Gm@branch{#1}{layoutoffset}{layoutvoffset}}%

‘twocolumn’ The option sets \twocolumn switch.
547 \define@key{Gm}{twocolumn}[true]{%
548 \Gm@doif{twocolumn}{#1}{\csname @twocolumn\Gm@bool\endcsname}}%

‘onecolumn’ This option has the reverse effect of twocolumn option.
549 \define@key{Gm}{onecolumn}[true]{%
550 \Gm@doifelse{onecolumn}{#1}{\@twocolumnfalse}{\@twocolumtrue}}%

‘reversemp’ The both options set \reversemargin.
‘reversemarginpar’ 551 \define@key{Gm}{reversemp}[true]{%
552 \Gm@doif{reversemp}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%
553 \define@key{Gm}{reversemarginpar}[true]{%
554 \Gm@doif{reversemarginpar}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%

‘dviver’
555 \define@key{Gm}{driver}{\ifGm@preamble{driver}{%
556 \edef\@tempa{#1}\edef\@auto{auto}\edef\@none{none}}%
557 \ifx\@tempa\@empty\let\Gm@driver\relax\else
558 \ifx\@tempa\@none\let\Gm@driver\relax\else
559 \ifx\@tempa\@auto\let\Gm@driver\@empty\else
560 \setkeys{Gm}{#1}\fi\fi\fi\let\@auto\relax\let\@none\relax}}%

‘dvips’ The geometry package supports dvips, dvipdfm, pdflatex, luatex and vtex. dvipdfm works like
‘dvipdfm’ dvips.
‘dvipdfmx’ 561 \define@key{Gm}{dvips}[true]{\ifGm@preamble{dvips}{%
‘xdvipdfmx’ 562 \Gm@doifelse{dvips}{#1}{\Gm@setdriver{dvips}}{\Gm@unsetdriver{dvips}}}}%
‘pdftex’ 563 \define@key{Gm}{dvipdfm}[true]{\ifGm@preamble{dvipdfm}{%
‘luatex’ 564 \Gm@doifelse{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{\Gm@unsetdriver{dvipdfm}}}}%
‘xetex’ 565 \define@key{Gm}{dvipdfmx}[true]{\ifGm@preamble{dvipdfm}{%
‘vtex’ 566 \Gm@doifelse{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{\Gm@unsetdriver{dvipdfm}}}}%
567 \define@key{Gm}{xdvipdfmx}[true]{\ifGm@preamble{dvipdfm}{%
568 \Gm@doifelse{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{\Gm@unsetdriver{dvipdfm}}}}%
569 \define@key{Gm}{pdftex}[true]{\ifGm@preamble{pdftex}{%
570 \Gm@doifelse{pdftex}{#1}{\Gm@setdriver{pdftex}}{\Gm@unsetdriver{pdftex}}}}%
571 \define@key{Gm}{luatex}[true]{\ifGm@preamble{luatex}{%
572 \Gm@doifelse{luatex}{#1}{\Gm@setdriver{luatex}}{\Gm@unsetdriver{luatex}}}}%
573 \define@key{Gm}{xetex}[true]{\ifGm@preamble{xetex}{%
574 \Gm@doifelse{xetex}{#1}{\Gm@setdriver{xetex}}{\Gm@unsetdriver{xetex}}}}%
575 \define@key{Gm}{vtex}[true]{\ifGm@preamble{vtex}{%
576 \Gm@doifelse{vtex}{#1}{\Gm@setdriver{vtex}}{\Gm@unsetdriver{vtex}}}}%

‘verbose’ The verbose mode.
577 \define@key{Gm}{verbose}[true]{\ifGm@preamble{verbose}{\Gm@setbool{verbose}{#1}}}%

‘reset’ The option cancels all the options specified before reset, except pass. mag ( $\neq$  1000) with
truedimen cannot be also reset.
578 \define@key{Gm}{reset}[true]{\ifGm@preamble{reset}{%
579 \Gm@doifelse{reset}{#1}{\Gm@restore@org\Gm@initall
580 \ProcessOptionsKV[c]{Gm}\Gm@setdefaultpaper}{}}}%

‘resetpaper’ If resetpaper is set to true, the paper size redefined in the package is discarded and the original
one is restored. This option may be useful to print nonstandard sized documents with normal
printers and papers.
581 \define@key{Gm}{resetpaper}[true]{\ifGm@preamble{resetpaper}{%
582 \Gm@setbool{resetpaper}{#1}}}%

‘mag’ mag is expanded immediately when it is specified. So reset can’t reset mag when it is set with
truedimen.
583 \define@key{Gm}{mag}{\ifGm@preamble{mag}{\mag=#1}}%

```

‘truedimen’ If truedimen is set to true, all of the internal explicit dimensions is changed to *true* dimensions, e.g., 1in is changed to 1truein.

```
584 \define@key{Gm}{truedimen}[true]{\ifGm@preamble{truedimen}{%
585 \Gm@doifelse{truedimen}{#1}{\let\Gm@truedimen\Gm@true}%
586 {\let\Gm@truedimen\@empty}}}%
```

‘pass’ The option makes all the options specified ineffective except verbose switch.

