

LyXの図表・フロート・註釈・ボックス・外部素材に関する詳細説明書

LyXプロジェクトチーム*

第2.4.x版

2024年5月14日

*コメントや間違いの指摘がありましたら、LyX 文書化メーリングリスト lyx-docs@lists.lyx.org までお送り下さい。

目次

1. 図	1
1.1. 画像ダイアログ	1
1.2. 図フロート	3
1.3. 画像形式	5
2. 表	7
2.1. はじめに	7
2.2. 表ダイアログ	7
2.3. 表ツールバー	8
2.4. 表編集メニュー	9
2.5. 表フロート	10
2.6. 多頁表	10
2.6.1. 多頁表での脚註	13
2.6.2. 多頁表の縦横揃え	13
2.6.3. 多頁表のキャプション	14
2.6.3.1. 多頁表への参照	14
2.6.3.2. キャプションの幅	15
2.6.3.3. 表の各ページごとに異なるキャプション	16
2.7. 多頁表にまつわる諸点	18
2.7.1. 多頁表の計算	18
2.7.2. フロートと多頁表	18
2.7.3. 強制改頁	18
2.8. 複数行・連結列・連結行	22
2.8.1. 表セル内の複数行	22
2.8.2. 連結列	23
2.8.2.1. 連結列の基礎	23
2.8.2.2. 連結列の計算	23
2.8.3. 連結行	24
2.8.4. 対角線付きのセル	25
2.9. フォーマル様式の表	28
2.10. 表の垂直揃え	30
2.11. 色付きの表	31
2.11.1. 色付きのセル	31
2.11.2. 色付き行	33
2.11.3. 色付き罫線	34
2.12. 表セルの回転	36

目次

2.13. 表幅の計算	37
2.14. 表の調整	38
2.14.1. 行の余白取り	38
2.14.2. 特殊なセル揃え	40
2.14.3. 自製のセル書式および列書式	41
2.14.4. 罫線幅	42
2.14.5. 点罫	43
3. フロート	47
3.1. はじめに	47
3.2. フロート型	47
3.2.1. アルゴリズムフロート	48
3.3. フロートの連番	48
3.4. フロートの参照	49
3.4.1. 相互参照の書式	50
3.4.2. 参照名の自動命名	51
3.4.3. 参照位置	51
3.5. フロートの設定	52
3.5.1. フロートの配置	52
3.5.2. フロートの中身の揃え	54
3.5.3. 他のフロートオプション	54
3.6. 回転フロート	55
3.7. 従属フロート	55
3.8. 横並びのフロート	55
3.9. キャプションの書式設定	58
3.10. キャプションの配置	59
3.11. フロート一覧	61
4. 註釈	63
4.1. LyX 註釈	63
4.2. 脚註	64
4.2.1. 脚註の付番	65
4.2.2. 脚註の配置	66
4.3. 傍註	68
5. ボックス	71
5.1. はじめに	71
5.2. ボックスダイアログ	71
5.2.1. 寸法	71
5.2.2. 文字揃え	72
5.2.3. 装飾	74
5.3. ボックスの詳細な設定	75
5.4. ミニページ	75
5.5. 段落ボックス (parbox)	76

5.6. 単語や文字のボックス	77
5.6.1. ハイフネーションの回避	77
5.6.2. 垂直揃え	77
5.7. 色付きボックス	78
5.8. ボックスの伸縮回転	79
5.8.1. ボックスの回転	79
5.8.2. ボックスの伸縮	80
6. 本文回り込みオブジェクト	83
6.1. 折返しフロート	83
6.2. 固定回り込みオブジェクト	84
6.3. 頭文字	85
7. 外部文書構成要素	87
7.1. 外部素材	87
7.2. 子文書	90
7.2.1. 外部文書の第1小節	91
7.2.2. 外部文書の第2小節	92
8. プログラムコードリスト	95
A. LyX で使用できる単位	99
B. 画像を含む場合のファイル出力形式	101
B.1. DVI	101
B.2. PostScript	101
B.3. PDF	102
C. dvips 色名一覧	103
D. 第 (2.1) 式の説明	105
参考文献	107
索引	109

1. 図

1.1. 画像ダイアログ

画像を文書に挿入するには、文中の挿入箇所にカーソルを置いて、ツールバーアイコンをクリックするか、**挿入▷画像**メニューを使用して下さい。すると、ダイアログが現れるので、読み込むファイルを選択して下さい。出力中での画像は、本文中の挿入位置に正確に出力されます。

画像ダイアログは、画像をクリックすることによって、いつでも呼び出すことができます。このダイアログには、以下の三つのタブがあります。

画像 ここでは、画像ファイルを選択し、その出力での表示のしかたを調整することができます。画像寸法で使用するこのできる単位については付録 A に説明があります。

回転角と回転原点を指定することによって、反時計廻り方向へ画像を回転することができます。画像は、 L_YX 内でも回転して表示されます。

画像は、パーセント値を指定するか、幅と高さを明示的に設定することによって、伸縮させることができます。幅と高さのどちらか一方のみを指定した場合には、残りの寸法は自動的に決定されます。両方とも指定した場合には、指定した寸法に画像が変換されますので、変形してしまう可能性があります。画像が変形するのを防ぐには、**縦横比を維持する**オプションを指定して下さい。すると画像は、幅と高さが指定した寸法を越えないように伸縮されます。

画像を右クリックすると現れる文脈依存メニューで**外部で編集**という項目を選ぶことによって、画像を指定したプログラムで開くことができます。このプログラムは、 L_YX 設定のファイル書式設定で、各画像書式ごとに設定することができます。

切り抜き 伸縮単位を使用する代わりに、出力における画像の縦横の長さを調整するのに、画像座標の設定をすることができます。座標は、**ファイルから取得**ボタンを押すことによって、自動的に計算することもできます。**境界枠で切り抜く**オプションは、指定された座標内の画像領域のみ印字します。通常は、画像座標を気にする必要はないので、このタブは無視して構いません。

$\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$ と $\text{L}_\text{Y}_\text{X}$ のオプション このタブでは、 $\text{L}_\text{Y}_\text{X}$ 内での画像表示法を調整できる他、 $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$ のエキスパート向けに $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$ オプションを追加することができるようになっていきます。

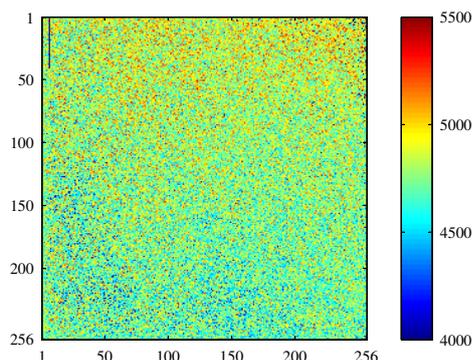
下書きモードオプションを指定すると、画像は出力には現れず、画像と同じ大きさの枠だけが表示されます。

画像グループのセクションでは、画像設定グループを定義したり、既存のもの

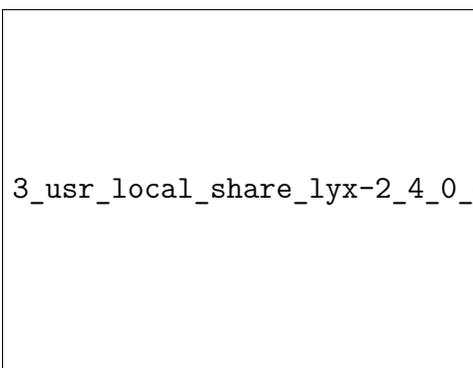
1. 図

に所属させたりすることができます。これらのグループ内の画像は、共通した設定を持つので、グループ内の一つの画像の設定をいじると、自動的にグループ内の全ての画像の設定が同じように変更されます。したがって、たとえば、たくさんの画像の寸法を変更したいときにも、一つ一つの寸法を手動で変更せずに済むのです。既存のグループに帰属させるには、画像の文脈依存メニューを使用して、望むグループ名をチェックすることで行うこともできます。

