

Das geometry Paket

Hideo Umeki

latexgeometry@gmail.com

The translation was create by Hans-Martin Haase (hmhaase@pclinuxosusers.de)

2010/09/12 v5.6

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 T_EX 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 X_YL_AT_EX 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 Seitenlayoutsmaße, 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) (optional `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.

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

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

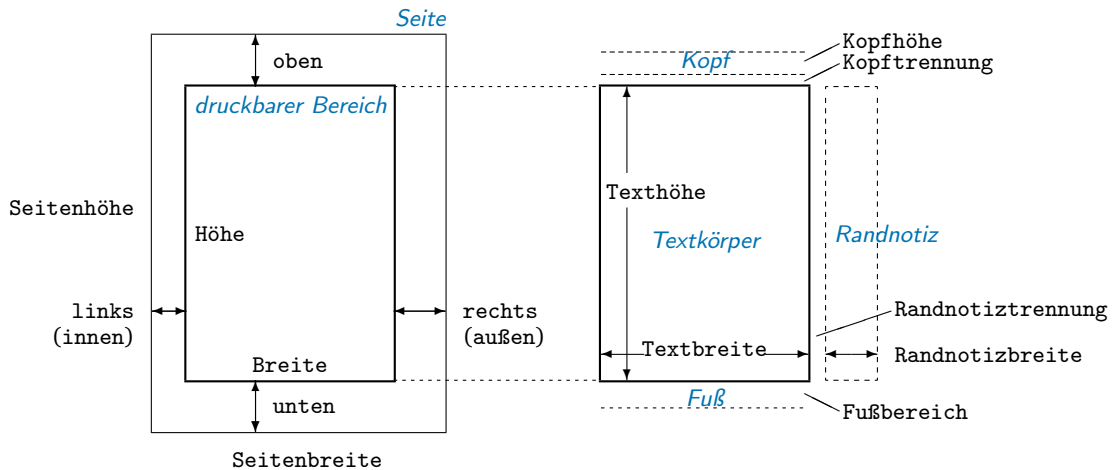


Abbildung 1: Namen von Maßen, die im geometry Paket genutzt werden: `width = textwidth` (Textbreite) und `height = textheight` (Texthöhe) sind Standard. `left` (links), `right` (rechts), `top` (oben) und `bottom` (unten) sind Ränder. Wenn die Ränder auf der Rückseite durch die `twoside` (Zweiseitigkeit) 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`.

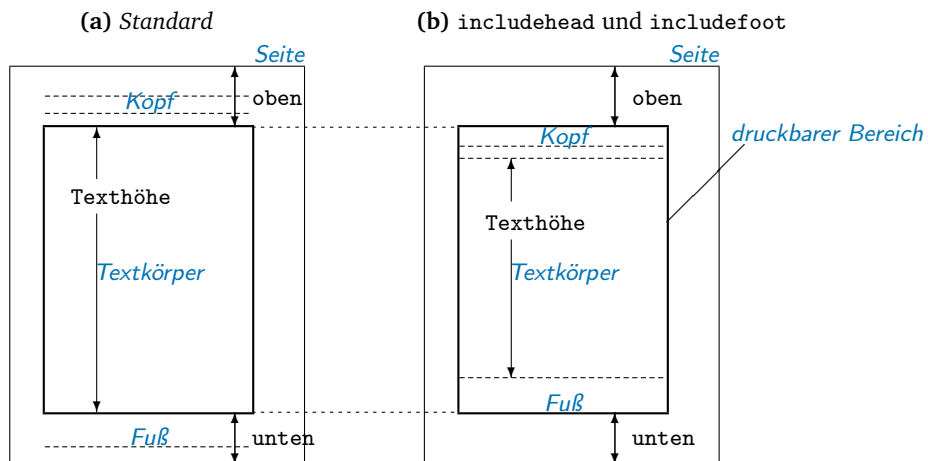


Abbildung 2: `includehead` und `includefoot` schließen den Kopf und den Fuß mit in den *total body* (druckbaren Bereich) ein. (a) `height = textheight` (Standard). (b) `height = textheight + headheight + headsep + footskip`, wenn `includehead` und `includefoot` aktiv. Wenn der obere und untere Rand angegeben 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 $\text{width}:=\text{textwidth}$, während marginparsep und marginparwidth nur in der width (Breite) enthalten sind, wenn die includeemp Option auf true gesetzt ist. In Gleichung (4) ist standardmäßig $\text{height}:=\text{textheight}$. 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

$$L = a + b + c \quad (5)$$

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’.

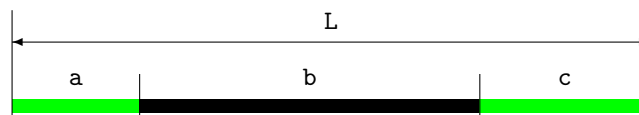


Abbildung 3: Ein einfaches Modell der Seitenmaße.

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[textheight=20\baselineskip+10pt]{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 einzelнем 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` (Papiernamen)
gibt die Papiergröße anhand des Namens an. `paper=\langle \text{Papiernamen} \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`,
`ansipaper`, `ansipaper`, `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.

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

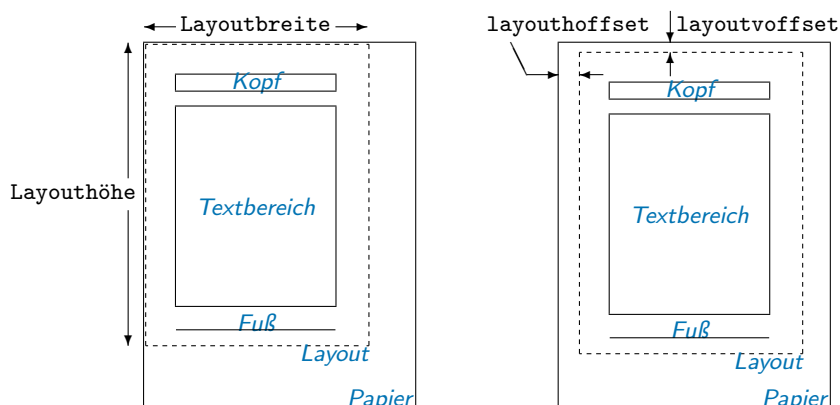


Abbildung 4: Die Maße sind mit denen der Layoutgröße verwandt. Beachte, dass die Layoutgrößenvorgaben zu der Größe des Papiers korrespondieren, sodass sie die meisten layoutbezogenen Optionen nicht explizit angeben müssen.

5.3 Größe des Textbereichs

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

<code>hscale</code>	Verhältnis der Breite des <i>druckbaren Bereichs</i> zur <code>\paperwidth</code> (Papierbreite). <code>hscale=\langle h-scale \rangle</code> , z. B. <code>hscale=0.8</code> ist äquivalent zu <code>width=0.8\paperwidth</code> (Papierbreite). (0.7 ist der Standard)
<code>vscale</code>	Verhältnis der Höhe des <i>druckbaren Bereichs</i> zur <code>\paperheight</code> (Papierhöhe), z. B. <code>vscale=\langle v-scale \rangle</code> . (0.7 ist der Standard) <code>vscale=0.9</code> ist äquivalent zu <code>height=0.9\paperheight</code> (Papierhöhe).
<code>scale</code>	Verhältnis des <i>druckbaren Bereichs</i> zum Papier. <code>scale=\{\langle h-scale \rangle,\langle v-scale \rangle\}</code> oder <code>scale=\langle scale \rangle</code> . (0.7 ist der Standard)
<code>width</code> <code>totalwidth</code>	Breite des <i>druckbaren Bereichs</i> . <code>width=\langle\text{Länge}\rangle</code> oder <code>totalwidth=\langle\text{Länge}\rangle</code> . Dieses Maß korrespondiert zur <code>textwidth</code> (Textbreite), aber wenn <code>includemp</code> auf <code>true</code> 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.