```
587 \define@key{Gm}{pass}[true]{\ifGm@preamble{pass}{\Gm@setbool{pass}{#1}}}%
```

‘showframe’ The showframe option prints page frames to help you understand what the resulting layout is like.

```
588 \define@key{Gm}{showframe}[true]{\Gm@setbool{showframe}{#1}}%
```

‘showcrop’ The showcrop option prints crop marks at each corner of the layout area.

```
589 \define@key{Gm}{showcrop}[true]{\Gm@setbool{showcrop}{#1}}%
```

`\Gm@setdefaultpaper` The macro stores paper dimensions. This macro should be called after `\ProcessOptionsKV[c]{Gm}`. If the landscape option in `\documentclass` is specified, the class immediately swaps the paper dimensions.

```
590 \def\Gm@setdefaultpaper{%
591 \ifx\Gm@paper\@undefined
592 \Gm@setsize{paper}{\strip@pt\paperwidth,\strip@pt\paperheight}{pt}%
593 \Gm@setsize{Gm@layout}{\strip@pt\paperwidth,\strip@pt\paperheight}{pt}%
594 \Gm@swap@papersizefalse
595 \fi}%
```

`\Gm@adjustpaper` The macro checks if paperwidth/height is larger than 0pt, which is used in `\Gm@process`. The paper dimensions can be swapped when paper orientation is changed over by landscape and portrait options.

```
596 \def\Gm@adjustpaper{%
597 \ifdim\paperwidth>p\@else
598 \PackageError{geometry}{%
599 \string\paperwidth\space(\the\paperwidth) too short}{%
600 Set a paper type (e.g., ‘a4paper’).}%
601 \fi
602 \ifdim\paperheight>p\@else
603 \PackageError{geometry}{%
604 \string\paperheight\space(\the\paperheight) too short}{%
605 Set a paper type (e.g., ‘a4paper’).}%
606 \fi
607 \ifGm@swap@papersize
608 \setlength\@tempdima{\paperwidth}%
609 \setlength\paperwidth{\paperheight}%
610 \setlength\paperheight{\@tempdima}%
611 \fi
612 \ifGm@layout\else
613 \setlength\Gm@layoutwidth{\paperwidth}%
614 \setlength\Gm@layoutheight{\paperheight}%
615 \fi}%
```

`\Gm@checkmp` The macro checks whether or not the marginpars overrun the page.

```
616 \def\Gm@checkmp{%
617 \ifGm@includemp\else
618 \@tempcnta\z@\@tempcntb\@ne
619 \if@twocolumn
620 \@tempcnta\@ne
621 \else
622 \if@reversemargin
623 \@tempcnta\@ne\@tempcntb\z@
624 \fi
625 \fi
626 \@tempdima\marginparwidth
627 \advance\@tempdima\marginparsep
628 \ifnum\@tempcnta=\@ne
629 \@tempdimc\@tempdima
```

```

630 \setlength\@tempdimb{\Gm@lmargin}%
631 \advance\@tempdimc-\@tempdimb
632 \ifdim\@tempdimc>\z@
633 \Gm@warning{The marginal notes overrun the paper.^^J
634 \spaces Add \the\@tempdimc\space and more to the left margin}%
635 \fi
636 \fi
637 \ifnum\@tempcntb=\@ne
638 \@tempdimc\@tempdima
639 \setlength\@tempdimb{\Gm@rmargin}%
640 \advance\@tempdimc-\@tempdimb
641 \ifdim\@tempdimc>\z@
642 \Gm@warning{The marginal notes overrun the paper.^^J
643 \spaces Add \the\@tempdimc\space and more to the right margin}%
644 \fi
645 \fi
646 \fi}%

```

`\Gm@adjustmp` The macro sets marginpar correction when `includemp` is set, which is used in `\Gm@process`. The variables `\Gm@wd@mp`, `\Gm@odd@mp` and `\Gm@even@mp` are set here. Note that `\Gm@even@mp` should be used only for twoside layout.

```

647 \def\Gm@adjustmp{%
648 \ifGm@includemp
649 \@tempdimb\marginparwidth
650 \advance\@tempdimb\marginparsep
651 \Gm@wd@mp\@tempdimb
652 \Gm@odd@mp\z@
653 \Gm@even@mp\z@
654 \if@twocolumn
655 \Gm@wd@mp2\@tempdimb
656 \Gm@odd@mp\@tempdimb
657 \Gm@even@mp\@tempdimb
658 \else
659 \if@reversemargin
660 \Gm@odd@mp\@tempdimb
661 \if@mparswitch\else
662 \Gm@even@mp\@tempdimb
663 \fi
664 \else
665 \if@mparswitch
666 \Gm@even@mp\@tempdimb
667 \fi
668 \fi
669 \fi
670 \fi}%

```

`\Gm@adjustbody` If the horizontal dimension of *body* is specified by user, `\Gm@width` is set properly here.