下記は PDF 書式¹の画像の例で、単独の中央揃えの段落に置いてあります。



下記は上と同じ画像ですが、下書きモードのもので、



¹画像書式は、第 1.3 節に説明されています。

1.2. 図フロート

フロートについての一般的な説明は、第 3.1 節をご覧ください。

ツールバーボタンの **挿入** ▾ **フロート** ▾ **図** メニューを用いると、「図#」（#は実際には数字です）というラベルのキャプションを持ったフロートが挿入されます。画像は、図 1.1 のようにキャプションの上に置くこともできますし、図 1.2 のようにキャプションの下に置くこともできます。キャプションの配置について詳しくは、第 3.10 節に述べられています。

第 1.1 図と第 1.2 図は、参照された図の例です。図は、本文中でラベルを参照することによって参照できます。これをするには、まず、**挿入** ▾ **ラベル** メニューかツールバーボタンを使用して、キャプションにラベルを挿入してください。その後、**挿入** ▾ **相互参照** メニューかツールバーボタンを使用して、ラベルを参照することができます。L^AT_EX は仕上文書でのフロートの位置を調整するので、「上図」のような曖昧な参照のしかたをするのではなく、フロートを直接参照することが重要です。仕上文書では、「上図」ではないことが、大いにありうるからです。

参照は、第 3.4 節で詳細に説明されています。

通常、図フロートには一つの画像だけを挿入しますが、場合によっては、二つの画像に別々の副キャプションをつけて挿入したいこともあるかもしれません。その場合は、既存の図フロートの中に図フロートを入れ込むことによって、実現することができます。ただし、図一覧にはフロートの主キャプションのみが表示されることに留意してください。図 1.3 は、二つの画像が横に並んでいる図フロートの例です。二つめの画像を一つめの下に持ってくることも可能です。図 1.3a と図 1.3b が内部図になります。

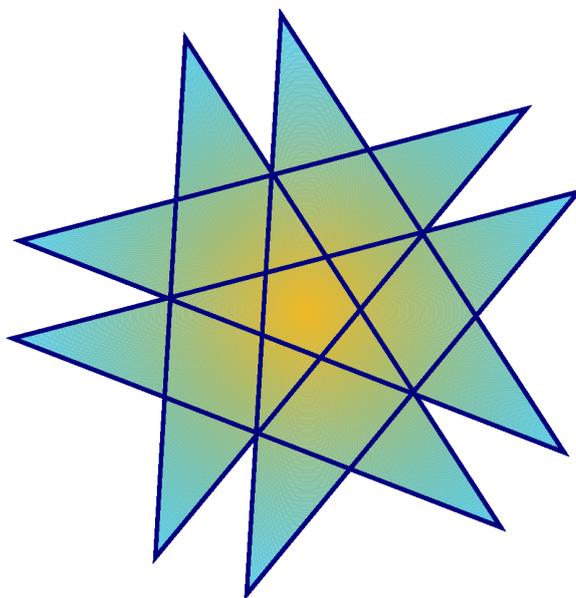
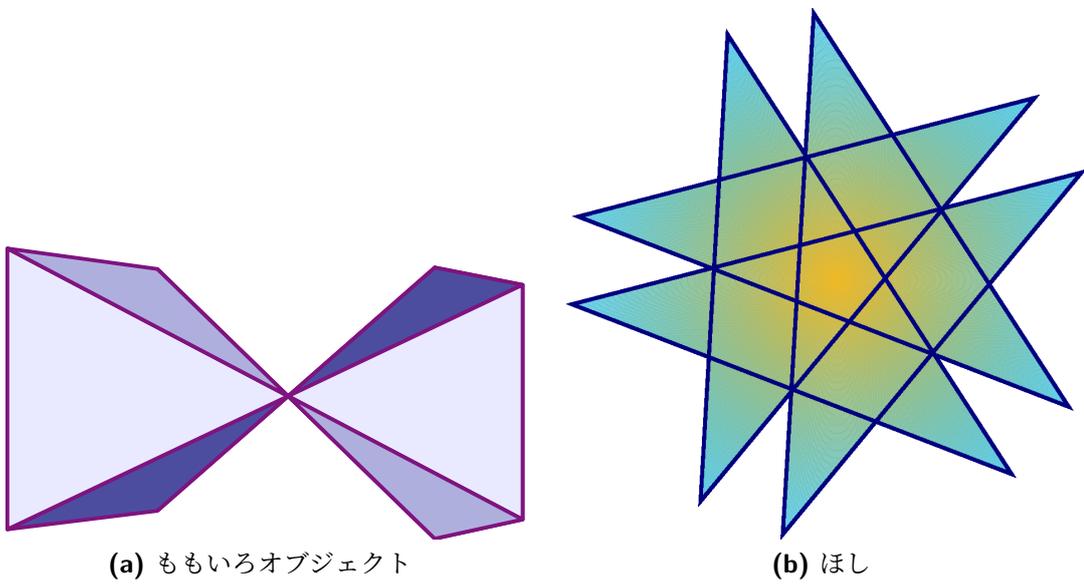
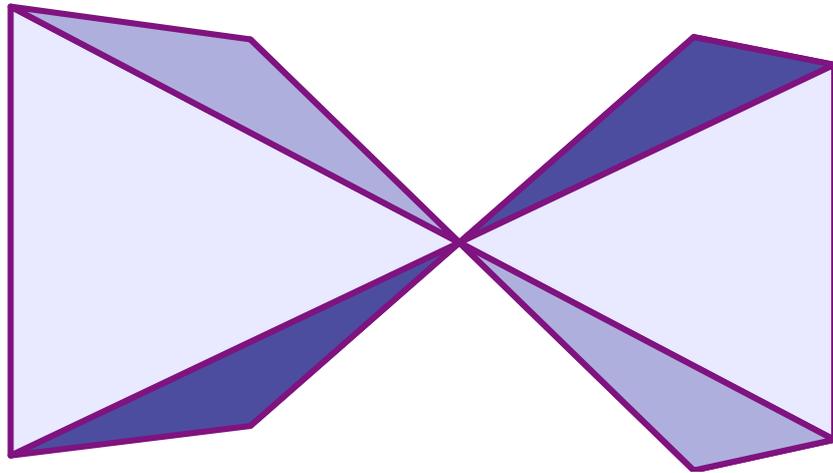


図 1.1.: フロート中の星形

図 1.2.: 3D オブジェクト



(a) ももいろオブジェクト

(b) ほし

図 1.3.: 二つの像. 両画像とも「distorted」画像設定グループに属しています.

1.3. 画像形式

画像は、既知のファイル形式であれば、どのような形式でも挿入することができます。しかし、付録 B で説明されているように、出力に関しては、どの出力文書形式も、いくつかの画像形式しか受け付けません。したがって LyX は、画像を正しい形式に変換するのに、バックグラウンドで ImageMagick というプログラムを使用しています。バックグラウンドで変換が発生することを避けて作業能率を上げようとお考えならば、出力ファイル形式に直接埋め込める画像形式のみを使用するとよいでしょう。出力ファイル形式は、付録 B に説明されています。

フォントと同じく、画像形式にも以下の二つの型があります。

ビットマップ画像 は、ピクセル値を（時により圧縮された形で）保持しています。したがって、この画像形式は完全に伸縮可能ではなく、拡大するとピクセルが目立つようになります。よく知られたビットマップ画像形式には、「Graphics Interchange Format（画像交換形式）」（GIF・ファイル拡張子は「.gif」）、「Portable Network Graphics（可搬性ネットワーク画像ファイル）」（PNG・ファイル拡張子は「.png」）、「Joint Photographic Experts Group（写真専門家連合組織）」（JPG・ファイル拡張子は「.jpg」または「.jpeg」）などがあります。

ベクトル画像 はベクトル値を保持しているので、データ損失なしに任意の寸法の伸縮させることができます。プレゼンテーションは、ビデオプロジェクタがかならず伸縮操作を行うことになるので、プレゼンテーションを作成したいときには画像が伸縮できることが必須です。また、伸縮性は、オンライン文書でユーザーが図の拡大ができるようにするのに便利です。

よく知られた伸縮可能画像形式には、「Scalable Vector Graphics（伸縮可能ベクトル画像ファイル）」（SVG・ファイル拡張子は「.svg」）、「Encapsulated PostScript（カプセル化ポストスクリプト）」（EPS・ファイル拡張子は「.eps」）、「Portable Document Format（可搬文書形式）」（PDF・ファイル拡張子は「.pdf」）、「Windows メタファイル」（WMF・ファイル拡張子は「.wmf」）があります。「あります」と言ったのは、ビットマップ画像はいつでも PDF や EPS に変換することができますが、変換後も依然としてビットマップ画像のままだからです。このような場合には、画像特性を示すヘッダが元々の画像に付け加えられるだけなのです²。たとえば、Adobe Photoshop が生成する PDF ファイルは、ビットマップ画像です。