height totalheight	Höhe des <i>druckbaren Bereichs</i> , wobei Kopf und Fuß standardmäßig ausgeschlossen sind. Wenn includehead oder includefoot gesetzt ist, dann beinhaltet die height (Höhe) den Kopf oder den Fuß der Seite sowie die textheight (Texthöhe). height= $\langle\text{Länge}\rangle$ oder totalheight= $\langle\text{Länge}\rangle$. Falls beide, textheight (Texthöhe) und height (Höhe), angegeben sind, wird die height (Höhe) ignoriert werden.
total	Breite und Höhe des <i>druckbaren Bereichs</i> . total= $\{\langle\text{Breite}\rangle, \langle\text{Höhe}\rangle\}$ oder total= $\langle\text{Länge}\rangle$.
textwidth	gibt die \textwidth (Textbreite), die Breite des <i>Körpers</i> (der Textbereich) an. textwidth= $\langle\text{Länge}\rangle$.
textheight	gibt die \textheight (Texthöhe), die Höhe des <i>Körpers</i> (der Textkörper) an. textheight= $\langle\text{Länge}\rangle$.
text body	gibt beides, die \textwidth (Textbreite) und \textheight (Texthöhe) des Körpers der Seite an. body= $\{\langle\text{Breite}\rangle, \langle\text{Höhe}\rangle\}$ oder text= $\langle\text{Länge}\rangle$.
lines	erlaubt dem Benutzer, die \textheight (Texthöhe) durch die Anzahl der Zeilen anzugeben. lines= $\langle\text{ganzzahliger Wert}\rangle$.
includehead	schließt den Kopf der Seite, die \headheight (Kopfhöhe) und \headsep (Kopftrennung) in den <i>druckbaren Bereich</i> ein. Diese Option ist in den Voreinstellungen durch false deaktiviert. Es ist das Gegenteil von ignorehead. Betrachten sie auch Abbildung 2 und Abbildung 5.
includefoot	schließt den Fuß der Seite, \footskip (Fußbereich), in den <i>druckbaren Bereich</i> ein. Es ist das Gegenteil von ignorefoot. Die Voreinstellung ist false (deaktiviert). Betrachten sie auch Abbildung 2 und Abbildung 5.
includeheadfoot	setzt beides, includehead und includefoot, auf true, dies ist das Gegenteil von ignoreheadfoot. Betrachten sie auch Abbildung 2 und Abbildung 5.
includemp	schließt die Randnotizen, die \marginparwidth (Randnotizbreite) und \marginparsep (Randnotiztrennung) in den <i>Körper</i> ein, wenn die horizontalen Maße berechnet werden.
includeall	setzt beides, includeheadfoot und includemp auf true (aktiv). Betrachten sie auch Abbildung 5.
ignorehead	missachtet den Kopf der Seite, headheight (Kopfhöhe) und headsep (Kopftrennung), sind ausschlaggebend für das vertikale Layout, aber diese ändern dessen Längen nichts. Es ist äquivalent zu includehead=false. Diese Option ist mit true voreingestellt. Lesen sie auch includehead.
ignorefoot	missachtet den Fuß der Seite, der footskip (Fußbereich), ist ausschlaggebend für das vertikale Layout, aber dies ändert die Länge nicht. Diese Voreinstellung der Option ist true. Lesen sie auch includefoot.
ignoreheadfoot	setzt beides, ignorehead und ignorefoot auf true. Lesen sie auch includeheadfoot.
ignoremp	vernachlässigt die Randnotizen bei der Berechnung der horizontalen Ränder (der Standard ist true). Wenn die Randnotizen die Seite überfüllen, dann wird eine Warnmeldung angezeigt werden, wenn verbose=true aktiviert ist. Sehen sie auch includemp und Abbildung 5.
ignoreall	setzt beides, ignoreheadfoot und ignoremp auf true. Lesen sie auch includeall.
heightrounded	Diese Option rundet die \textheight (Texthöhe), um n -Mal mehr (n : ist eine ganze Zahl) \baselineskip plus \topskip, um “underfull vbox” in manchen Fällen zu vermeiden. Zum Beispiel, wenn die \textheight (Texthöhe) 486pt ist, mit \baselineskip 12pt und \topskip 10pt, dann $(39 \times 12\text{pt} + 10\text{pt}) = 478\text{pt} < 486\text{pt} < 490\text{pt} (= 40 \times 12\text{pt} + 10\text{pt}),$ als Ergebnis wird die \textheight (Texthöhe) auf 490 pt aufgerundet. heightrounded=false 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 0pt. 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.

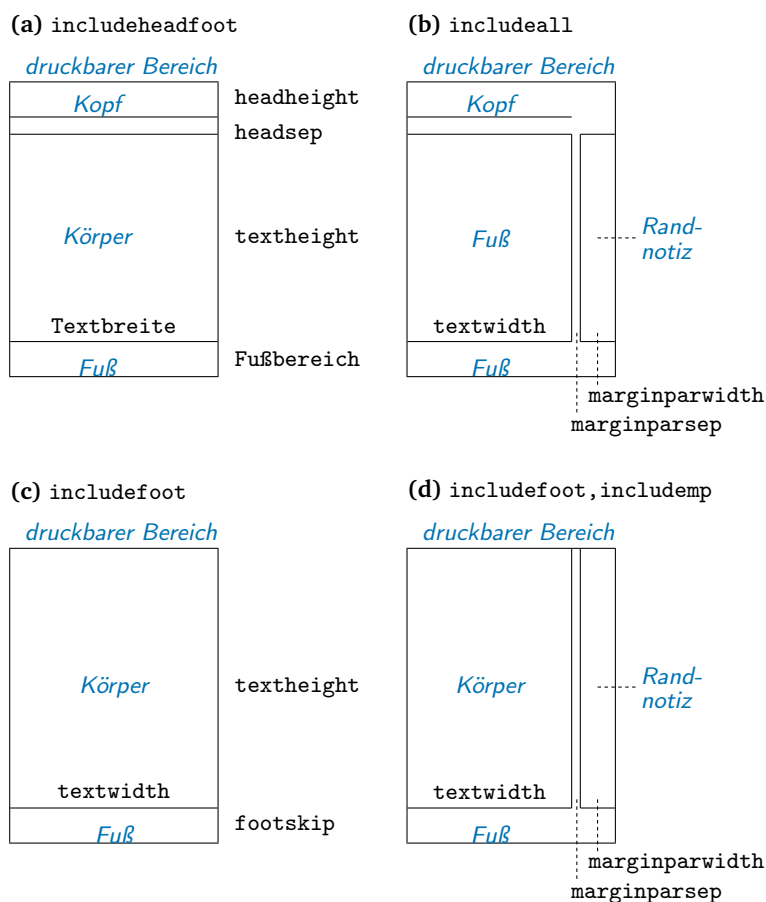


Abbildung 5: Musterlayouts für den *druckbaren Bereich* mit verschiedenen Veränderungen. (a) `includeheadfoot`, (b) `includeall`, (c) `includefoot` und (d) `includefoot,includemp`. Wenn `reversemp` auf `true` gesetzt ist, dann werden die Orte für die Randnotizen auf jeder Seite getauscht. Die Option `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={linker Rand, rechter Rand} oder hmargin= <i>Länge</i> .
vmargin	oberer und unterer Rand. vmargin={oberer Rand, unterer Rand} oder vmargin= <i>Länge</i> .
margin	margin={A, B} ist äquivalent zu hmargin={A, B} und vmargin={A, B}. margin=A wird automatisch erweitert zu hmargin=A und vmargin=A.
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={horizontales Verhältnis, vertikales Verhältnis} 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> .

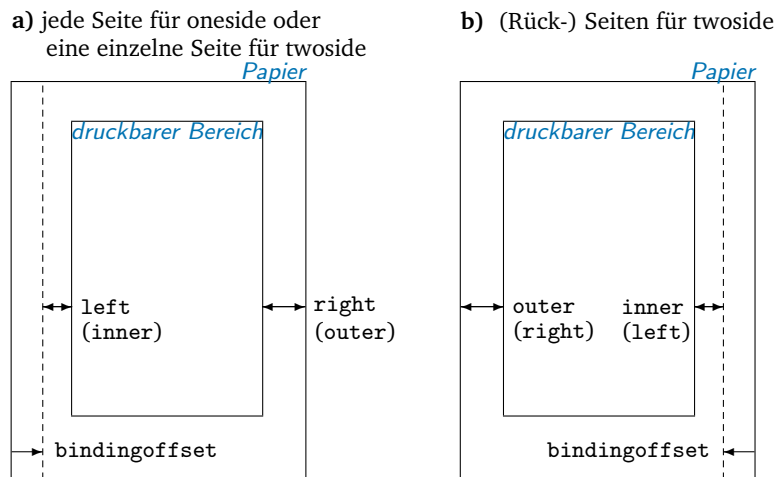


Abbildung 6: Die Option `bindingoffset` fügt eine angegebene Länge zum inneren Rand hinzu. Beachten sie, dass die `twoside` Option die horizontalen Ränder und die Randnotizen zusammen mit `bindingoffset` auf jeder Seite tauscht (siehe **b**)), aber die `asymmetric` Option unterdrücke den Tausch der Ränder 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`, `xetex` und `vtex`. Sie können auch `dvipdfm` für `dvipdfmx` einstellen und `xdvipdfmx`. `pdftex` für `pdflatex`, und `vtex` für die \TeX 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.