```

671 \def\Gm@adjustbody{
672 \ifGm@hbody
673 \ifx\Gm@width\@undefined
674 \ifx\Gm@hscale\@undefined
675 \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
676 \else
677 \Gm@defbylen{width}{\Gm@hscale\Gm@layoutwidth}%
678 \fi
679 \fi
680 \ifx\Gm@textwidth\@undefined\else
681 \setlength\@tempdima{\Gm@textwidth}%
682 \ifGm@includemp
683 \advance\@tempdima\Gm@wd@mp
684 \fi
685 \edef\Gm@width{\the\@tempdima}%
686 \fi
687 \fi

```

If the vertical dimension of *body* is specified by user, `\Gm@height` is set properly here.

```

688 \ifGm@vbody
689   \ifx\Gm@height\@undefined
690     \ifx\Gm@vscale\@undefined
691       \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
692     \else
693       \Gm@defbylen{height}{\Gm@vscale\Gm@layoutheight}%
694     \fi
695   \fi
696   \ifx\Gm@lines\@undefined\else

```

`\topskip` has to be adjusted so that the formula “ $\text{textheight} = (\text{lines} - 1) \times \text{baselineskip} + \text{topskip}$ ” to be correct even if large font sizes are specified by users. If `\topskip` is smaller than `\ht\strutbox`, then `\topskip` is set to `\ht\strutbox`.

```

697   \ifdim\topskip<\ht\strutbox
698     \setlength\@tempdima{\topskip}%
699     \setlength\topskip{\ht\strutbox}%
700     \Gm@warning{\noexpand\topskip was changed from \the\@tempdima\space
701       to \the\topskip}%
702   \fi
703   \setlength\@tempdima{\baselineskip}%
704   \multiply\@tempdima\Gm@lines
705   \addtolength\@tempdima{\topskip}%
706   \addtolength\@tempdima{-\baselineskip}%
707   \edef\Gm@textheight{\the\@tempdima}%
708 \fi
709 \ifx\Gm@textheight\@undefined\else
710   \setlength\@tempdima{\Gm@textheight}%
711   \ifGm@includehead
712     \addtolength\@tempdima{\headheight}%
713     \addtolength\@tempdima{\headsep}%
714   \fi
715   \ifGm@includefoot
716     \addtolength\@tempdima{\footskip}%
717   \fi
718   \edef\Gm@height{\the\@tempdima}%
719 \fi
720 \fi}%

```

`\Gm@process` The main macro processing the specified dimensions is defined.

```
721 \def\Gm@process{%
```

If `pass` is set, the original dimensions and switches are restored and process is ended here.

```

722 \ifGm@pass
723   \Gm@restore@org
724 \else
725   \Gm@@process
726 \fi}%

```

The main processing macro.

```

727 \def\Gm@@process{%
728   \Gm@expandlengths
729   \Gm@adjustpaper
730   \addtolength\Gm@layoutwidth{-\Gm@bindingoffset}%
731   \Gm@adjustmp
732   \Gm@adjustbody
733   \Gm@detail{h}{width}{lmargin}{rmargin}%
734   \Gm@detail{v}{height}{tmargin}{bmargin}%

```

The real dimensions are set properly according to the result of the auto-completion calculation.

```

735   \setlength\textwidth{\Gm@width}%
736   \setlength\textheight{\Gm@height}%
737   \setlength\topmargin{\Gm@tmargin}%
738   \setlength\oddsidemargin{\Gm@lmargin}%
739   \addtolength\oddsidemargin{-1\Gm@truedimen in}%

```

If `includemp` is set to true, `\textwidth` and `\oddsidemargin` are adjusted.

```
740 \ifGm@includemp
741   \advance\textwidth-\Gm@wd@mp
742   \advance\oddsidemargin\Gm@odd@mp
743 \fi
```

Determining `\evensidemargin`. In the twoside page layout, the right margin value `\Gm@rmargin` is used. If the marginal note width is included, `\evensidemargin` should be corrected by `\Gm@even@mp`.

```
744 \if@mparswitch
745   \setlength\evensidemargin{\Gm@rmargin}%
746   \addtolength\evensidemargin{-1\Gm@truedimen in}%
747   \ifGm@includemp
748     \advance\evensidemargin\Gm@even@mp
749   \fi
750 \else
751   \evensidemargin\oddsidemargin
752 \fi
```

The `bindingoffset` correction for `\oddsidemargin`.

```
753 \advance\oddsidemargin\Gm@bindingoffset
754 \addtolength\topmargin{-1\Gm@truedimen in}%
```

If the head of the page is included in *total body*, `\headheight` and `\headsep` are removed from `\textheight`, otherwise from `\topmargin`.

```
755 \ifGm@includehead
756   \addtolength\textheight{-\headheight}%
757   \addtolength\textheight{-\headsep}%
758 \else
759   \addtolength\topmargin{-\headheight}%
760   \addtolength\topmargin{-\headsep}%
761 \fi
```

If the foot of the page is included in *total body*, `\footskip` is removed from `\textheight`.

```
762 \ifGm@includefoot
763   \addtolength\textheight{-\footskip}%
764 \fi
```

If `heightrounded` is set, `\textheight` is rounded.