通常、ビットマップ画像を伸縮可能画像には変換できず、その逆のみが可能です。PDF や PostScript 出力ファイルに直接埋め込むことができるのは、それぞれ PDF 画像書式と EPS 画像書式のみです。いまのところ適切な WMF/SVG→PDF/EPS 変換子がないので、SVG および WMF 画像は、出力ファイルが生成される際にビットマップに計算し直されます。

²PDF の場合には、さらに元画像が圧縮されます。

2. 表

2.1. はじめに

表を挿入するには、ツールバーボタンか、**挿入**▷**表**メニューを選択します。ツールバーボタンを押すと、グラフィカルな選択ができます。マウスを動かして、生成する表の行数・列数を設定し、マウスボタンを押して下さい。表を作るのにメニューを使う場合は、ダイアログが現れ、行数と列数を訊いてきます。

既定の表では、各セルの周囲に罫線が引かれ、第一行が表の他の部分から分離して表示されます。この分離が起こるのは二重線のためです。第一行のセルが下に罫線を伴っているのと同時に、第二行のセルも上に罫線を伴っているのです。以下は表の一例です。

	1	2	3
い			
ろ			
は			

2.2. 表ダイアログ

表を右クリックすると、文脈依存メニューが表示されて、表を変更することができます。メニュー項目詳細以下では、現在カーソルが置かれているセルないし行・列の設定を調節することができます。文脈依存メニュー詳細▷設定を選ぶと、ダイアログボックスが開いて、更に詳細な設定を行うことができます。また、ダイアログオプションのほとんどは、選択範囲に対しても動作します。つまり、複数のセルや行・列を選択していれば、操作は選択範囲の全てに対して行われます。セルの**内容**を選択すると、セル自体を選択するのは異なることに注意して下さい。

ダイアログオプションの直ちに適用を使用すると、ダイアログ中に行った変更が直ちに適用されるようになります。カーソルを別のセルに移動すると、ダイアログも新しいセルのパラメーターに更新されます。ダイアログ中の長さの値を変更した場合には、リターンを押さないと変更が適用されません。

表ダイアログの以下の各タブを使用して、表に変更を加えることができます。

表の設定 ここでは、現在列の幅と水平揃えを設定することができます。小数点でとある揃えは、第2.14.2で説明されています。幅を設定した場合には、現在行の垂直揃えも調整することができます。列幅を与えると、改行や多段落の文章を

2. 表

セルに入れることができます。これに関しては、第 2.8.1 節をご参照下さい。列幅を設定しない場合には、列幅は、もっとも幅の広いセル内容の幅になります。さらに、同一行の一つまたは複数のセルを連結列セルとして指定することができます。これに関しては、第 2.8.2 節をご参照下さい。あるいは同一列のセルを連結行セルとして指定することもできます。これについては、第 2.8.3 節をご参照ください。

回転関連のチェックボックスは、現在セルや選択範囲あるいは表全体を反時計廻りに 90 度回転させます。回転は、 $\text{L}_\text{Y}\text{X}$ 内部では反映されず、出力にのみ現れます。詳細については、第 2.12 節をご覧ください。

[註] DVI ビューアのすべてが回転を表示できるわけではありません。

下記は、本文行中の表に、表全体の垂直揃えを適用したものです。

各種の揃えを適用した同一行中の表：

	1
A	

	1
A	

	1
A	

また、特殊な表設定に必要な $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 引数を入力することもできます。これに関しては、第 2.8.2.2 節および第 2.11 節をご参照下さい。

罫線 このタブでは、現在列ないし現在行の罫線を引いたり消したりすることができます。

様式オプションとして**フォーマル様式**を使用すると、第 2.9 節に述べられているフォーマル様式の表に表を変換します。

またここで、第 2.14.1 節に述べられているように、表の行に空白を追加することもできます。

多頁表 このタブは、複数のページにわたる、いわゆる「**多頁表**」を作るためのものです。第 2.6 節および第 2.7 節に、多頁表の機能に関する詳細な説明があります。

2.3. 表ツールバー

表ツールバーは、表ダイアログの代わりに、表をすばやく変更できるようにするためのものです。通常、カーソルが表の内部にあるときに、 $\text{L}_\text{Y}\text{X}$ のメインウィンドウの底部に表示されます。ご希望であれば、 $\text{L}_\text{Y}\text{X}$ のメインメニューバーを右クリックすることによって、つねに表示されるように切り替えることもできます。

ツールバーには、以下のアイコンがあります。

は、現在セルあるいは選択範囲の下に行を追加します。

は、現在セルあるいは選択範囲の右に列を追加します。

は、現在行あるいは選択範囲行を削除します。

は、現在列あるいは選択範囲列を削除します。

は、行を上を移動します。

- は、列を左に移動します。
- は、行を下に移動します。
- は、列を右に移動します。
- は、現在セルや現在行、あるいは選択範囲の上部に罫線を描画します。
- は、現在セルや現在行、あるいは選択範囲の下部に罫線を描画します。
- は、現在セルや現在行、あるいは選択範囲の左側に罫線を描画します。
- は、現在セルや現在行、あるいは選択範囲の右側に罫線を描画します。
- は、現在セルあるいは選択範囲の周囲に罫線を描画します。現在セルが連結列でなければ、現在行と現在列にも影響します。
- は、現在セルあるいは選択範囲のすべての罫線を消去します。現在セルが連結列でなければ、現在行と現在列にも影響します。
- は、現在セルあるいは現在列の内容を左揃えにします。
- は、現在セルあるいは現在列の内容を水平方向中央揃えにします。
- は、現在セルあるいは現在列の内容を右揃えにします。
- は、現在セルの内容を特定文字位置で揃えます。
- は、現在セルの内容を垂直方向上揃えにします。
- は、現在セルの内容を垂直方向中央揃えにします。
- は、現在セルの内容を垂直方向下揃えにします。
- は、現在セルあるいは選択範囲を反時計廻りに90度回転します。
- は、表全体を反時計廻りに90度回転します。
- は、現在セルあるいは選択範囲を連結列にします。
- は、現在セルあるいは選択範囲を連結行にします。

[註] 出力では、行の最初のセルの垂直揃えが、同一行の他のセルすべてに適用されます。

2.4. 表編集メニュー

表ダイアログと表ツールバーの他に、編集▷表メニューでも、現在行や現在列の罫線を引いたり消したりすることができ、あるいは現在の選択範囲を連結列に指定したりすることができます。このメニューは、カーソルが表内部にあるときのみ、表示されます。

2. 表

表 2.1.: 表フロート

1	2	3
森嶋	宇沢	二階堂
$\int x^2 dx$	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$	$1 + 1 = 2$

2.5. 表フロート

フロートの一般的な説明については、第 3.1 節をご覧ください。

表フロートは、**挿入**▷**フロート**▷**表**メニューかツールバーボタンを使うと挿入することができます。

フロートは折りたたむことのできるボックスとして表示され、中には「表 #:」(#は実際には表番号です)というラベルのついたキャプションがあります。表は、このキャプションの上にも下にも入れることができます。

表 2.1 は、表フロートに入れた表の例です。

キャプションを表の上に入れるのは、標準的な組版方法ですが、残念ながら L^AT_EX の標準クラスではサポートされていません。つまり、`article`・`book`・`letter`・`report` のような文書クラスを使用した場合には、キャプションと表のあいだに余白が入りません。必要な余白を入れるには、使用中の文書のプリアンブルに、LaTeX パッケージの `caption` を

`tableposition=top`

というオプションとともに `load` コマンドを使って読み込んでください¹。キャプション書式を調整するには、第 3.9 節にもなべられているように `caption` パッケージを使用します。

表は、本文中において、ラベルを参照することによって相互参照することができます。そのためには、**挿入**▷**ラベル**メニューかツールバーボタンのを用いて、キャプションにラベルを挿入して下さい。然る後に、**挿入**▷**相互参照**メニューかツールバーボタンのを用いて、このラベルを参照することができます。