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

[†] <code>verbose</code>	zeigt die Parameterergebnisse im Terminal an. <code>verbose=false</code> (Standard) schreibt diese nur in die Log-Datei.
[†] <code>reset</code>	setzt die Layoutmaße zurück und nimmt die Einstellungen an, die vor dem Laden von <code>geometry</code> vorgegeben waren. Die Optionen, die in <code>geometry.cfg</code> angegeben sind, werden auch entfernt. Beachten sie, dass dies nicht pass und mag mit <code>truedimen</code> zurücksetzen kann. <code>reset=false</code> hat keine Auswirkungen und kann ein vorheriges <code>reset(=true)</code> , falls es dies gibt, nicht aufheben. Zum Beispiel, wenn sie <pre> \documentclass[landscape]{article} \usepackage[twoside,reset,left=2cm]{geometry} </pre> eingeben, mit <code>\ExecuteOptions{scale=0.9}</code> in <code>geometry.cfg</code> , dann bleiben <code>landscape</code> und <code>left=2cm</code> wirksam und <code>scale=0.9</code> und <code>twoside</code> sind unwirksam.
[†] <code>mag</code>	setzt den Vergrößerungswert (<code>\mag</code>) und modifiziert automatisch <code>\hoffset</code> und <code>\voffset</code> entsprechend der Vergrößerung. <code>mag=<Wert></code> . Beachten sie, dass <code><Wert></code> ein ganzzahliger Wert sein sollte, wobei 1000 die Normalgröße ist. Ein Beispiel: <code>mag=1414</code> mit <code>a4paper</code> bietet eine erweiterte Druckgröße in <code>a3paper</code> , 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 truedimen Option.

- † truedimen 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, truedimen” 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, truedimen”.
- † 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 T_EX 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*.

Priorität: niedrig \rightarrow hoch

$$\left\{ \begin{array}{l} \text{hscale} \\ \text{vscale} \\ \text{scale} \end{array} \right\} < \left\{ \begin{array}{l} \text{width} \\ \text{height} \\ \text{total} \end{array} \right\} < \left\{ \begin{array}{l} \text{textwidth} \\ \text{textheight} \\ \text{text} \end{array} \right\} < \text{lines}.$$

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 $0.7 \times \textit{Papier}$)
- `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.

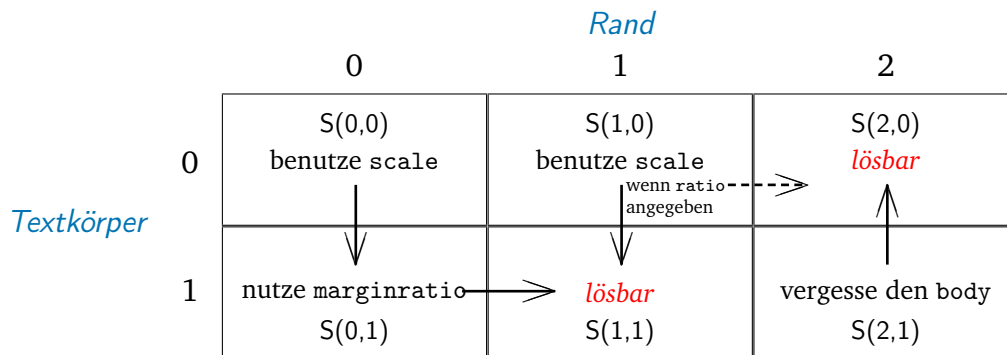


Abbildung 7: Die Spezifikationen S(0,0) zu S(2,1) und die Komplettierungsregeln (Pfeile). Spalten- und Zeilennummern bezeichnen die Anzahl der explizit angegebenen Längen für den Rand und den Körper. S(m,b) bezeichnet eine Spezifikation mit einer Menge an Zahlen (Rand, Körper) = (m, b).

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:

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

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

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

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

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

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 14cm 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 Dokuments 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)

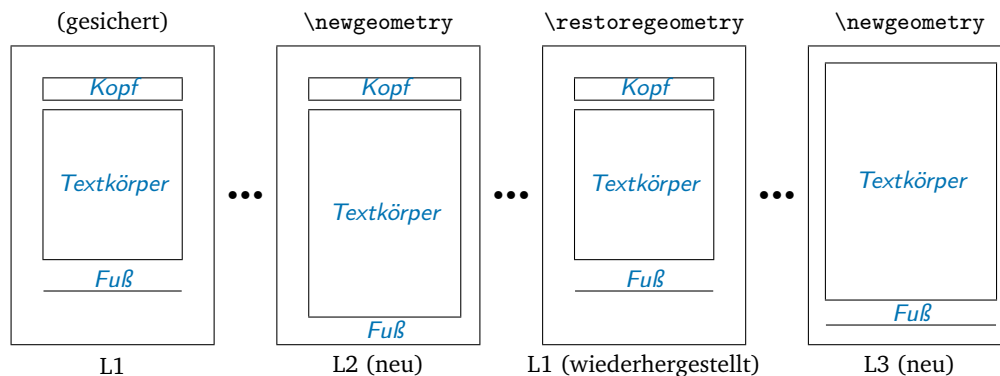
  \restoregeometry

  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

1. 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`

2. 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, $\text{width} = 0.7 \times \text{layoutwidth}$ ($= 0.7 \times \text{paperwidth}$ standardmäßig).

3. 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`

4. 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`

- 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`.

- 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`

- 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`

- Ein Bildschirmgrößenslayout, verwendbar für Präsentationen mit dem PC oder dem Beamer:

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

- 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 $\text{mag} \neq 1000$ und `trueedimen`, `paperwidth` und `paperheight` ist die Anzeige im verbose-Modus verschieden von der wirklichen Größe der resultierenden PDF. Die PDF ist aber korrekt.
- Mit $\text{mag} \neq 1000$, *no trueedimen* und `hyperref` sollte `hyperref` vor `geometry` geladen werden. Andernfalls wird die PDF-Größe fehlerhaft.
- Mit dem `crop` Paket und $\text{mag} \neq 1000$ 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 Flenner, 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 <*package>
This package requires the following packages: keyval, ifpdf, ifvtex and ifxetex.
2 \RequirePackage{keyval}%
3 \RequirePackage{ifpdf}%
4 \RequirePackage{ifvtex}%
5 \RequirePackage{ifxetex}%
    Internal switches are declared here.
6 \newif\ifGm@verbose
7 \newif\ifGm@landscape
8 \newif\ifGm@swap@papersize
9 \newif\ifGm@includehead
10 \newif\ifGm@includefoot
11 \newif\ifGm@includemp
12 \newif\ifGm@hbody
13 \newif\ifGm@vbody
14 \newif\ifGm@heightrounded
15 \newif\ifGm@showframe
16 \newif\ifGm@showcrop
17 \newif\ifGm@pass
18 \newif\ifGm@resetpaper
19 \newif\ifGm@layout
20 \newif\ifGm@newgm

\Gm@cnth The counters for horizontal and vertical partitioning patterns.
\Gm@cntv 21 \newcount\Gm@cnth
22 \newcount\Gm@cntv

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

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

\Gm@wd@mp Correction lengths for \textwidth, \oddsidemargin and \evensidemargin in includemp mode.
\Gm@odd@mp 25 \newdimen\Gm@wd@mp
\Gm@even@mp 26 \newdimen\Gm@odd@mp
27 \newdimen\Gm@even@mp

\Gm@layoutwidth The dimensions for the layout area.
\Gm@layoutheight 28 \newdimen\Gm@layoutwidth
\Gm@layoutoffset 29 \newdimen\Gm@layoutheight
\Gm@layoutvoffset 30 \newdimen\Gm@layoutoffset
31 \newdimen\Gm@layoutvoffset

\Gm@dimlist The token in which LATEX native dimensions can be stored.
32 \newtoks\Gm@dimlist

\Gm@warning The macro to print warning messages.
33 \def\Gm@warning#1{\PackageWarningNoLine{geometry}{#1}}%

\ifGm@preamble The macro executes the option given as an argument only if it's specified in the preamble, as the
options of \usepackage and/or the argument of \geometry. Otherwise, the macro would print the
warning message and ignores the option setting.
34 \def\ifGm@preamble#1{%
35   \ifGm@newgm
36     \Gm@warning{'#1': not available in '\string\newgeometry'; skipped}%
37   \else
38     \expandafter\@firstofone
39   \fi}%

```


`\Gm@Dhratio` The default values for the horizontal and vertical *marginalratio* are defined. `\Gm@Dhratio` 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`.

```

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

```

`\Gm@Dhscale` The default values for the horizontal and vertical *scale* are defined with 0.7.

```

43 \def\Gm@Dhscale{0.7}%
44 \def\Gm@Dvscale{0.7}%

```

`\Gm@dvi` The driver names.

```

45 \def\Gm@dvi{dvips}%
46 \def\Gm@dvi{dvipdfm}%
47 \def\Gm@dvi{pdftex}%
48 \def\Gm@dvi{xetex}%
49 \def\Gm@dvi{vtex}%

```

`\Gm@true` The macros for true and false.

```

50 \def\Gm@true{true}%
51 \def\Gm@false{false}%

```

`\Gm@orgpw` These macros keep original paper (media) size intact.

```

52 \edef\Gm@orgpw{\the\paperwidth}%
53 \edef\Gm@orgph{\the\paperheight}%

```

`\Gm@savelength` The macro saves the specified length to `\Gm@restore`.