```
765 \ifGm@heightrounded
766   \setlength\@tempdima{\textheight}%
767   \addtolength\@tempdima{-\topskip}%
768   \@tempcnta\@tempdima
769   \@tempcntb\baselineskip
770   \divide\@tempcnta\@tempcntb
771   \setlength\@tempdimb{\baselineskip}%
772   \multiply\@tempdimb\@tempcnta
773   \advance\@tempdima-\@tempdimb
774   \multiply\@tempdima\tw@
775   \ifdim\@tempdima>\baselineskip
776     \addtolength\@tempdimb{\baselineskip}%
777   \fi
778   \addtolength\@tempdimb{\topskip}%
779   \textheight\@tempdimb
780 \fi
```

The paper width is set back by adding `\Gm@bindingoffset`.

```
781 \advance\oddsidemargin\Gm@layoutoffset%
782 \advance\evensidemargin\Gm@layoutoffset%
783 \advance\topmargin\Gm@layoutvoffset%
784 \addtolength\Gm@layoutwidth{\Gm@bindingoffset}%
785 }% end of \Gm@@process
```

`\Gm@detectdriver` The macro checks the typeset environment and changes the driver option if necessary. To make the engine detection more robust, the macro is rewritten with packages `ifpdf`, `ifvtex` and `ifxetex`.

```
786 \def\Gm@detectdriver{%
```

If the driver option is not specified explicitly, then driver auto-detection works.

```

787 \ifx\Gm@driver\@empty
788 \typeout{*geometry* driver: auto-detecting}%
\ifpdf is defined in ifpdf package in 'oberdiek' bundle.
789 \ifpdf
790 \ifx\pdfextension\undefined
791 \Gm@setdriver{pdftex}%
792 \else
793 \Gm@setdriver{luatex}%
794 \fi
795 \else
796 \Gm@setdriver{dvips}%
797 \fi
\ifvtex is defined in ifvtex package in 'oberdiek' bundle.

```

```

798 \ifvtex
799 \Gm@setdriver{vtex}%
800 \fi
\ifxetex is defined in ifxetex package written by Will Robertson.
801 \ifxetex
802 \Gm@setdriver{xetex}
803 \fi

```

When the driver option is set by the user, check if it is valid or not.

```

804 \else
805 \ifx\Gm@driver\Gm@xetex %%
806 \ifxetex\else
807 \Gm@warning{Wrong driver setting: 'xetex'; trying 'pdftex' driver}%
808 \Gm@setdriver{pdftex}
809 \fi
810 \fi
811 \ifx\Gm@driver\Gm@vtex
812 \ifvtex\else
813 \Gm@warning{Wrong driver setting: 'vtex'; trying 'dvips' driver}%
814 \Gm@setdriver{dvips}%
815 \fi
816 \fi
817 \fi
818 \ifx\Gm@driver\relax
819 \typeout{*geometry* detected driver: <none>}%
820 \else
821 \typeout{*geometry* detected driver: \Gm@driver}%
822 \fi}%

```

`\Gm@showparams` Prints the resulted parameters and dimensions to STDOUT if verbose is true. `\Gm@width` and `\Gm@height` are expanded to get the real size.

```

823 \def\Gm@showparams#1{%
824 \ifGm@verbose\expandafter\typeout\else\expandafter\wlog\fi
825 {\Gm@logcontent{#1}}}%
826 \def\Gm@showdim#1{* \string#1=\the#1^^J}%
827 \def\Gm@showbool#1{\@nameuse{ifGm#1}#1\space\fi}%

```

`\Gm@logcontent` The content of geometry parameters and native dimensions for the page layout.

```

828 \def\Gm@logcontent#1{%
829 *geometry* verbose mode - [ #1 ] result:^^J%
830 \ifGm@pass * pass: disregarded the geometry package!^^J%
831 \else
832 * driver: \if\Gm@driver<none>\else\Gm@driver\fi^^J%
833 * paper: \ifx\Gm@paper\undefined<default>\else\Gm@paper\fi^^J%
834 * layout: \ifGm@layout<custom>\else<same size as paper>\fi^^J%
835 \ifGm@layout
836 * layout(width,height): (\the\Gm@layoutwidth,\the\Gm@layoutheight)^^J%
837 \fi
838 * layoutoffset: (h,v)=(\the\Gm@layoutoffset,\the\Gm@layoutvoffset)^^J%

```