参照は、第 3.4 節で詳細に説明されています。

2.6. 多頁表

表が長すぎて 1 ページに収まりきらないときには、表ダイアログの多頁表タブにある多頁表を使うオプションを使用して、複数ページに自動的に分割させることができます。このオプションを有効にすると、いくつかのチェックボックスが使えるようになり、以下のものを定義することができるようになります。

¹詳しくは、第 3.10 節をご覧ください。

ヘッダ：カーソル位置の行が、多頁表の全ページに表示されるヘッダ行として定義されます。ただし、冒頭ヘッダが定義されているときは、冒頭ページを除きます。これと区別して、このヘッダは本体ヘッダと呼びます。

冒頭ヘッダ：カーソル位置の行が、多頁表の最初のページに表示されるヘッダ行として定義されます。

フッタ：カーソル位置の行が、多頁表の全ページに表示されるフッタ行として定義されます。ただし、末尾フッタが定義されているときは、末尾ページを除きます。

末尾フッタ：カーソル位置の行が、多頁表の最後のページに表示されるフッタ行として定義されます。

キャプション：カーソル位置の行を表のキャプションにします。この行は一列にリセットされ、キャプションが挿入されます。多頁表のキャプションについての詳細は、第2.6.3節で説明されています。

加えて、表が分割される行を指定することができます。多頁表の動作を見るために以下の例を参照してください。

例示用電話帳（名前は無視してください）		
氏名		TEL.
Annovi	Silvia	111
Bertoli	Stefano	111
Bozzi	Walter	111
Cachia	Maria	111
Cachia	Maurizio	111
Cinquemani	Giusi	111
Colin	Bernard	111
Concli	Gianfranco	111
Dal Bosco	Carolina	111
Dalpiaz	Annamaria	111
Feliciello	Domenico	111
Focarelli	Paola	111
Galletti	Oreste	111
Gasparini	Franca	111
Rizzardi	Paola	111
Lassini	Giancarlo	111
Malfatti	Luciano	111
Malfatti	Valeriano	111
Meneguzzo	Roberto	111
Mezzadra	Roberto	111
Pirpamer	Erich	111

次頁へつづく

2. 表

例示用電話帳		
氏名		TEL.
Pochiesa	Paolo	111, 222
Radina	Claudio	111
Stuffer	Oskar	111
Tacchelli	Ugo	111
Tezzele	Margit	111
Unterkalmsteiner	Frieda	111
Vieider	Hilde	111
Vigna	Jürgen	111
Weber	Maurizio	111
Winkler	Franz	111
Annovi	Silvia	555
Bertoli	Stefano	555
Bozzi	Walter	555
Cachia	Maria	555
Cachia	Maurizio	555
Cinquemani	Giusi	555
Colin	Bernard	555
Concli	Gianfranco	555
Dal Bosco	Carolina	555
Dalpiaz	Annamaria	555
Feliciello	Domenico	555
Focarelli	Paola	555
Galletti	Oreste	555
Gasparini	Franca	555
Rizzardi	Paola	555
Lassini	Giancarlo	555
Malfatti	Luciano	555
Malfatti	Valeriano	555
Meneguzzo	Roberto	555
Mezzadra	Roberto	555
Pirpamer	Erich	555
Pochiesa	Paolo	555, 222
Radina	Claudio	555
Stuffer	Oskar	555

次頁へつづく

例示用電話帳		
氏名		TEL.
Tacchelli	Ugo	555
Tezzele	Margit	555
Unterkalmsteiner	Frieda	555
Vieider	Hilde	555
Vigna	Jürgen	999
Weber	Maurizio	555
Winkler	Franz	555

以上

2.6.1. 多頁表での脚註

脚註は、どの多頁表セルにも入れることができます。脚註は、脚註のついた表セルのあるページの下部に表示されます。たとえば、表 2.6 には脚註が付いています。

2.6.2. 多頁表の縦横揃え

表設定ダイアログでは、多頁表の水平揃えを指定することができます。左揃えや右揃えでは、表は直接ページ境界に寄せられます。表と境界の間に余白を加えるには、中央揃えに設定してから

`\setlength{\LTleft}{値}`

という行を当該多頁表の前に TeX コードとして挿入して、長さ `\LTleft` および `\LTright` の値を変更しなくてはなりません。ここで「値」は、第 A.1 表に挙げてある単位であれば、どの単位を取ることもできます。`\LTleft` は、左頁余白から多頁表までの水平方向の距離を制御し、`\LTright` は、右頁余白からの距離を制御します。これらの長さの既定値は `\fill` となっており、L_AT_EX の水平フィルと同等になっています。

以下の多頁表は、中央揃えの上、`\LTleft` を `0.1\columnwidth` に設定してあります。

1	2	3	4	5
いろは	ろ	ろ	ろ	いろは
いろは	ろ	ろ	ろ	いろは
いろは	ろ	ろ	ろ	いろは
いろは	いろは	いろは	いろは	いろは

2. 表

2.6.3. 多頁表のキャプション

フロートは一頁に収まらなくてはならないので、多頁表を表フロートに入れることはできません。しかし、第2.6節に述べられているように、ある行で多頁表オプションの**キャプション**を使用すれば、フロートのキャプション環境を多頁表でも使用することができます。キャプションは、**冒頭ヘッダ・ヘッダ・フッタ・末尾フッタ**のいずれかに入れなくてはなりません。それぞれのヘッダ・フッタに入れることができる行は、ひとつだけです。

以下は、動作を確認するための短い多頁表です。

表 2.2.: キャプション付き多頁表

1	2	3	4	5
いろは	ろ	ろ	ろ	いろは
いろは	ろ	ろ	ろ	いろは
いろは	ろ	ろ	ろ	いろは
いろは	いろは	いろは	いろは	いろは

[註1] 表番号は、多頁表にキャプションを付けなかったとしても、各多頁表毎に加番されます。このため、あいだにキャプションのない二つの多頁表が挟まっていたならば、表一覧では、たとえば表2.1の後に表2.4が来ることがあります。これを回避するためには、キャプションのないすべての多頁表の後に、 \TeX コードで

```
\addtocounter{table}{-1}
```

というコマンドを挿入する方法があります。

全多頁表にキャプションがなかったり、文書プリアンプルに以下のコードを書き加えていれば、以上のことは必要ありません。

```
\let\myEnd\endlongtable
```

```
\renewcommand{\endlongtable}{\myEnd\addtocounter{table}{-1}}
```

[註2] 相互参照をリンクするために、**文書設定ダイアログ**のPDF特性で `hyperref` を使用している場合、多頁表キャプションへのリンクは、つねに文頭に向けられます。

2.6.3.1. 多頁表への参照

表 2.3.: 参照された多頁表

1	2	3	4	5
いろは	ろ	ろ	ろ	いろは
いろは	ろ	ろ	ろ	いろは

いろは	ろ	ろ	ろ	いろは
いろは	いろは	いろは	いろは	いろは

多頁表を参照するには、キャプションにラベルを挿入してください。

これは表 2.3 への参照です。

キャプションのレイアウトは、文書中の他のすべてのキャプションと一緒に、 \LaTeX パッケージの `caption` を使用して設定することができます。第 3.9 節をご覧ください。

2.6.3.2. キャプションの幅

キャプション行の最大長は、長さ `\LTcapwidth` によって指定することができます。既定値は 4in です。これを変更するには、現文書の文書プリアンブルか、調整したい多頁表の前の文中に \TeX コードとして

```
\setlength{\LTcapwidth}{幅}
```

というコマンドを書き加えて下さい。ここで「幅」には、付録 A に挙げられている単位を取ることができます。

以下の各表で違いを示します。

表 2.4.: 既定幅のなが〜いフルタイトル・既定幅のなが〜いフルタイトル・既定幅のなが〜いフルタイトル

1	2	3	4	5
いろは	ろ	ろ	ろ	いろは
いろは	ろ	ろ	ろ	いろは
いろは	ろ	ろ	ろ	いろは
いろは	いろは	いろは	いろは	いろは

表 2.5.: 5 cm 幅のなが〜いフルタイトル・5 cm 幅のなが〜いフルタイトル・5 cm 幅のなが〜いフルタイトル

1	2	3	4	5
いろは	ろ	ろ	ろ	いろは
いろは	ろ	ろ	ろ	いろは
いろは	ろ	ろ	ろ	いろは

2. 表

いろは	いろは	いろは	いろは	いろは
-----	-----	-----	-----	-----

[註] この文書のように L^AT_EX パッケージの `caption` を使用している時には、`\LTcapwidth` に既定値の 4in を使用している場合、ページの全幅が用いられます。この場合、正確に 4in 幅のキャプションを作るには、3.99in のように 4.0in と少し違う値を指定するか、`caption` パッケージが提供している `\captionsetup{width=値}` という L^AT_EX コマンドを用いる方法があります。