```

54 \def\Gm@savelength#1{%
55   \g@addto@macro\Gm@restore{\expandafter\noexpand\expandafter\csname
56     #1\endcsname\expandafter=\expandafter\the\csname #1\endcsname\relax}}%

```

`\Gm@saveboolean` The macro saves the specified boolean to `\Gm@restore`.

```

57 \def\Gm@saveboolean#1{%
58   \csname if#1\endcsname
59     \g@addto@macro\Gm@restore{\expandafter\noexpand\csname #1true\endcsname}%
60   \else
61     \g@addto@macro\Gm@restore{\expandafter\noexpand\csname #1false\endcsname}%
62   \fi}%

```

`\Gm@restore` The initialization for `\Gm@restore`.

```

63 \def\Gm@restore{}%

```

`\Gm@save` The definition of the macro saving the real lengths \TeX options.

```

64 \def\Gm@save{%
65   \Gm@savelength{\paperwidth}%
66   \Gm@savelength{\paperheight}%
67   \Gm@savelength{\textwidth}%
68   \Gm@savelength{\textheight}%
69   \Gm@savelength{\evensidemargin}%
70   \Gm@savelength{\oddsidemargin}%
71   \Gm@savelength{\topmargin}%
72   \Gm@savelength{\headheight}%
73   \Gm@savelength{\headsep}%
74   \Gm@savelength{\topskip}%
75   \Gm@savelength{\footskip}%
76   \Gm@savelength{\baselineskip}%
77   \Gm@savelength{\marginparwidth}%
78   \Gm@savelength{\marginparsep}%
79   \Gm@savelength{\columnsep}%
80   \Gm@savelength{\hoffset}%
81   \Gm@savelength{\voffset}%
82   \Gm@savelength{\Gm@layoutwidth}%
83   \Gm@savelength{\Gm@layoutheight}%
84   \Gm@savelength{\Gm@layouthoffset}%

```

```

85 \Gm@savelength{Gm@layoutvoffset}%
86 \Gm@saveboolean{@twocolumn}%
87 \Gm@saveboolean{@twoside}%
88 \Gm@saveboolean{@mparswitch}%
89 \Gm@saveboolean{@reversemargin}}%

```

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

```

90 \def\Gm@initnewgm{%
91   \Gm@passfalse
92   \Gm@swap@papersizefalse
93   \Gm@dimlist={}
94   \Gm@hbodyfalse
95   \Gm@vbodyfalse
96   \Gm@heightroundedfalse
97   \Gm@includeheadfalse
98   \Gm@includefootfalse
99   \Gm@includempfalse
100  \let\Gm@width\@undefined
101  \let\Gm@height\@undefined
102  \let\Gm@textwidth\@undefined
103  \let\Gm@textheight\@undefined
104  \let\Gm@lines\@undefined
105  \let\Gm@hscale\@undefined
106  \let\Gm@vscale\@undefined
107  \let\Gm@hmarginratio\@undefined
108  \let\Gm@vmarginratio\@undefined
109  \let\Gm@lmargin\@undefined
110  \let\Gm@rmargin\@undefined
111  \let\Gm@tmargin\@undefined
112  \let\Gm@bmargin\@undefined
113  \Gm@layoutfalse
114  \Gm@layoutoffset\z@
115  \Gm@layoutvoffset\z@
116  \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.

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

159 \def\Gm@defbylen#1#2{%
160   \begingroup\setlength\@tempdima{#2}%
161   \expandafter\xdef\csname Gm@#1\endcsname{\the\@tempdima}\endgroup}%
162 \def\Gm@defbycnt#1#2{%
163   \begingroup\setcounter{Gm@tempcnt}{#2}%
164   \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.

```

165 \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@mratio` 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.

```

166 \def\Gm@setbyratio[#1]#2#3#4{% determine #4 by ratio
167   \expandafter\Gm@sep@ratio\Gm@mratio\relax
168   \if#1b
169     \edef\@tempa{\the\@tempcnta}%
170     \@tempcnta=\@tempcntb
171     \@tempcntb=\@tempa\relax
172   \fi
173   \expandafter\setlength\expandafter\@tempdimb\expandafter
174     {\csname Gm@#3\endcsname}%
175   \ifnum\@tempcntb>\z@
176     \multiply\@tempdimb\@tempcnta
177     \divide\@tempdimb\@tempcntb
178   \fi
179   \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.

```

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

```

```

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

```

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

```

`\Gm@detall` 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.

```

220 \def\Gm@detall#1#2#3#4{%
221   \@tempcnta=z@
222   \if#1h
223     \let\Gm@ratio\Gm@hmarginratio
224     \edef\Gm@Dmratio{\if@twoside\Gm@Dhratio\else\Gm@Dhratio\fi}%
225   \else
226     \let\Gm@ratio\Gm@vmarginratio
227     \edef\Gm@Dmratio{\Gm@Dvratio}%
228   \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.

```

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

```

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

```

240 \ifcase\@tempcnta
241 \if#1h
242 \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
243 \else
244 \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
245 \fi
246 \Gm@detiandiii{#2}{#3}{#4}%

```

Case 001 (=1) with right(bottom) fixed:

```

247 \or
248 \ifx\Gm@mratio\@undefined
249 \if#1h
250 \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
251 \else
252 \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
253 \fi
254 \setlength\@tempdimc{\@nameuse{Gm@#4}}%
255 \Gm@detiandiii{#2}{#3}{#4}%
256 \expandafter\let\csname Gm@#2\endcsname\@undefined
257 \Gm@defbylen{#4}{\@tempdimc}%
258 \else
259 \Gm@setbyratio[f]{#1}{#4}{#3}%
260 \fi
261 \Gm@detiv{#2}{#3}{#4}{#2}%

```

Case 010 (=2) with width(height) fixed:

```

262 \or\Gm@detiandiii{#2}{#3}{#4}%

```

Case 011 (=3) with both width(height) and right(bottom) fixed:

```

263 \or\Gm@detiv{#2}{#2}{#4}{#3}%

```

Case 100 (=4) with left(top) fixed:

```

264 \or
265 \ifx\Gm@mratio\@undefined
266 \if#1h
267 \Gm@defbylen{width}{\Gm@Dhscale\Gm@layoutwidth}%
268 \else
269 \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
270 \fi
271 \setlength\@tempdimc{\@nameuse{Gm@#3}}%
272 \Gm@detiandiii{#2}{#4}{#3}%
273 \expandafter\let\csname Gm@#2\endcsname\@undefined
274 \Gm@defbylen{#3}{\@tempdimc}%
275 \else
276 \Gm@setbyratio[b]{#1}{#3}{#4}%
277 \fi
278 \Gm@detiv{#2}{#3}{#4}{#2}%

```

Case 101 (=5) with both left(top) and right(bottom) fixed:

```

279 \or\Gm@detiv{#2}{#3}{#4}{#2}%

```

Case 110 (=6) with both left(top) and width(height) fixed:

```

280 \or\Gm@detiv{#2}{#2}{#3}{#4}%

```

Case 111 (=7) with all fixed though it is over-specified:

```

281 \or\Gm@warning{Over-specification in '#1'-direction.%
282 ~J\@spaces '#2' (\@nameuse{Gm@#2}) is ignored}%
283 \Gm@detiv{#2}{#3}{#4}{#2}%
284 \else\fi}%

```

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

```

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

```

```

291 \let\Gm@hscale\@undefined
292 \let\Gm@width\@undefined
293 \let\Gm@textwidth\@undefined
294 \fi
295 \ifGm@vbody\else
296 \let\Gm@vscale\@undefined
297 \let\Gm@height\@undefined
298 \let\Gm@textheight\@undefined
299 \fi
300 }%

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

345 \def\Gm@expandlengths{%

```



```

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

```

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

```

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

```

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

```

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

```

Various paper size are defined here.

```

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

```

`'paper'` paper takes a paper name as its value.

```

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

```

`'a[0-6]paper'` The following paper names are available.

```

' b[0-6]j' 395 \define@key{Gm}{a0paper}[true]{\Gm@setpaper@ifpre{a0paper}}%
' ansi[a-e]paper' 396 \define@key{Gm}{a1paper}[true]{\Gm@setpaper@ifpre{a1paper}}%
' letterpaper' 397 \define@key{Gm}{a2paper}[true]{\Gm@setpaper@ifpre{a2paper}}%
' legalpaper' 398 \define@key{Gm}{a3paper}[true]{\Gm@setpaper@ifpre{a3paper}}%
' executivepaper' 399 \define@key{Gm}{a4paper}[true]{\Gm@setpaper@ifpre{a4paper}}%
' screen' 400 \define@key{Gm}{a5paper}[true]{\Gm@setpaper@ifpre{a5paper}}%