```

839 \@ifundefined{Gm@lines}{-* lines: \Gm@lines^^J}%
840 \@ifundefined{Gm@hmarginratio}{-* hratio: \Gm@hmarginratio^^J}%
841 \@ifundefined{Gm@vmarginratio}{-* vratio: \Gm@vmarginratio^^J}%
842 \ifdim\Gm@bindingoffset=\z@\else
843 * bindingoffset: \the\Gm@bindingoffset^^J\fi
844 * modes: %
845 \Gm@showbool{landscape}%
846 \Gm@showbool{includehead}%
847 \Gm@showbool{includefoot}%
848 \Gm@showbool{includemp}%
849 \if@twoside twoside\space\fi%
850 \if@mparswitch\else\if@twoside asymmetric\space\fi\fi%
851 \Gm@showbool{heightrounded}%
852 \ifx\Gm@truedimen\@empty\else truedimen\space\fi%
853 \Gm@showbool{showframe}%
854 \Gm@showbool{showcrop}%
855 ^^J%
856 * h-part:(L,W,R)=(\Gm@lmargin, \Gm@width, \Gm@rmargin)^^J%
857 * v-part:(T,H,B)=(\Gm@tmargin, \Gm@height, \Gm@bmargin)^^J%
858 \fi
859 \Gm@showdim{\paperwidth}%
860 \Gm@showdim{\paperheight}%
861 \Gm@showdim{\textwidth}%
862 \Gm@showdim{\textheight}%
863 \Gm@showdim{\oddsidemargin}%
864 \Gm@showdim{\evensidemargin}%
865 \Gm@showdim{\topmargin}%
866 \Gm@showdim{\headheight}%
867 \Gm@showdim{\headsep}%
868 \Gm@showdim{\topskip}%
869 \Gm@showdim{\footskip}%
870 \Gm@showdim{\marginparwidth}%
871 \Gm@showdim{\marginparsep}%
872 \Gm@showdim{\columnsep}%
873 * \string\skip\string\footins=\the\skip\footins^^J%
874 \Gm@showdim{\hoffset}%
875 \Gm@showdim{\voffset}%
876 \Gm@showdim{\mag}%
877 * \string\@twocolumn\if@twocolumn true\else false\fi^^J%
878 * \string\@twoside\if@twoside true\else false\fi^^J%
879 * \string\@mparswitch\if@mparswitch true\else false\fi^^J%
880 * \string\@reversemargin\if@reversemargin true\else false\fi^^J%
881 * (lin=72.27pt=25.4mm, 1cm=28.453pt)^^J}%

```

Macros for the page frames and cropmarks.

```

882 \def\Gm@cropmark(#1,#2,#3,#4){%
883 \begin{picture}(0,0)
884 \setlength\unitlength{1truemm}%
885 \linethickness{0.25pt}%
886 \put(#3,0){\line(#1,0){17}}%
887 \put(0,#4){\line(0,#2){17}}%
888 \end{picture}}%
889 \providecommand*{\vb@xt@{\vbox to}}%
890 \def\Gm@vrule{\vrule width 0.2pt height\textheight depth\z@}%
891 \def\Gm@hrule{\hrule height 0.2pt depth\z@ width\textwidth}%
892 \def\Gm@hruled{\hrule height\z@ depth0.2pt width\textwidth}%
893 \newcommand*{\Gm@vrules@mpi}{%
894 \hb@xt@\@tempdima{\llap{\Gm@vrule}\ignorespaces
895 \hskip \textwidth\Gm@vrule\hskip \marginparsep
896 \llap{\Gm@vrule}\hfil\Gm@vrule}}%
897 \newcommand*{\Gm@vrules@mpii}{%
898 \hb@xt@\@tempdima{\hskip-\marginparwidth\hskip-\marginparsep
899 \llap{\Gm@vrule}\ignorespaces
900 \hskip \marginparwidth\rlap{\Gm@vrule}\hskip \marginparsep
901 \llap{\Gm@vrule}\hskip\textwidth\rlap{\Gm@vrule}\hss}}%

```

```

902 \newcommand*{\Gm@pageframes}{%
903   \vb@xt@\z@{%
904     \ifGm@showcrop
905       \vb@xt@\z@{\vskip-1\Gm@truedimen in\vskip\Gm@layoutvoffset%
906         \hb@xt@\z@{\hskip-1\Gm@truedimen in\hskip\Gm@layouthoffset%
907           \vb@xt@\Gm@layoutheight{%
908             \let\protect\relax
909             \hb@xt@\Gm@layoutwidth{\Gm@cropmark(-1,1,-3,3)\hfil\Gm@cropmark(1,1,3,3)}}%
910             \vfil
911             \hb@xt@\Gm@layoutwidth{\Gm@cropmark(-1,-1,-3,-3)\hfil\Gm@cropmark(1,-1,3,-3)}}}%
912       \hss}%
913     \vss}%
914   \fi%
915 \ifGm@showframe
916   \if@twoside
917     \ifodd\count\z@
918       \let\@themargin\oddsidemargin
919     \else
920       \let\@themargin\evensidemargin
921     \fi
922   \fi
923   \moveright\@themargin%
924   \vb@xt@\z@{%
925     \vskip\topmargin\vbox@z@{\vss\Gm@hrule}%
926     \vskip\headheight\vbox@z@{\vss\Gm@hruled}%
927     \vskip\headsep\vbox@z@{\vss\Gm@hrule}%
928     \@tempdima\textwidth
929     \advance\@tempdima by \marginparsep
930     \advance\@tempdima by \marginparwidth
931     \if@mparswitch
932       \ifodd\count\z@
933         \Gm@vrules@mpi
934       \else
935         \Gm@vrules@mpii
936       \fi
937     \else
938       \Gm@vrules@mpi
939     \fi
940     \vb@xt@\z@{\vss\Gm@hrule}%
941     \vskip\footskip\vbox@z@{\vss\Gm@hruled}%
942     \vss}%
943   \fi%
944 }}%

```

`\ProcessOptionsKV` This macro can process class and package options using ‘key=value’ scheme. Only class options are processed with an optional argument ‘c’, package options with ‘p’, and both of them by default.