2.6.3.3. 表の各ページごとに異なるキャプション

次頁以降の表キャプションを、1 頁めとは異なるものにしたい場合には、キャプションを入れる空行をヘッダに指定して下さい。それから、キャプションを右クリックして、文脈依存メニューからキャプション(無番号頁表)を選択してください。表 2.6 は、異なるヘッダを持つ多頁表の一例です。2 番目のキャプションには表番が現れないようにしてあります。

表 2.6.: 例示用電話帳

例示用電話帳 (名前は無視の事)		
氏名		TEL.
Annovi	Silvia	111
Bertoli	Stefano	111
Bozzi	Walter	111
Cachia	Maria	111
Cachia	Maurizio	111
Cinquemani	Giusi	111
Colin	Bernard	111
Concli	Gianfranco	111
Dal Bosco	Carolina	111
Dalpiaz	Annamaria	111
Feliciello	Domenico	111
Focarelli	Paola	111
Galletti	Oreste	111
Gasparini	Franca	111
Rizzardi ²	Paola	111
Lassini	Giancarlo	111
Malfatti	Luciano	111

次頁に続く

²例示用脚註

例示用電話帳の続き

例示用電話帳		
氏名		TEL.
Malfatti	Valeriano	111
Meneguzzo	Roberto	111
Mezzadra	Roberto	111
Pirpamer	Erich	111
Pochiesa	Paolo	111, 222
Radina	Claudio	111
Stuffer	Oskar	111
Tacchelli	Ugo	111
Tezzele	Margit	111
Unterkalmsteiner	Frieda	111
Vieider	Hilde	111
Vigna	Jürgen	111
Weber	Maurizio	111
Winkler	Franz	111
Annovi	Silvia	555
Bertoli	Stefano	555
Bozzi	Walter	555
Cachia	Maria	555
Cachia	Maurizio	555
Cinquemani	Giusi	555
Colin	Bernard	555
Concli	Gianfranco	555
Dal Bosco	Carolina	555
Dalpiaz	Annamaria	555
Feliciello	Domenico	555
Focarelli	Paola	555
Galletti	Oreste	555
Gasparini	Franca	555
Rizzardi	Paola	555
Lassini	Giancarlo	555
Malfatti	Luciano	555
Malfatti	Valeriano	555

次頁に続く

2. 表

例示用電話帳の続き

例示用電話帳		
氏名		TEL.
Meneguzzo	Roberto	555
Mezzadra	Roberto	555

2.7. 多頁表にまつわる諸点

2.7.1. 多頁表の計算

L^AT_EX は、表ページの頁高と改頁をいわゆる単位塊（チャンク chunk）を使って計算します。単位塊は、一度に L^AT_EX のメモリに読み込まれる表の断片です。歴史上の経緯から、既定値はたった表 20 行分に設定されています。多頁にわたる多頁表がある場合、この設定では文書の生成に時間がかかるかもしれません。この場合、

```
\setcounter{LTchunksizes}{100}
```

というコマンド行を文書のプリアンブルに書き加えることで、単位塊の大きさを 100 から 1000 の値に支障なく増やすことができます。

2.7.2. フロートと多頁表

多頁表が開始するのと同じページにフロートが置かれると問題が発生する可能性があります。このような状況を避けるためには、`\clearpage` コマンドを多頁表の前に T_EX コードとして書き加えてください。

2.7.3. 強制改頁

既定の設定では、表は行の間でのみ改頁されます。もし複数行を含むセルがあり、そのセル中で改頁を行いたい時には、セル中の改頁してもよい場所に T_EX コードで改行コマンド「`\\`」を挿入します。`\\` コマンドの前には、現セルよりも右にある列数と同じ数の「`&`」文字を、T_EX コードで挿入してやる必要があります。この `&` は、表セルの区切り文字です。各 `&` の後に、右側の本来のセルに入るべき内容を T_EX コードで書き込み、右側のセルの内容は削除してください。

`\\` コマンドの後ろには、現在列の左側にある列数と同じ数の `&` 文字を挿入してください。表 2.7 では、改頁さるべきセルは、全 3 列中 2 列めにあります。したがって、

```
& 111\\ \newpage  
&
```

というコマンドが、セル中「*Castelchiodato,*」の後に T_EX コードとして挿入してあります。

同一行 3 列目の「111」は削除してあります。 `\newpage` は、改行がこの場所でかならず起こるように指定したいときのみ必要です。これを書かなければ、改頁の候補として指定するに過ぎません。何らかの理由で多頁表のフッタ行に上罫線を引いていないけれども、セルを改頁する場所で水平線を引きたい場合には、代わりに

```
& 111\\
\hline &
```

というコマンドを使用して下さい。

改頁をしたいセルが最右列にある場合には、

```
\setlength{\parfillskip}{0pt}
```

というコマンドを当該セルの初めに T_EX コードで挿入しなくてはなりません。こうすることで、新規頁部分に表示されるセル内の部分が、正しく全幅を使用することができます。

表 2.7.: 表セル内で強制改頁をしている表

例示用電話帳（氏名は無視の事）		
氏名		TEL.
Annovi	Silvia	111
Bertoli	Stefano	111
Bozzi	Walter	111
Cachia	Maria	111
Cachia	Maurizio	111
Cinquemani	Giusi	111
Colin	Bernard	111
Concli	Gianfranco	111
Dal Bosco	Carolina	111
Dalpiaz	Annamaria	111
Feliciello	Domenico	111
Focarelli	Paola	111
Galletti	Oreste	111
Gasparini	Franca	111
Lassini	Giancarlo	111
Malfatti	Luciano	111
Malfatti	Valeriano	111
Meneguzzo	Roberto	111
Mezzadra	Roberto	111

次頁に続く

2. 表

例示用電話帳の続き

例示用電話帳		
	氏名	TEL.
Pirpamer	Erich	111
Pochiesa	Paolo	111, 222
Radina	Claudio	111
Rizzardi	Paolo, 11. Fürst von Montecompati, 11. Fürst von Sulmona und Vivaro, 10. Fürst von Rossano, 5. Herzog von Canemorte, 11. Herzog von Palombara, 5. Herzog von Castelchiodato,	111

次頁に続く

例示用電話帳の続き

例示用電話帳		
氏名		TEL.
	11. Herzog von Poggionativo, 11. Markis von Mentana, Norma, Civitella, Pratica, Moricone und Percille, 11. Graf von Valinfreda, 11. Baron von Cropalati, 11. Herr von Scarpa, Edelmann von Rom, Patrizier von Venedig, Neapel und Genua	
Stuffer	Oskar	111
Tacchelli	Ugo	111
Tezzele	Margit	111
Unterkalmsteiner	Frieda	111
Vieider	Hilde	111
Vigna	Jürgen	111
Weber	Maurizio	111
Winkler	Franz	111

次頁に続く

2. 表

2.8. 複数行・連結列・連結行

2.8.1. 表セル内の複数行

表 2.8.: セル内に複数の行がある表

複数の 行	ろ	は
に	ほ	へ
と	ち	り

列に固定幅を指定することによって、文章をハイフネーションを伴った複数行の段落として、入れることができるようになります。

表 2.8 を生成するには、まず 3×3 表を作り、最初のセルを右クリックして現れるメニューから、**設定** メニューを選択します。すると表ダイアログが現れるので、セル幅を 2.5 cm 幅とし、水平揃え及び垂直揃えをともに中央揃えに設定します。垂直揃えは、その行のすべてのセルに適用されます。この例の文章は 2.5 cm よりも短いため、一行しか表示されません。そこで二行とも表示するために、両端揃え改行（ショートカット）を挿入してあります。もし文章が設定されたセル幅よりも広い場合には、文章は自動的にいくつかの行に改行されます。