```

```

401 \define@key{Gm}{a6paper}[true]{\Gm@setpaper@ifpre{a6paper}}%
402 \define@key{Gm}{b0paper}[true]{\Gm@setpaper@ifpre{b0paper}}%
403 \define@key{Gm}{b1paper}[true]{\Gm@setpaper@ifpre{b1paper}}%
404 \define@key{Gm}{b2paper}[true]{\Gm@setpaper@ifpre{b2paper}}%
405 \define@key{Gm}{b3paper}[true]{\Gm@setpaper@ifpre{b3paper}}%
406 \define@key{Gm}{b4paper}[true]{\Gm@setpaper@ifpre{b4paper}}%
407 \define@key{Gm}{b5paper}[true]{\Gm@setpaper@ifpre{b5paper}}%
408 \define@key{Gm}{b6paper}[true]{\Gm@setpaper@ifpre{b6paper}}%
409 \define@key{Gm}{c0paper}[true]{\Gm@setpaper@ifpre{c0paper}}%
410 \define@key{Gm}{c1paper}[true]{\Gm@setpaper@ifpre{c1paper}}%
411 \define@key{Gm}{c2paper}[true]{\Gm@setpaper@ifpre{c2paper}}%
412 \define@key{Gm}{c3paper}[true]{\Gm@setpaper@ifpre{c3paper}}%
413 \define@key{Gm}{c4paper}[true]{\Gm@setpaper@ifpre{c4paper}}%
414 \define@key{Gm}{c5paper}[true]{\Gm@setpaper@ifpre{c5paper}}%
415 \define@key{Gm}{c6paper}[true]{\Gm@setpaper@ifpre{c6paper}}%
416 \define@key{Gm}{b0j}[true]{\Gm@setpaper@ifpre{b0j}}%
417 \define@key{Gm}{b1j}[true]{\Gm@setpaper@ifpre{b1j}}%
418 \define@key{Gm}{b2j}[true]{\Gm@setpaper@ifpre{b2j}}%
419 \define@key{Gm}{b3j}[true]{\Gm@setpaper@ifpre{b3j}}%
420 \define@key{Gm}{b4j}[true]{\Gm@setpaper@ifpre{b4j}}%
421 \define@key{Gm}{b5j}[true]{\Gm@setpaper@ifpre{b5j}}%
422 \define@key{Gm}{b6j}[true]{\Gm@setpaper@ifpre{b6j}}%
423 \define@key{Gm}{ansiapaper}[true]{\Gm@setpaper@ifpre{ansiapaper}}%
424 \define@key{Gm}{ansibpaper}[true]{\Gm@setpaper@ifpre{ansibpaper}}%
425 \define@key{Gm}{ansicpaper}[true]{\Gm@setpaper@ifpre{ansicpaper}}%
426 \define@key{Gm}{ansidpaper}[true]{\Gm@setpaper@ifpre{ansidpaper}}%
427 \define@key{Gm}{ansiepaper}[true]{\Gm@setpaper@ifpre{ansiepaper}}%
428 \define@key{Gm}{letterpaper}[true]{\Gm@setpaper@ifpre{letterpaper}}%
429 \define@key{Gm}{legalpaper}[true]{\Gm@setpaper@ifpre{legalpaper}}%
430 \define@key{Gm}{executivepaper}[true]{\Gm@setpaper@ifpre{executivepaper}}%
431 \define@key{Gm}{screen}[true]{\Gm@setpaper@ifpre{screen}}%

```

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

```

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

```

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

```

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

```

‘landscape’ Paper orientation setting.

```

‘portrait’ 443 \define@key{Gm}{landscape}[true]{\ifGm@preamble{landscape}{%
444 \Gm@doifelse{landscape}{#1}%
445 {\ifGm@landscape\else\Gm@landscapetrue\Gm@reverse{swap@papersize}\fi}%
446 {\ifGm@landscape\Gm@landscapetrue\Gm@reverse{swap@papersize}\fi}}%
447 \define@key{Gm}{portrait}[true]{\ifGm@preamble{portrait}{%
448 \Gm@doifelse{portrait}{#1}%
449 {\ifGm@landscape\Gm@landscapetrue\Gm@reverse{swap@papersize}\fi}%
450 {\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’ 451 \define@key{Gm}{hscale}{\Gm@hbodytrue\edef\Gm@hscale{#1}}%
‘scale’ 452 \define@key{Gm}{vscale}{\Gm@vbodytrue\edef\Gm@vscale{#1}}%
453 \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’

‘totalwidth’
‘totalheight’

```

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

‘textwidth’ These options directly sets the dimensions \textwidth and \textheight. body is an alias of text.
‘textheight’ 459 \define@key{Gm}{textwidth}{\Gm@hbodytrue\Gm@defbylen{textwidth}{#1}}%
‘text’ 460 \define@key{Gm}{textheight}{\Gm@vbodytrue\Gm@defbylen{textheight}{#1}}%
‘body’ 461 \define@key{Gm}{text}{\Gm@branch{#1}{textwidth}{textheight}}%
462 \let\KV@Gm@body\KV@Gm@text

‘lines’ The option sets \textheight with the number of lines.
463 \define@key{Gm}{lines}{\Gm@vbodytrue\Gm@defbycnt{lines}{#1}}%

‘includehead’ The options take the corresponding dimensions as part of body.
‘includefoot’ 464 \define@key{Gm}{includehead}[true]{\Gm@setbool{includehead}{#1}}%
‘includeheadfoot’ 465 \define@key{Gm}{includefoot}[true]{\Gm@setbool{includefoot}{#1}}%
‘incluemp’ 466 \define@key{Gm}{includeheadfoot}[true]{\Gm@doifelse{includeheadfoot}{#1}}%
‘includeall’ 467 {\Gm@includeheadtrue\Gm@includefoottrue}%
468 {\Gm@includeheadfalse\Gm@includefootfalse}%
469 \define@key{Gm}{incluemp}[true]{\Gm@setbool{incluemp}{#1}}%
470 \define@key{Gm}{includeall}[true]{\Gm@doifelse{includeall}{#1}}%
471 {\Gm@includeheadtrue\Gm@includefoottrue\Gm@incluemptrue}%
472 {\Gm@includeheadfalse\Gm@includefootfalse\Gm@incluempfalse}%

‘ignorehead’ These options exclude head, foot and marginpars when determining body.
‘ignorefoot’ 473 \define@key{Gm}{ignorehead}[true]{%
‘ignoreheadfoot’ 474 \Gm@setboolrev[ignorehead]{includehead}{#1}}%
‘ignoremp’ 475 \define@key{Gm}{ignorefoot}[true]{%
‘ignoreall’ 476 \Gm@setboolrev[ignorefoot]{includefoot}{#1}}%
477 \define@key{Gm}{ignoreheadfoot}[true]{\Gm@doifelse{ignoreheadfoot}{#1}}%
478 {\Gm@includeheadfalse\Gm@includefootfalse}%
479 {\Gm@includeheadtrue\Gm@includefoottrue}%
480 \define@key{Gm}{ignoremp}[true]{%
481 \Gm@setboolrev[ignoremp]{incluemp}{#1}}%
482 \define@key{Gm}{ignoreall}[true]{\Gm@doifelse{ignoreall}{#1}}%
483 {\Gm@includeheadfalse\Gm@includefootfalse\Gm@incluempfalse}%
484 {\Gm@includeheadtrue\Gm@includefoottrue\Gm@incluemptrue}%

‘heightrounded’ The option rounds \textheight to n-times of \baselineskip plus \topskip.
485 \define@key{Gm}{heightrounded}[true]{\Gm@setbool{heightrounded}{#1}}%