```

945 \def\ProcessOptionsKV{\@ifnextchar[%]
946   {\@ProcessOptionsKV}{\@ProcessOptionsKV[]}}%
947 \def\@ProcessOptionsKV[#1]#2{%
948   \let\@tempa\@empty
949   \@tempcnta\z@
950   \if#1p\@tempcnta\@ne\else\if#1c\@tempcnta\tw\fi\fi
951   \ifodd\@tempcnta
952     \edef\@tempa{\@optionlist{\@currname.\@current}}%
953   \else
954     \@for\CurrentOption:=\@classoptionslist\do{%
955       \@ifundefined{KV@#2@\CurrentOption}%
956       {}{\edef\@tempa{\@tempa,\CurrentOption,}}}%
957     \ifnum\@tempcnta=\z@
958       \edef\@tempa{\@tempa,\@optionlist{\@currname.\@current}}%
959     \fi
960   \fi
961   \edef\@tempa{\noexpand\setkeys{#2}{\@tempa}}%
962   \@tempa
963   \AtEndOfPackage{\let\@unprocessedoptions\relax}}%

```

```

964 \def\Gm@setkeys{\setkeys{Gm}}%
\Gm@processconf \ExecuteOptions is replaced with \Gm@setkey to make it possible to deal with 'key='(value)' as
its argument.
965 \def\Gm@processconfig{%
966 \let\Gm@origExecuteOptions\ExecuteOptions
967 \let\ExecuteOptions\Gm@setkeys
968 \InputIfFileExists{geometry.cfg}{-}{-}
969 \let\ExecuteOptions\Gm@origExecuteOptions}%

The original page layout before loading geometry is saved here. \Gm@restore@org is defined
here for reset option.
970 \Gm@save
971 \edef\Gm@restore@org{\Gm@restore}%
972 \Gm@initall

Processing config file.
973 \Gm@processconfig

The optional arguments to \documentclass are processed here.
974 \ProcessOptionsKV[c]{Gm}%

Paper dimensions given by class default are stored.
975 \Gm@setdefaultpaper

The optional arguments to \usepackage are processed here.
976 \ProcessOptionsKV[p]{Gm}%

Actual settings and calculation for layout dimensions are processed.
977 \Gm@process

\AtBeginDocument The processes for verbose, showframe and drivers are added to \AtBeginDocument. \Gm@restore@org
is redefined here with the paper size specified in the preamble for \newgeometry to use it. This
should be done before magnifying the paper size with \mag because the layout calculation would
be affected by changing the paper size.
978 \AtBeginDocument{%
979 \Gm@savelength{paperwidth}%
980 \Gm@savelength{paperheight}%
981 \edef\Gm@restore@org{\Gm@restore}%

The original paper size is used if resetpaper.
982 \ifGm@resetpaper
983 \edef\Gm@pw{\Gm@orgpw}%
984 \edef\Gm@ph{\Gm@orgph}%
985 \else
986 \edef\Gm@pw{\the\paperwidth}%
987 \edef\Gm@ph{\the\paperheight}%
988 \fi

If pass is not set, the paper size is multiplied according to the specified mag.
989 \ifGm@pass\else
990 \ifnum\mag=\@m\else
991 \Gm@magtooffset
992 \divide\paperwidth\@m
993 \multiply\paperwidth\the\mag
994 \divide\paperheight\@m
995 \multiply\paperheight\the\mag
996 \fi
997 \fi

Checking the driver options.
998 \Gm@detectdriver

If xetex and \pdfpagewidth is defined, \pdfpagewidth and \pdfpageheight would be set.
999 \ifx\Gm@driver\Gm@xetex
1000 \ifundefined{pdfpagewidth}{-}{-}
1001 \setlength\pdfpagewidth{\Gm@pw}%
1002 \setlength\pdfpageheight{\Gm@ph}}%

```

```

1003 \ifnum\mag=\@m\else
1004 \ifx\Gm@truedimen\Gm@true
1005 \setlength\paperwidth{\Gm@pw}%
1006 \setlength\paperheight{\Gm@ph}%
1007 \fi
1008 \fi
1009 \fi

```

If pdftex is set to true, pdf-commands are set properly. To avoid pdftex magnification problem, \pdfhorigin and \pdfvorigin are adjusted for \mag.

```

1010 \ifx\Gm@driver\Gm@pdftex
1011 \@ifundefined{pdfpagewidth}{-}{%
1012 \setlength\pdfpagewidth{\Gm@pw}%
1013 \setlength\pdfpageheight{\Gm@ph}}%
1014 \ifnum\mag=\@m\else
1015 \@tempdima=\mag sp%
1016 \@ifundefined{pdfhorigin}{-}{%
1017 \divide\pdfhorigin\@tempdima
1018 \multiply\pdfhorigin\@m
1019 \divide\pdfvorigin\@tempdima
1020 \multiply\pdfvorigin\@m}%
1021 \ifx\Gm@truedimen\Gm@true
1022 \setlength\paperwidth{\Gm@pw}%
1023 \setlength\paperheight{\Gm@ph}%
1024 \fi
1025 \fi
1026 \fi

```

If luatex is set to true, pdf-commands are set properly. To avoid luatex magnification problem, \horigin and \vorigin are adjusted for \mag.