固定幅のセルに長い単語を入れた場合、もしそれがセルの最初にあると、 \LaTeX はハイフネーションを行うことができません。したがって、その単語がセルの最初に来ないようにするためには、何かを入れなくてはなりません。そこで、その単語の前に 0pt 幅の水平空白を挿入します。空白がゼロ幅なので、出力は変わりません。表 2.9 が違いを示しています³。

表 2.9.: ハイフネーション有/無の表

verylongtablecell	ろ	は
に	ほ	へ
と	ち	り

very-longtablecell-word	ろ	は
に	ほ	へ
と	ち	り

³ [訳註] \pLaTeX を使用している場合、日本語の文章は正しく改行されます。

2.8.2. 連結列

2.8.2.1. 連結列の基礎

一つのセルが複数の列にまたがるようにするためには、行中から一つの連結セルにまとめるべき各セルを選択して、表ツールバーボタンを押すか、**編集**▷**表**▷**連結列**メニューを使用するか、あるいは選択したセルを右クリックして現れるメニューから連結列を有効にするかしてください。

連結列はセル設定が独立しています。つまり、セル罫線やセルの縦横揃え、幅設定等を変更した場合には、現在の連結列のみに適用されるのです。以下は、第一行に連結行があり、最下行に上罫線のない連結行がある表の例です。

いろは	にほへとちり		ぬるを
甲	乙	丙	丁
一	二	三	四

2.8.2.2. 連結列の計算

LyX は、連結列を直接にサポートしていますが、同列にある連結セル以外のセル幅については、考慮が必要です。

表 2.10.: 中央揃え連結列の下にちょうど半分の幅を持つ列が2つある表

複数行にわたる連結列		は
に	ほ	へ
と	ち	り

たとえば、表 2.10 を生成するには、3×3 表の第一行の最初の二つのセルを選択して、右クリックして現れるメニューから設定を選択します。そして現れる表の設定ダイアログで、このセルを**連結列**、**中央揃え**、2.5 cm 幅に設定します。被連結列のセルが連結列セルのちょうど半分の幅になるようにしたいので、第一列の幅を 1.25 cm 幅に設定します。すると第二列は自動的に 1.25 cm 幅（連結列幅マイナス第一列の幅）になります。表 2.11 もこのようにして作成してあります。

ご覧になって分かるように、第一列が、連結セルの幅の半分よりも少し大きくなっています。その原因は、セル幅が、指定された幅よりもつねに少しだけ大きく描かれるために、指定したセル幅 $W_{\text{指}}$ がセルの全幅 $W_{\text{全}}$ と等しくならないためです。付録 D がこの事情を詳細に説明しています。

n 列を連結する際、各列の全幅が $W_{\text{全:連結}}/n$ となるために必要な指定幅 $W_{\text{指:n}}$ は、

$$W_{\text{指:n}} = (W_{\text{指:連結列}} + (1 - n) \cdot (12.4 \text{ pt})) / n \quad (2.1)$$

2. 表

表 2.11.: 連結セルの下の列が正確に半分になっていない表

複数行にわたる連結列		は
に	ほ	へ
と	ち	り

で計算することができます。上記の例では $n = 2$ かつ $W_{\text{指:連結列}} = 2.5 \text{ cm}$ かつ長さは既定値ですから、第 2.1 式は

$$W_{\text{指:2}} = 1.25 \text{ cm} - 6.2 \text{ pt} \quad (2.2)$$

となります。L^AT_EX 中で計算を可能にするには、文書プリアンブル行に

```
\usepackage{calc}
```

と書いて、L^AT_EX パッケージ `calc` を読み込まなくてはなりません。L^YX の「表の設定」ダイアログの幅フィールドでは、長さを計算することができません。したがって、ダイアログの L^AT_EX の引数欄に書き込むことによって列を整形しなくてはなりません。以下は、その引数の概要です。

- `p{幅}` は固定幅セルを生成し、内容文は垂直方向上揃えになります。
- `m{幅}` は固定幅セルを生成し、内容文は垂直方向中央揃えになります。
- `b{幅}` は固定幅セルを生成し、内容文は垂直方向下揃えになります。

L^AT_EX 変数を入力すると、表ダイアログで設定されたすべてのセル設定は無効になります。

[註] バグのせいで、L^YX は無効にされた設定も表示してしまいます。

文章を水平方向中央揃えにしたいので、`\centering` コマンドを入れます。したがって、以下のような L^AT_EX 変数を、最初の被連結列に入れます。

```
>\centering m{1.25cm-6.2pt}
```

`>{ }` コマンドは、セルが生成される前に中括弧内のコマンドを適用するという意味です。

こうして被連結列セルの文章に、中央揃えを選択しましたが、連結セルの文章は依然として左揃えになっています。これは、L^YX は揃えを一行分のみ適用するためです。したがって、連結セルには

```
>\centering m{2.5cm}
```

という L^AT_EX 変数を指定しなくてはなりません。

2.8.3. 連結行

複数行にかかるセルを連結するには、同一列中の統合したいセルをすべて選択し、ツールバーボタンを押すか、編集 ▾ 表 ▾ 連結行 メニューを選ぶか、選択したセルを右クリックして、文脈依存メニューから 連結行 を選んでください。

下記は一列目に連結列を適用した表の例です。

長いセル項目	ろ	は
連結列	に	ほ
	と	ち

連結列の幅は、列全体の幅と揃えを設定することによって設定することができます。**垂直オフセット**は、正または負の値を与えると、連結列の内容を上下にシフトします。

連結行セルの列に明示的な幅が与えられていない場合、連結行セルは当該列の揃えを継承します。それもなければ左揃えです。既定値を上書きするには、

`\renewcommand{\multirowsetup}{\centering}`

というコマンドを使用して、コマンド`\multirowsetup`を書き換えて下さい。すると、文書中の連結行の中身が、すべて中央揃えになります。もしいくつかの表だけ中央揃えにしたいのであれば、プリアンブルではなく表の直前にTeXコードボックスを入れて、このコマンドを書き換えれば済みます。文章を右揃えにしたい場合には、`\centering`の代わりに`\raggedleft`を使用して下さい。左揃えに戻すには、`\raggedright`とします。

下記は、-4ミリの垂直オフセットを設定し、3センチ幅の列中にある右揃えの連結列セルがある表の例です。

長いセル項目	ろ	は
り	ぬ	る
連結列	に	ほ
	と	ち

連結行で新しい行に移るには、強制改行(ショートカット)を使用できますが、この場合、下記の例に示されているようにセル行との整合を顧慮しません。

いいい	ははは
いいい	ににに
おかしな セル	ほほほ

したがって、強制改行を使うのは注意が必要です。

2.8.4. 対角線付きのセル

読者に表中の行見出しや列見出しが何を意味しているかを伝えるには、表の第1セルに対角線を一本ないし二本引いて分割すると便利です。これは、L^AT_EXパッケージの`diagbox`を文書プリアンブルに読み込むことによって実現できます。このパッケー

2. 表

ジには、`\diagbox` というコマンドがありますので、以下のように、これに2ないし3の引数をつけて、`TEX` コードで挿入します。

`\diagbox[オプション]{左下}{右上}`

`\diagbox[オプション]{左下}{左上}{右上}`

最初の書式は、第1セルを一本の対角線で分割し、引数の内容がその上下に入れられます。二つ目の書式は、第1セルを二本の線で分割します。

下記は、`\diagbox` を第1セル中に2引数書式で使った表の例です。

パラメータ 名称	厚み (nm)	透明度 (%)	硬度 (MPa)
Perfect Green	45	80	504
Dark Multi-P	60	72	982
Crystal Clear Pro	18	86	374

3引数書式は、例えばスポーツリーグや距離を示すクロス表に便利です。

結果 ホームチーム アウェイチーム	Tigers	Panthers	Falcons
Tigers	—	1:3	0:1
Panthers	3:1	—	0:0
Falcons	1:0	0:0	—

大きなクロス表では、下記のように、2つないし全部のコーナーセルに対角線を入れると良いかもしれません。

結果 ホームチーム アウェイチーム	Tigers	Panthers	Falcons	ホームチーム アウェイチーム 結果
Tigers	—	1:3	0:1	Tigers
Panthers	3:1	—	0:0	Panthers
Falcons	1:0	0:0	—	Falcons
ホームチーム 結果 アウェイチーム	Tigers	Panthers	Falcons	アウェイチーム ホームチーム 結果