‘hdivide’ The options are useful to specify partitioning in each direction of the paper.
‘vdivide’ 486 \define@key{Gm}{hdivide}{\Gm@parse@divide{#1}{lmargin}{width}{rmargin}}%
‘divide’ 487 \define@key{Gm}{vdivide}{\Gm@parse@divide{#1}{tmargin}{height}{bmargin}}%
488 \define@key{Gm}{divide}{\Gm@parse@divide{#1}{lmargin}{width}{rmargin}}%
489 \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’ 490 \define@key{Gm}{lmargin}{\Gm@defbylen{lmargin}{#1}}%
‘left’ 491 \define@key{Gm}{rmargin}{\Gm@defbylen{rmargin}{#1}}%
‘inner’ 492 \let\KV@Gm@left\KV@Gm@lmargin
‘innermargin’ 493 \let\KV@Gm@inner\KV@Gm@lmargin
‘right’ 494 \let\KV@Gm@innermargin\KV@Gm@lmargin
‘outer’ 495 \let\KV@Gm@right\KV@Gm@rmargin
‘outermargin’ 496 \let\KV@Gm@outer\KV@Gm@rmargin
‘top’ 497 \let\KV@Gm@outermargin\KV@Gm@rmargin
‘bottom’ 498 \define@key{Gm}{tmargin}{\Gm@defbylen{tmargin}{#1}}%
499 \define@key{Gm}{bmargin}{\Gm@defbylen{bmargin}{#1}}%
500 \let\KV@Gm@top\KV@Gm@tmargin
501 \let\KV@Gm@bottom\KV@Gm@bmargin

```

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

‘vmargin’ 502 \define@key{Gm}{hmargin}{\Gm@branch{#1}{\lmargin}{\rmargin}}%

‘margin’ 503 \define@key{Gm}{vmargin}{\Gm@branch{#1}{\tmargin}{\bmargin}}%

504 \define@key{Gm}{margin}{\Gm@branch{#1}{\lmargin}{\tmargin}}%

505 \Gm@branch{#1}{\rmargin}{\bmargin}}%

‘hmarginratio’ Options specifying the margin ratios.

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

‘marginratio’ 507 \define@key{Gm}{vmarginratio}{\edef\Gm@vmarginratio{#1}}%

‘hratio’ 508 \define@key{Gm}{marginratio}{\Gm@branch{#1}{hmarginratio}{vmarginratio}}%

‘vratio’ 509 \let\KV@Gm@hratio\KV@Gm@hmarginratio

‘ratio’ 510 \let\KV@Gm@vratio\KV@Gm@vmarginratio

511 \let\KV@Gm@ratio\KV@Gm@marginratio

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

‘vcentering’ 512 \define@key{Gm}{hcentering}[true]{\Gm@doifelse{hcentering}{#1}}%

‘centering’ 513 {\def\Gm@hmarginratio{1:1}}{}}%

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

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

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

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

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

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

519 {\@twosidettrue\@mparswitchtrue}{\@twosidefalse\@mparswitchfalse}}%

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

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

521 {\@twosidettrue\@mparswitchfalse}}{}}%

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

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

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

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

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

‘head’ 525 \define@key{Gm}{footskip}{\Gm@setlength\footskip{#1}}%

‘foot’ 526 \let\KV@Gm@head\KV@Gm@headheight

527 \let\KV@Gm@foot\KV@Gm@footskip

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

‘nofoot’ 528 \define@key{Gm}{nohead}[true]{\Gm@doifelse{nohead}{#1}}%

‘noheadfoot’ 529 {\Gm@setlength\headheight\z@\Gm@setlength\headsep\z@}{}}%

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

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

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

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

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

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

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

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

‘marginpar’ 536 \define@key{Gm}{marginparwidth}{\Gm@setlength\marginparwidth{#1}}%

‘marginparsep’ 537 \let\KV@Gm@marginpar\KV@Gm@marginparwidth

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

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

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

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

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

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

‘offset’ The former two options set native dimensions \hoffset and \voffset. offset can set both of them with the same value.

```

‘offset’ 542 \define@key{Gm}{hoffset}{\Gm@setlength\hoffset{#1}}%
543 \define@key{Gm}{voffset}{\Gm@setlength\voffset{#1}}%
544 \define@key{Gm}{offset}{\Gm@branch{#1}{hoffset}{voffset}}%

```

‘layouthoffset’

‘layoutvoffset’ 545 \define@key{Gm}{layouthoffset}{\Gm@setlength\Gm@layouthoffset{#1}}%

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

```

547 \define@key{Gm}{layoutoffset}{\Gm@branch{#1}{layouthoffset}{layoutvoffset}}%

```

‘twocolumn’ The option sets \twocolumn switch.

```

548 \define@key{Gm}{twocolumn}[true]{%
549 \Gm@doif{twocolumn}{#1}{\csname @twocolumn\Gm@bool\endcsname}}%

```

‘onecolumn’ This option has the reverse effect of twocolumn option.

```

550 \define@key{Gm}{onecolumn}[true]{%
551 \Gm@doifelse{onecolumn}{#1}{\@twocolumnfalse}{\@twocolumntrue}}%

```

‘reversemp’ The both options set \reversemargin.

‘reversemarginpar’ 552 \define@key{Gm}{reversemp}[true]{%

```

553 \Gm@doif{reversemp}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%
554 \define@key{Gm}{reversemarginpar}[true]{%
555 \Gm@doif{reversemarginpar}{#1}{\csname @reversemargin\Gm@bool\endcsname}}%

```

‘dviewer’

```

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

```

‘dvips’ The geometry package supports dvips, dvipdfm, pdflatex and vtex. dvipdfm works like dvips.

‘dvipdfm’ 562 \define@key{Gm}{dvips}[true]{\ifGm@preamble{dvips}{%

‘pdftex’ 563 \Gm@doifelse{dvips}{#1}{\Gm@setdriver{dvips}}{\Gm@unsetdriver{dvips}}}%}

‘xetex’ 564 \define@key{Gm}{dvipdfm}[true]{\ifGm@preamble{dvipdfm}{%

‘vtex’ 565 \Gm@doifelse{dvipdfm}{#1}{\Gm@setdriver{dvipdfm}}{\Gm@unsetdriver{dvipdfm}}}%}

```

566 \define@key{Gm}{pdftex}[true]{\ifGm@preamble{pdftex}{%
567 \Gm@doifelse{pdftex}{#1}{\Gm@setdriver{pdftex}}{\Gm@unsetdriver{pdftex}}}%}
568 \define@key{Gm}{xetex}[true]{\ifGm@preamble{xetex}{%
569 \Gm@doifelse{xetex}{#1}{\Gm@setdriver{xetex}}{\Gm@unsetdriver{xetex}}}%}
570 \define@key{Gm}{vtex}[true]{\ifGm@preamble{vtex}{%
571 \Gm@doifelse{vtex}{#1}{\Gm@setdriver{vtex}}{\Gm@unsetdriver{vtex}}}%}

```

‘verbose’ The verbose mode.

```

572 \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.

```

573 \define@key{Gm}{reset}[true]{\ifGm@preamble{reset}{%
574 \Gm@doifelse{reset}{#1}{\Gm@restoreorg\Gm@initall
575 \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.

```

576 \define@key{Gm}{resetpaper}[true]{\ifGm@preamble{resetpaper}{%
577 \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.

```

578 \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.

```

579 \define@key{Gm}{truedimen}[true]{\ifGm@preamble{truedimen}{%
580   \Gm@doifelse{truedimen}{#1}{\let\Gm@truedimen\Gm@true}%
581   {\let\Gm@truedimen\@empty}}}%

```

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

```

582 \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.

```

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

```

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

```

584 \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.

```

585 \def\Gm@setdefaultpaper{%
586   \ifx\Gm@paper\@undefined
587     \Gm@setsize{paper}{\strip@pt\paperwidth,\strip@pt\paperheight}{pt}%
588     \Gm@setsize{Gm@layout}{\strip@pt\paperwidth,\strip@pt\paperheight}{pt}%
589     \Gm@swap@papersizefalse
590   \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.

```

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

```

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

```

611 \def\Gm@checkmp{%
612   \ifGm@includempelse
613     \@tempcnta\z@\@tempcntb\@ne
614     \if@twocolumn
615       \@tempcnta\@ne
616     else
617       \if@reversemargin
618         \@tempcnta\@ne\@tempcntb\z@
619       \fi
620     \fi
621     \@tempdima\marginparwidth
622     \advance\@tempdima\marginparsep
623     \ifnum\@tempcnta=\@ne
624       \@tempdimc\@tempdima

```



```

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

```

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

```

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

```

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

```

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

```

683 \ifGm@vbody
684 \ifx\Gm@height\@undefined
685 \ifx\Gm@vscale\@undefined
686 \Gm@defbylen{height}{\Gm@Dvscale\Gm@layoutheight}%
687 \else
688 \Gm@defbylen{height}{\Gm@vscale\Gm@layoutheight}%
689 \fi
690 \fi
691 \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.

```

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

```

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

```

716 \def\Gm@process{%

```

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

```

717 \ifGm@pass
718 \Gm@restore@org
719 \else
720 \Gm@process
721 \fi}%

```

The main processing macro.

```

722 \def\Gm@@process{%
723 \Gm@expandlengths
724 \Gm@adjustpaper
725 \addtolength\Gm@layoutwidth{-\Gm@bindingoffset}%
726 \Gm@adjustmp
727 \Gm@adjustbody
728 \Gm@detall{h}{width}{lmargin}{rmargin}%
729 \Gm@detall{v}{height}{tmargin}{bmargin}%

```

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

```

730 \setlength\textwidth{\Gm@width}%
731 \setlength\textheight{\Gm@height}%
732 \setlength\topmargin{\Gm@tmargin}%
733 \setlength\oddsidemargin{\Gm@lmargin}%
734 \addtolength\oddsidemargin{-1\Gm@truedimen in}%

```

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

```
735 \ifGm@includemp
736   \advance\textwidth-\Gm@wd@mp
737   \advance\oddsidemargin\Gm@odd@mp
738 \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`.

```
739 \if@mparswitch
740   \setlength\evensidemargin{\Gm@rmargin}%
741   \addtolength\evensidemargin{-1\Gm@truedimen in}%
742   \ifGm@includemp
743     \advance\evensidemargin\Gm@even@mp
744   \fi
745 \else
746   \evensidemargin\oddsidemargin
747 \fi
```

The bindingoffset correction for `\oddsidemargin`.