```

1027 \ifx\Gm@driver\Gm@luatex
1028 \setlength\pagewidth{\Gm@pw}%
1029 \setlength\pageheight{\Gm@ph}%
1030 \ifnum\mag=\@m\else
1031 \@tempdima=\mag sp
1032 \edef\Gm@horigin{\pdfvariable horigin}%
1033 \edef\Gm@vorigin{\pdfvariable vorigin}%
1034 \divide\Gm@horigin\@tempdima
1035 \multiply\Gm@horigin\@m
1036 \divide\Gm@vorigin\@tempdima
1037 \multiply\Gm@vorigin\@m
1038 \ifx\Gm@truedimen\Gm@true
1039 \setlength\paperwidth{\Gm@pw}%
1040 \setlength\paperheight{\Gm@ph}%
1041 \fi
1042 \fi
1043 \fi

```

With VT_EX environment, VT_EX variables are set here.

```

1044 \ifx\Gm@driver\Gm@vtex
1045 \@ifundefined{mediawidth}{-}{%
1046 \mediawidth=\paperwidth
1047 \mediaheight=\paperheight}%
1048 \ifvtexdvi
1049 \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1050 \fi
1051 \fi

```

If dvips or dvipdfm is specified, paper size is embedded in dvi file with \special. For dvips, a landscape correction is added because a landscape document converted by dvips is upside-down in PostScript viewers.

```

1052 \ifx\Gm@driver\Gm@dvips
1053 \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1054 \ifx\Gm@driver\Gm@dvips\ifGm@landscape
1055 \AtBeginDvi{\special{! /landplus90 true store}}%
1056 \fi\fi

```

If dvipdfm is specified and atbegshi package in ‘oberdiek’ bundle is loaded, \AtBeginShipoutFirst is used instead of \AtBeginDvi for compatibility with hyperref and dvipdfm program.

```

1057 \else\ifx\Gm@driver\Gm@dvipdfm
1058   \ifcase\ifx\AtBeginShipoutFirst\relax\@ne\else
1059     \ifx\AtBeginShipoutFirst\@undefined\@ne\else\z\fi\fi
1060     \AtBeginShipoutFirst{\special{papersize=\the\paperwidth,\the\paperheight}}%
1061   \or
1062     \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1063   \fi
1064 \fi\fi

```

Page frames are shipped out when showframe=true, cropmarks for showcrop=true on each page. The atbegshi package is used for overloading \shipout.

```

1065 \@tempwafalse
1066 \ifGm@showframe
1067   \@tempwatrue
1068 \else\ifGm@showcrop
1069   \@tempwatrue
1070 \fi\fi
1071 \if@tempswa
1072   \RequirePackage{atbegshi}%
1073   \AtBeginShipout{\setbox\AtBeginShipoutBox=\vbox{%
1074     \baselineskip\z@skip\lineskip\z@skip\lineskiplimit\z@
1075     \Gm@pageframes\box\AtBeginShipoutBox}}%
1076 \fi

```

The layout dimensions for \restoregeometry are saved at the end of the \AtBeginDocument.

```

1077 \Gm@save
1078 \edef\Gm@restore@pkg{\Gm@restore}%

```

The package checks whether or not the marginpars overrun the page, if verbose and unless pass.

```

1079 \ifGm@verbose\ifGm@pass\else\Gm@checkmp\fi\fi

```

\Gm@showparams puts the resulting parameters and dimensions into the log file. With verbose, they are shown on the terminal as well.

```

1080 \Gm@showparams{preamble}%

```

The following lines free the memories no longer needed.

```

1081 \let\Gm@pw\relax
1082 \let\Gm@ph\relax
1083 }% end of \AtBeginDocument

```

`\geometry` The macro `\geometry` can be called multiple times in the preamble (before `\begin{document}`).

```

1084 \newcommand{\geometry}[1]{%
1085   \Gm@clean
1086   \setkeys{Gm}{#1}%
1087   \Gm@process}%
1088 \@onlypreamble\geometry

```

`\Gm@changelayout` The macro, which can be called from `\newgeometry`, `\restoregeometry` and `\loadgeometry`, changes the layout in the middle of the document.

```

1089 \DeclareRobustCommand\Gm@changelayout{%
1090   \setlength{\@colht}{\textheight}
1091   \setlength{\@colroom}{\textheight}%
1092   \setlength{\vsize}{\textheight}
1093   \setlength{\columnwidth}{\textwidth}%
1094   \if@twocolumn%
1095     \advance\columnwidth-\columnsep
1096     \divide\columnwidth\tw@%
1097   \@firstcolumntrue%
1098   \fi%
1099   \setlength{\hsize}{\columnwidth}%
1100   \setlength{\linewidth}{\hsize}}%

```

`\newgeometry` The macro `\newgeometry`, which changes the layout, can be used only in the document. It would reset the options specified in the preamble except for paper size options and `\mag`.