対角線を廻すために、コマンドオプションとして `dir=方角` を使用しています。方角は、`NE`・`SE`・`SW`・`NW` のいずれかで、コンパスでの位置を示します。例えば、`SW` は南西かど (south-west) を示します。したがって、上の表の左下のセルでは、

`\diagbox[dir=SW]{ホームチーム}{結果}{アウェイチーム}`

というコマンドを使用しています。方角を指定しなければ、NW を指定したのと同じ事になります。

対角線を引いたセルに必要な幅は、自動的に計算されます。したがって、計算された幅よりも小さい固定列幅を指定してしまうと、下記の例のように、セルの中身がはみ出てしまいます。

	パラ	メータ
名称	厚み (nm)	
Perfect Green	45	

対角線を引いたセルの列に一定の幅が必要な場合は、コマンドオプションとして `width=3cm` (必要に応じて「3cm」を必要な幅に変更) を与えてください。下記の表では、第1列の幅が 4 cm に設定してあります。 `\diagbox` 全体の幅は、本文幅 (この場合は 4 cm) に、本文とセル罫線との間隔の 2 倍を加えたものになるので、 `\diagbox` には $4\text{ cm} + 2\text{ \tabcolsep}$ の幅が必要です。幅の計算は、 `\diagbox` のコマンドオプション内では行うことができないので、前もって新しい長さを定義しておく必要があります。これは、 $\text{T}_{\text{E}}\text{X}$ コードコマンドで

```
\newlength{\diagtest}
\setlength{\diagtest}{4cm+2\tabcolsep}
```

とすれば実現できます。 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 内で計算を行うには、文書プリアンブル中で `calc` パッケージを読み込んでおく必要があります。

さて、この表の第1セルで

```
\diagbox[width=\diagtest]{A}{B}
```

というコマンドを使用します。

	B	厚み (nm)
A		
Perfect Green	45	

対角線の引いたセルは、当該列中で最も幅広くなっているわけではありません。さもないと、次の表のように、対角線の長さが正しく計算できなくなってしまいます。

	B	厚み (nm)
A		
Perfect Green		45

このような場合には、 `\diagbox` コマンドの `width` オプションを使用してください。こうすると、セル幅が列幅に等しくなるので、 2 \tabcolsep を足す必要はなくなります。下記の表では、3 cm 幅を使用しています。

2. 表

	B	厚み (nm)
A		
Perfect Green		45

対角線を引いたセルに複数行を入力するには、文章をセルに合わせるために高さを追加する必要があります。これは、`\diagbox` にオプション `height=3cm` を与えることで実現できます (「3cm」は必要な高さに置き換えてください)。下記の表では、第1セルに

```
\diagbox[width=3cm,height=1.4cm]{A\\B}{C\\D}
```

というコマンドを使用しています。ここでコマンド `\\` は改行をするためのものです⁴。

	C	厚み (nm)
A	D	
B		
Perfect Green		45

LaTeX パッケージ `diagbox` についての詳しい情報は、取扱説明書 [7] をご覧ください。

2.9. フォーマル様式の表

書籍中の表は、よく表 2.12 のように組版されます。この種の表は「フォーマル様式」と呼ばれます。表をフォーマル様式にするには、表作成ダイアログで「フォーマル」表様式を選択するか、表の設定ダイアログの罫線タブで**フォーマル様式**オプションを有効にして下さい (加えて、**既定フォーマル様式を使う** オプションを有効にすると、罫線は実用的な既定設定になります)。

表中の行に余白を加えるには、第 2.14.1 節に述べられているように、表の設定ダイアログの罫線タブを使用します。

通常の表と違って、フォーマル様式の表には縦罫線がありません。表の横罫線は、通常の表と同様に指定することができますが、出力でこれらの線幅は異なります。最初と最後の表罫線の既定幅は 0.08 em ですが、他の罫線の既定幅は 0.05 em です。

これらの既定幅は、以下のプレアンブル行を指定することによって変更することができます。

```
\let\mytoprule\toprule
\renewcommand{\toprule}{\mytoprule[幅]}
```

上の例は、`toprule` という最初の罫線を設定しています。最後の罫線の線幅を変更するには、`toprule` を `bottomrule` に置き換えて下さい。他の罫線の線幅を変更するに

⁴Ctrl+Return を使った改行は、列が固定幅の時のみ、使用することができます。

表 2.12.: フォーマル表の例

システム	チップ1	チップ2	
検知器の厚み [μm]	300	300	700
エッジ角 [度]	3.55	2.71	7.99
空間分解能 [μm]	4.26	10.17	10.56
f_{\max} における MTF	0.53	0.37	0.39
LSF 空間分解能			
μm 表示	129.7	52.75	50.78
ピクセルサイズに対する%表示	76.3	95.9	92.3

は、`toprule` を `midrule` に置き換えます。線幅を設定する上では、付録 A に挙げてあるすべての単位を使用することができます。

表 2.12 のように、フォーマル表では、列をグループ化するのに切り詰めた罫線をよく使います。これは、表ダイアログの罫線ウィジェットで設定することができます。フォーマル表の場合、上下の罫線には、右と左に小さく分割された部分罫線が表示されます。ここをクリックすると、対応する罫線を切り詰めることができます。

切り詰める罫線の長さや罫線の太さを調整したい場合には、 $\text{T}_{\text{E}}\text{X}$ コードに頼らなくてはなりません。これらを行うコマンドは、以下の通りです（これは切り詰められた真ん中の罫線にも使えます）。

`\cmidrule[幅](刈込){開始列-終了列}`

この線幅の既定値は 0.03em です。「開始列」は罫線を開始する列番号で、「終了列」は罫線を終了する列番号です。終了列は、罫線を一列だけに引きたいときも含めて、常に指定しなくてはなりません。非必須パラメーターの「刈込」は、 $l\{\text{刈込幅}\}$ あるいは $r\{\text{刈込幅}\}$ の値を取り、刈込幅の指定は非必須です。たとえば、 $l\{2\text{pt}\}$ というパラメーターを使うと、罫線が左端から 2pt だけ刈り込まれます。刈込幅を指定しないと、罫線は既定値の 0.5em だけ刈り込まれます（これはダイアログから刈り込みを行ったときの設定です）。

`\cmidrule` の「幅」オプションと「刈込幅」パラメーターは、現在のところ $\text{L}_{\text{Y}}\text{X}$ がサポートしていません。これらが必要ならば、行の第一列の最初のセル要素として、これらの設定を行う `\cmidrule` を $\text{T}_{\text{E}}\text{X}$ コードで入れる必要があります。すると、この罫線は、出力では現在行の上に引かれることになります。

ときには、表 2.13 で例示されているような重なり合った `\cmidrule` が欲しいことがあるかもしれません。ただし、これは良くない様式と見做されているので、本当に必要などきのみ使用してください。これは、 $\text{T}_{\text{E}}\text{X}$ コードで

`\morecmidrules`

コマンドを使用することによって実現することができます。表 2.13 の第二列には

2. 表

`\morecmidrules\cmidrule{2-4}`

というコマンドが使用されており，第六列には

`\morecmidrules\cmidrule{3-4}`

というコマンドが使用されています。

罫線の余白取りにどうしても満足が行かない場合には，

`\specialrule{幅}{上余白}{下余白}`

というコマンドを使用すれば，全列にわたる罫線を生成することができます。これらの特殊機能については，`LATEX` パッケージ `booktabs` [5] の取扱説明書をご参照下さい。

表 2.13.: 特殊な（汚い）フォーマル表

システム	チップ 1		チップ 2	
検知器の厚み [μm]	300	300	700	
エッジ角 [度]	3.55	2.71	7.99	
空間分解能 [μm]	4.26	10.17	10.56	
f_{max} における MTF	0.53	0.37	0.39	
LSF 空間分解能				
μm 表示	129.7	52.75	50.78	
ピクセルサイズに対する%表示	76.3	95.9	92.3	

2.10. 表の垂直揃え

本文行中で表の垂直揃えをするには，その表をボックスの中に入れなくてはなりません。そうした上で，第 5.2 節に述べられているように，ボックスの垂直揃えを行います。