```
748 \advance\oddsidemargin\Gm@bindingoffset
749 \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`.

```
750 \ifGm@includehead
751   \addtolength\textheight{-\headheight}%
752   \addtolength\textheight{-\headsep}%
753 \else
754   \addtolength\topmargin{-\headheight}%
755   \addtolength\topmargin{-\headsep}%
756 \fi
```

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

```
757 \ifGm@includefoot
758   \addtolength\textheight{-\footskip}%
759 \fi
```

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

```
760 \ifGm@heightrounded
761   \setlength\@tempdima{\textheight}%
762   \addtolength\@tempdima{-\topskip}%
763   \@tempcnta\@tempdima
764   \@tempcntb\baselineskip
765   \divide\@tempcnta\@tempcntb
766   \setlength\@tempdimb{\baselineskip}%
767   \multiply\@tempdimb\@tempcnta
768   \advance\@tempdima-\@tempdimb
769   \multiply\@tempdima\tw@
770   \ifdim\@tempdima>\baselineskip
771     \addtolength\@tempdimb{\baselineskip}%
772   \fi
773   \addtolength\@tempdimb{\topskip}%
774   \textheight\@tempdimb
775 \fi
```

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

```
776 \advance\oddsidemargin\Gm@layouthoffset%
777 \advance\evensidemargin\Gm@layouthoffset%
778 \advance\topmargin\Gm@layoutvoffset%
779 \addtolength\Gm@layoutwidth{\Gm@bindingoffset}%
780 }% 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`.

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

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

```

782 \ifx\Gm@driver\@empty
783 \typeout{*geometry* driver: auto-detecting}%
\ifpdf is defined in ifpdf package in 'oberdiek' bundle.
784 \ifpdf
785 \Gm@setdriver{pdftex}%
786 \else
787 \Gm@setdriver{dvips}%
788 \fi
\ifvtex is defined in ifvtex package in 'oberdiek' bundle.
789 \ifvtex
790 \Gm@setdriver{vtex}%
791 \fi
\ifxetex is defined in ifxetex package written by Will Robertson.

```

```

792 \ifxetex

```

```

793 \Gm@setdriver{xetex}
794 \fi

```

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

```

795 \else
796 \ifx\Gm@driver\Gm@xetex %%
797 \ifxetex\else
798 \Gm@warning{Wrong driver setting: 'xetex'; trying 'pdftex' driver}%
799 \Gm@setdriver{pdftex}
800 \fi
801 \fi
802 \ifx\Gm@driver\Gm@vtex
803 \ifvtex\else
804 \Gm@warning{Wrong driver setting: 'vtex'; trying 'dvips' driver}%
805 \Gm@setdriver{dvips}%
806 \fi
807 \fi
808 \fi
809 \ifx\Gm@driver\relax
810 \typeout{*geometry* detected driver: <none>}%
811 \else
812 \typeout{*geometry* detected driver: \Gm@driver}%
813 \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.

```

814 \def\Gm@showparams#1{%
815 \ifGm@verbose\expandafter\typeout\else\expandafter\wlog\fi
816 {\Gm@logcontent{#1}}}%
817 \def\Gm@showdim#1{* \string#1=\the#1^^J}%
818 \def\Gm@showbool#1{@nameuse{ifGm@#1}#1\space\fi}%

```

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

```

819 \def\Gm@logcontent#1{%
820 *geometry* verbose mode - [ #1 ] result:^^J%
821 \ifGm@pass * pass: disregarded the geometry package!^^J%
822 \else
823 * driver: \if\Gm@driver<none>\else\Gm@driver\fi^^J%
824 * paper: \ifx\Gm@paper\@undefined<default>\else\Gm@paper\fi^^J%
825 * layout: \ifGm@layout<custom>\else<same size as paper>\fi^^J%
826 \ifGm@layout
827 * layout(width,height): (\the\Gm@layoutwidth,\the\Gm@layoutheight)^^J%
828 \fi
829 * layoutoffset:(h,v)=(\the\Gm@layoutoffset,\the\Gm@layoutvoffset)^^J%
830 \@ifundefined{Gm@lines}{* lines: \Gm@lines^^J}%
831 \@ifundefined{Gm@hmarginratio}{* hratio: \Gm@hmarginratio^^J}%
832 \@ifundefined{Gm@vmarginratio}{* vratio: \Gm@vmarginratio^^J}%
833 \ifdim\Gm@bindingoffset=z@else

```

```

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

```

Macros for the page frames and cropmarks.

```

873 \def\Gm@cropmark(#1,#2,#3,#4){%
874 \begin{picture}(0,0)
875 \setlength\unitlength{1truemm}%
876 \linethickness{0.25pt}%
877 \put(#3,0){\line(#1,0){17}}%
878 \put(0,#4){\line(0,#2){17}}%
879 \end{picture}}%
880 \providecommand*\vb@xt@\{vbox to}%
881 \def\Gm@vrule{\vrule width 0.2pt height\textheight depth\z@}%
882 \def\Gm@hrule{\hrule height 0.2pt depth\z@ width\textwidth}%
883 \def\Gm@hruled{\hrule height\z@ depth0.2pt width\textwidth}%
884 \newcommand*\{Gm@vrules@mpi}{%
885 \hb@xt@\@tempdima{\llap{\Gm@vrule}\ignorespaces
886 \hskip \textwidth\Gm@vrule\hskip \marginparsep
887 \llap{\Gm@vrule}\hfil\Gm@vrule}}%
888 \newcommand*\{Gm@vrules@mpii}{%
889 \hb@xt@\@tempdima{\hskip-\marginparwidth\hskip-\marginparsep
890 \llap{\Gm@vrule}\ignorespaces
891 \hskip \marginparwidth\rlap{\Gm@vrule}\hskip \marginparsep
892 \llap{\Gm@vrule}\hskip\textwidth\rlap{\Gm@vrule}\hss}}%
893 \newcommand*\{Gm@pageframes}{%
894 \vb@xt@\z@{%
895 \ifGm@showcrop
896 \vb@xt@\z@\{vskip-1\Gm@truedimen in\vskip\Gm@layoutvoffset%

```

```

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

```

`\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.

```

936 \def\ProcessOptionsKV{\@ifnextchar[%]
937 {\@ProcessOptionsKV}{\@ProcessOptionsKV[]}}%
938 \def\@ProcessOptionsKV[#1]#2{%
939 \let\@tempa\@empty
940 \@tempcnta\z@
941 \if#1p\@tempcnta\@ne\else\if#1c\@tempcnta\tw\fi\fi
942 \ifodd\@tempcnta
943 \edef\@tempa{\@optionlist{\@currname.\@current}}%
944 \else
945 \@for\CurrentOption:=\@classoptionslist\do{%
946 \@ifundefined{KV@#2@\CurrentOption}%
947 {}{\edef\@tempa{\@tempa,\CurrentOption,}}}%
948 \ifnum\@tempcnta=\z@
949 \edef\@tempa{\@tempa,\@optionlist{\@currname.\@current}}%
950 \fi
951 \fi
952 \edef\@tempa{\noexpand\setkeys{#2}{\@tempa}}%
953 \@tempa
954 \AtEndOfPackage{\let\@unprocessedoptions\relax}}%
955 \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.

```

956 \def\Gm@processconfig{%
957   \let\Gm@origExecuteOptions\ExecuteOptions
958   \let\ExecuteOptions\Gm@setkeys
959   \InputIfFileExists{geometry.cfg}{-}{-}
960   \let\ExecuteOptions\Gm@origExecuteOptions}%

```

The original page layout before loading geometry is saved here. \Gm@restore@org is defined here for reset option.

```

961 \Gm@save
962 \edef\Gm@restore@org{\Gm@restore}%
963 \Gm@initall

```

Processing config file.

```
964 \Gm@processconfig
```

The optional arguments to \documentclass are processed here.

```
965 \ProcessOptionsKV[c]{Gm}%
```

Paper dimensions given by class default are stored.

```
966 \Gm@setdefaultpaper
```

The optional arguments to \usepackage are processed here.

```
967 \ProcessOptionsKV[p]{Gm}%
```

Actual settings and calculation for layout dimensions are processed.