```

1101 \newcommand{\newgeometry}[1]{%
1102   \clearpage
1103   \Gm@restore@org
1104   \Gm@initnewgm
1105   \Gm@newgmtrue
1106   \setkeys{Gm}{#1}%
1107   \Gm@newgmfalse
1108   \Gm@process
1109   \ifnum\mag=\@m\else\Gm@magtooffset\fi
1110   \Gm@changelayout
1111   \Gm@showparams{newgeometry}}%

```

`\restoregeometry` The macro restores the resulting layout specified in the preamble, namely the first-page layout right after `\begin{document}`.

```

1112 \newcommand{\restoregeometry}{%
1113   \clearpage
1114   \Gm@restore@pkg
1115   \Gm@changelayout}%

```

`\savegeometry` The macro saves the layout with the name specified with the argument. The saved layout can be loaded with `\loadgeometry{<name>}`.

```

1116 \newcommand*{\savegeometry}[1]{%
1117   \Gm@save
1118   \expandafter\edef\csname Gm@restore@@#1\endcsname{\Gm@restore}}%

```

`\loadgeometry` The macro loads the layout saved with `\savegeometry{<name>}`. If the name is not found, the macro would warn it and do nothing for the layout.

```

1119 \newcommand*{\loadgeometry}[1]{%
1120   \clearpage
1121   \@ifundefined{Gm@restore@@#1}{%
1122     \PackageError{geometry}{%
1123       \string\loadgeometry : name '#1' undefined}{%
1124         The name '#1' should be predefined with \string\savegeometry}%
1125     }{\@nameuse{Gm@restore@@#1}%
1126     \Gm@changelayout}}%
1127 \end{package}

```

12 Config file

In the configuration file `geometry.cfg`, one can use `\ExecuteOptions` to set the site or user default settings.

```

1128 (*config)
1129 %<<SAVE_INTACT
1130
1131 % Uncomment and edit the line below to set default options.
1132 %\ExecuteOptions{a4paper}
1133
1134 %SAVE_INTACT
1135 \end{config}

```

13 Sample file

Here is a sample document for the geometry package.

```

1136 (*samples)
1137 %<<SAVE_INTACT
1138 \documentclass[12pt]{article}% uses letterpaper by default
1139 % \documentclass[12pt,a4paper]{article}% for A4 paper
1140 %-----
1141 % Edit and uncomment one of the settings below
1142 %-----
1143 % \usepackage{geometry}
1144 % \usepackage[centering]{geometry}

```

```

1145 % \usepackage[width=10cm,vscale=.7]{geometry}
1146 % \usepackage[margin=1cm, papersize={12cm,19cm}, resetpaper]{geometry}
1147 % \usepackage[margin=1cm,includeheadfoot]{geometry}
1148 \usepackage[margin=1cm,includeheadfoot,includemp]{geometry}
1149 % \usepackage[margin=1cm,bindingoffset=1cm,twoside]{geometry}
1150 % \usepackage[hmarginratio=2:1, vmargin=2cm]{geometry}
1151 % \usepackage[hscale=0.5,twoside]{geometry}
1152 % \usepackage[hscale=0.5,asymmetric]{geometry}
1153 % \usepackage[hscale=0.5,heightrounded]{geometry}
1154 % \usepackage[left=1cm,right=4cm,top=2cm,includefoot]{geometry}
1155 % \usepackage[lines=20,left=2cm,right=6cm,top=2cm,twoside]{geometry}
1156 % \usepackage[width=15cm, marginparwidth=3cm, includemp]{geometry}
1157 % \usepackage[hdivide={1cm,2cm}, vdivide={3cm,8in,}, nohead]{geometry}
1158 % \usepackage[headsep=20pt, head=40pt,foot=20pt,includeheadfoot]{geometry}
1159 % \usepackage[text={6in,8in}, top=2cm, left=2cm]{geometry}
1160 % \usepackage[centering,includemp,twoside,landscape]{geometry}
1161 % \usepackage[mag=1414,margin=2cm]{geometry}
1162 % \usepackage[mag=1414,margin=2truecm,truedimen]{geometry}
1163 % \usepackage[a5paper, landscape, twocolumn, twoside,
1164 % left=2cm, hmarginratio=2:1, includemp, marginparwidth=43pt,
1165 % bottom=1cm, foot=.7cm, includefoot, textheight=11cm, heightrounded,
1166 % columnsep=1cm,verbose]{geometry}
1167 %-----
1168 % No need to change below
1169 %-----
1170 \geometry{verbose,showframe}% the options appended.
1171 \usepackage{lipsum}% for dummy text of 150 paragraphs
1172 \newcommand\mynote{\marginpar[\raggedright
1173 A sample margin note in the left side.]}
1174 {\raggedright A sample margin note.}}%
1175 \newcommand\myfootnote{\footnote{This is a sample footnote text.}}
1176 \begin{document}
1177 \lipsum[1-2]\mynote\lipsum[3-4]\mynote
1178 \lipsum[5-11]\mynote\lipsum[12]\myfootnote
1179 \lipsum[13-22]\mynote\lipsum[23-32]
1180 \end{document}
1181 %SAVE_INTACT
1182 </samples>

```