```
968 \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.

```

969 \AtBeginDocument{%
970   \Gm@savelength{paperwidth}%
971   \Gm@savelength{paperheight}%
972   \edef\Gm@restore@org{\Gm@restore}%

```

The original paper size is used if resetpaper.

```

973   \ifGm@resetpaper
974     \edef\Gm@pw{\Gm@orgpw}%
975     \edef\Gm@ph{\Gm@orgph}%
976   \else
977     \edef\Gm@pw{\the\paperwidth}%
978     \edef\Gm@ph{\the\paperheight}%
979   \fi

```

If pass is not set, the paper size is multiplied according to the specified mag.

```

980   \ifGm@pass\else
981     \ifnum\mag=\@m\else
982       \Gm@magtooffset
983       \divide\paperwidth\@m
984       \multiply\paperwidth\the\mag
985       \divide\paperheight\@m
986       \multiply\paperheight\the\mag
987     \fi
988   \fi

```

Checking the driver options.

```
989 \Gm@detectdriver
```

If xetex and \pdfpagewidth is defined, \pdfpagewidth and \pdfpageheight would be set.

```

990 \ifx\Gm@driver\Gm@xetex
991   \ifundefined{pdfpagewidth}{-}{-}%
992   \setlength\pdfpagewidth{\Gm@pw}%
993   \setlength\pdfpageheight{\Gm@ph}%
994   \ifnum\mag=\@m\else
995     \ifx\Gm@truedimen\Gm@true
996       \setlength\paperwidth{\Gm@pw}%
997       \setlength\paperheight{\Gm@ph}%

```

```

998     \fi
999     \fi
1000  \fi

```

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

```

1001  \ifx\Gm@driver\Gm@pdftex
1002    \ifundefined{pdfpagewidth}{-}{%
1003      \setlength{pdfpagewidth}{\Gm@pw}%
1004      \setlength{pdfpageheight}{\Gm@ph}}%
1005    \ifnum\mag=\@m\else
1006      \@tempdima=\mag sp%
1007      \ifundefined{pdfhorigin}{-}{%
1008        \divide\pdfhorigin\@tempdima
1009        \multiply\pdfhorigin\@m
1010        \divide\pdfvorigin\@tempdima
1011        \multiply\pdfvorigin\@m}%
1012      \ifx\Gm@truedimen\Gm@true
1013        \setlength{paperwidth}{\Gm@pw}%
1014        \setlength{paperheight}{\Gm@ph}%
1015      \fi
1016    \fi
1017  \fi

```

With VT_EX environment, VT_EX variables are set here.

```

1018  \ifx\Gm@driver\Gm@vtex
1019    \ifundefined{mediawidth}{-}{%
1020      \mediawidth=\paperwidth
1021      \mediaheight=\paperheight}%
1022    \ifvtexdvi
1023      \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1024    \fi
1025  \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.

```

1026  \ifx\Gm@driver\Gm@dvips
1027    \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1028    \ifx\Gm@driver\Gm@dvips\ifGm@landscape
1029      \AtBeginDvi{\special{! /landplus90 true store}}%
1030    \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.

```

1031  \else\ifx\Gm@driver\Gm@dvipdfm
1032    \ifcase\ifx\AtBeginShipoutFirst\relax\@ne\else
1033      \ifx\AtBeginShipoutFirst\@undefined\@ne\else\z\@fi\fi
1034      \AtBeginShipoutFirst{\special{papersize=\the\paperwidth,\the\paperheight}}%
1035    \or
1036      \AtBeginDvi{\special{papersize=\the\paperwidth,\the\paperheight}}%
1037    \fi
1038  \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.

```

1039  \@tempwafalse
1040  \ifGm@showframe
1041    \@tempwattrue
1042  \else\ifGm@showcrop
1043    \@tempwattrue
1044  \fi\fi
1045  \if@tempwa
1046    \RequirePackage{atbegshi}%
1047    \AtBeginShipout{\setbox\AtBeginShipoutBox=\vbox{%
1048      \baselineskip\z@skip\lineskip\z@skip\lineskiplimit\z@
1049      \Gm@pageframes\box\AtBeginShipoutBox}}%

```



```

1050 \fi
    The layout dimensions for \restoregeometry are saved at the end of the \AtBeginDocument.
1051 \Gm@save
1052 \edef\Gm@restore@pkg{\Gm@restore}%
    The package checks whether or not the marginpars overrun the page, if verbose and unless pass.
1053 \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.
1054 \Gm@showparams{preamble}%
    The following lines free the memories no longer needed.
1055 \let\Gm@pw\relax
1056 \let\Gm@ph\relax
1057 }% end of \AtBeginDocument

```

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

```

1058 \newcommand{\geometry}[1]{%
1059 \Gm@clean
1060 \setkeys{Gm}{#1}%
1061 \Gm@process}%
1062 \@onlypreamble\geometry

```

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

```

1063 \DeclareRobustCommand\Gm@changelayout{%
1064 \setlength{\@colht}{\textheight}
1065 \setlength{\@colroom}{\textheight}%
1066 \setlength{\vsize}{\textheight}
1067 \setlength{\columnwidth}{\textwidth}%
1068 \if@twocolumn%
1069 \advance\columnwidth-\columnsep
1070 \divide\columnwidth\tw@%
1071 \@firstcolumntrue%
1072 \fi%
1073 \setlength{\hsize}{\columnwidth}%
1074 \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`.

```

1075 \newcommand{\newgeometry}[1]{%
1076 \clearpage
1077 \Gm@restore@org
1078 \Gm@initnewgm
1079 \Gm@newgmtrue
1080 \setkeys{Gm}{#1}%
1081 \Gm@newgmfalse
1082 \Gm@process
1083 \ifnum\mag=\@m\else\Gm@magtooffset\fi
1084 \Gm@changelayout
1085 \Gm@showparams{newgeometry}}%

```

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

```

1086 \newcommand{\restoregeometry}{%
1087 \clearpage
1088 \Gm@restore@pkg
1089 \Gm@changelayout}%

```

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

```

1090 \newcommand*{\savegeometry}[1]{%
1091 \Gm@save
1092 \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.

```

1093 \newcommand*{\loadgeometry}[1]{%
1094   \clearpage
1095   \@ifundefined{Gm@restore@@#1}{%
1096     \PackageError{geometry}{%
1097       \string\loadgeometry : name ‘#1’ undefined}{%
1098       The name ‘#1’ should be predefined with \string\savegeometry}%
1099   }{\@nameuse{Gm@restore@@#1}%
1100   \Gm@changelayout}}%
1101 \end{package}

```

12 Config file

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

```

1102 <*config>
1103 %<<SAVE_INTACT
1104
1105 % Uncomment and edit the line below to set default options.
1106 %\ExecuteOptions{a4paper}
1107
1108 %SAVE_INTACT
1109 </config>

```

13 Sample file

Here is a sample document for the geometry package.

```

1110 <*samples>
1111 %<<SAVE_INTACT
1112 \documentclass[12pt]{article}% uses letterpaper by default
1113 % \documentclass[12pt,a4paper]{article}% for A4 paper
1114 %-----
1115 % Edit and uncomment one of the settings below
1116 %-----
1117 % \usepackage{geometry}
1118 % \usepackage[centering]{geometry}
1119 % \usepackage[width=10cm,vscale=.7]{geometry}
1120 % \usepackage[margin=1cm, papersize={12cm,19cm}, resetpaper]{geometry}
1121 % \usepackage[margin=1cm,includeheadfoot]{geometry}
1122 \usepackage[margin=1cm,includeheadfoot,includemp]{geometry}
1123 % \usepackage[margin=1cm,bindingoffset=1cm,twoside]{geometry}
1124 % \usepackage[hmarginratio=2:1, vmargin=2cm]{geometry}
1125 % \usepackage[hscale=0.5,twoside]{geometry}
1126 % \usepackage[hscale=0.5,asymmetric]{geometry}
1127 % \usepackage[hscale=0.5,heightrounded]{geometry}
1128 % \usepackage[left=1cm,right=4cm,top=2cm,includefoot]{geometry}
1129 % \usepackage[lines=20,left=2cm,right=6cm,top=2cm,twoside]{geometry}
1130 % \usepackage[width=15cm, marginparwidth=3cm, includemp]{geometry}
1131 % \usepackage[hdivide={1cm,2cm}, vdivide={3cm,8in},, nohead]{geometry}
1132 % \usepackage[headsep=20pt, head=40pt,foot=20pt,includeheadfoot]{geometry}
1133 % \usepackage[text={6in,8in}, top=2cm, left=2cm]{geometry}
1134 % \usepackage[centering,includemp,twoside,landscape]{geometry}
1135 % \usepackage[mag=1414,margin=2cm]{geometry}
1136 % \usepackage[mag=1414,margin=2truecm,truedimen]{geometry}
1137 % \usepackage[a5paper, landscape, twocolumn, twoside,
1138 %   left=2cm, hmarginratio=2:1, includemp, marginparwidth=43pt,
1139 %   bottom=1cm, foot=.7cm, includefoot, textheight=11cm, heightrounded,
1140 %   columnsep=1cm,verbose]{geometry}
1141 %-----
1142 % No need to change below
1143 %-----

```

```

1144 \geometry{verbose,showframe}% the options appended.
1145 \usepackage{lipsum}% for dummy text of 150 paragraphs
1146 \newcommand\mynote{\marginpar[\raggedright
1147 A sample margin note in the left side.]}%
1148 {\raggedright A sample margin note.}}%
1149 \newcommand\myfootnote{\footnote{This is a sample footnote text.}}
1150 \begin{document}
1151 \lipsum[1-2]\mynote\lipsum[3-4]\mynote
1152 \lipsum[5-11]\mynote\lipsum[12]\myfootnote
1153 \lipsum[13-22]\mynote\lipsum[23-32]
1154 \end{document}
1155 %SAVE_INTACT
1156 \samples

```