以下の例では，各表は 15 列幅%を持つ `minipage`⁵ ボックスの中に入れてあります。

- 動作試験

い	に	と
ろ	ほ	ち
は	へ	り

- 動作試験

い	に	と
ろ	ほ	ち
は	へ	り

⁵`minipage` は第 5.4 節で説明されています。

• 動作試験	い	に	と
	ろ	ほ	ち
	は	へ	り

• 動作試験	い	に	と
	ろ	ほ	ち
	は	へ	り

動作試験	い	に	と
	ろ	ほ	ち
	は	へ	り

上記のとおり、表の第一行あるいは最終行の中身が、表外の本文行と揃っていません。これらを揃えるには、minipage ボックスを `raisebox`⁶ の中に入れなくてはなりません。上記の例では、最初の項目の第二表は、ボックスの前に

```
\raisebox{0.85\baselineskip}{
```

という $\text{T}_{\text{E}}\text{X}$ コードコマンドを使用して揃えています。ボックスの後には、閉じ括弧}を $\text{T}_{\text{E}}\text{X}$ コードとして挿入してあります。最後の項目の第二表には、

```
\raisebox{-0.32\baselineskip}{
```

というコマンドを使用しています。

【註】 表の行の中身の周囲の本文行への揃えは、つねに同じ値をとるわけではありません。この揃え方に必要な `\raisebox` コマンドの値は、文書フォントやフォント寸法、表罫線の線幅によって変わります。

2.11. 色付きの表

2.11.1. 色付きのセル

表 2.14.: `colortbl` パッケージを使用しない色付き表

い	ろ	は
に	ほ	へ
と	ち	り

もし色の付いた文章が欲しいだけならば、そのセルを選択して、編集▷文字様式メニューで色を選んでください。表 2.14 を作るのに、この方法が使われています。それ以外の場合には、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ パッケージの `colortbl` を使用しなくてはなりません。

色付きの表を作成するには、プリアンブルに

```
\usepackage{colortbl}
```

という行を書いて `colortbl` を読み込まなくてはなりません。列の色は、`>{ }` コマンドの中に

⁶`raisebox` は、第 5.6.2 節で説明されています。

2. 表

`\columncolor{色彩名}`

というコマンドを書いて指定することができます。>{}コマンドに関する詳細は、第 2.8.2.2 節に述べられています。

以下の色彩名が事前に定義されています。

red · green · yellow · blue · cyan · magenta · black · white

また、文書設定中で、文書クラスオプションに

`dvipsnames`

を与えると、さらに 61 色を使用することができます。これらの色は、附録 C に列挙されています。それらの色の 1 つ、Maroon が、例として表 2.15 に使用されています。

また

`\definecolor{色彩名}{色彩モデル}{色彩値}`

というコマンドを使用すれば、自分自身の色彩を定義することができます。ここで色彩モデルは

cmk : シアン (cyan) · マゼンタ (magenta) · 黄 (yellow) · 黒 (black)

rgb : 赤 (red) · 緑 (green) · 青 (blue)

gray : グレー

から選択することができ、色彩値は、0 から 1 の間の数字をコンマで区切ったもので、それぞれ当該色彩モデルにおける各色彩の大きさを示します。

たとえば、プリアンブルに

```
\definecolor{darkgreen}{cmk}{0.5, 0, 1, 0.5}
```

と書くことによって、「*darkgreen*」色を定義することができますし、

```
\definecolor{lightgray}{gray}{0.8}
```

と書けば、「*lightgray*」色を定義することができます。

罫線は

`\rowcolor{色彩名}`

というコマンドを使用して色付けすることができ、セルは

`\cellcolor{色彩名}`

というコマンドで色付けすることができます。どちらのコマンドも、セルの冒頭に \TeX コードで書き込みます。

表中の文字に色付けをするには、セルを選択してから、 LYX のメニュー **編集**▷**文字様式**を使用します。セルに TEX コードが入っている場合には、文章部分だけ選択してください。さもないと色付け指定された TEX コードが $\text{L}\text{A}\text{T}\text{E}\text{X}$ エラーを引き起こしてしまいます。

[註] DVIビューアーのすべてが自己定義の色を表示できるわけではありません。

表 2.15 を作成するには、以下のようにしてください。第一列の色は *darkgreen* にしたいので、この列の L^AT_EX 変数として

```
>{\columncolor{darkgreen}\centering}c
```

と書き込みます。第一列は青にしたいので、この列の最初のセルに

```
\rowcolow{cyan}
```

という T_EX コードコマンドを書き込みます。これによって、第一セルの列色指定が上書きされてしまうことに注意してください。最終行の最後のセルは、

```
\cellcolor{Maroon}
```

という T_EX コードコマンドを書き込むことで、マルーンに色づけされています。この後、編集▷文字様式メニューを使って、各文字に色付けをすることができます。

表 2.15.: colortbl パッケージを使用した色付き表

い	ろ	は
に	ほ	へ
と	ち	り

2.11.2. 色付き行

表の行を交互に色付けする必要がある場合、前節に述べたのとは別な方法があります。文書プリアンブルに

```
\usepackage[table]{xcolor}
```

というコマンドを使用して、*xcolor* パッケージを読み込むのです。*xcolor* は、自動的に *colortbl* パッケージも読み込むので、L^AT_EX エラーを防ぐためには、お使いの L^AT_EX 頒布版中に *colortbl* も導入されていなくてはなりません。

表の行は、文書プリアンブル中か、色付けする最初の表の前に T_EX コードで

```
\rowcolors[コマンド]{行番号}{奇数行の色}{偶数行の色}
```

というコマンドを使用すれば、色付けすることができます。ここで、「行番号」は、奇数行色で色付けする最初の行の番号です。「コマンド」は、各行の前で実行したいコマンドを付加する時に用います。したがって、例えば

```
\rowcolors{1}{white}{lightgray}
```

というコマンドは、表 2.16 のようなカラースキームをもたらします。また、

```
\rowcolors[\hline]{1}{lightgray}{cyan}
```

というコマンドを用いると、表 2.17 のようなカラースキームになります。「奇数行の色」ないし「偶数行の色」(またはその両方)が空のときは、色付けはされませんので、

2. 表

`\rowcolors{1}{}{}`

というコマンドはすべての色付けを削除することになります。 `\rowcolors` は、コマンドに続くすべての表に影響を及ぼします。

表 2.16.: 隔行が灰色に着色された表

せ	す	ん
1	2	3
4	5	6
7	8	9

表 2.17.: 隔行でシアン色に着色され、行間に線の入った表

せ	す	ん
1	2	3
4	5	6
7	8	9

2.11.3. 色付き罫線

第 2.14.4 節に述べられているように、表中の全罫線の線幅は、長さ `\arrayrulewidth` を使って調節することができます。本節の表では、すべて 1.5 pt に設定されています。

たとえば縦罫線を緑に着色するには、第 2.14.3 節の説明にしたがって、以下の列書式を文書プレアンプル中で作成して下さい。

`\newcolumntype{W}{!{\color{green}\vline}}`

表 2.18 では、最右列に \LaTeX 変数 `WcW`、他の列に \LaTeX 変数 `Wc` が使用されています。

複数の色を用いたいときには、列書式をもっと定義してください。

表 2.18.: 色付き縦罫線の表

sd		
	sd	
		sd

表 2.19 のように、たとえば横罫線を赤にしたいときには、表ないし表フロートの前に \TeX コードで

```
\let\myHlineC\hline
\renewcommand{\hline}
{\arrayrulecolor{red}\myHlineC\arrayrulecolor{black}}
```

というコマンドを入力して下さい。

表 2.19.: 色付き横罫線の表

sd		
	sd	
		sd

既定値の黒罫線に戻るには、

```
\renewcommand{\hline}{\myHlineC}
```

というコマンドを、表ないし表フロートの後に T_EX コードで入力して下さい。表 2.20 は、縦横罫線を着色した例です。

表 2.20.: 色付き罫線の表

sd		
	sd	
		sd

一部の罫線のみを着色するには、別の攻略法が必要です。以後の横罫線を全て着色するには、下記のコマンドを、最初の着色罫線の直前に来る行の、行末の T_EX コードとして挿入してください。

```
\\\arrayrulecolor{blue}%
```

黒色の罫線に戻るには、黒色罫線の直前に来る行の行頭に下記のコマンドを加えてください。

```
\arrayrulecolor{black}
```

このようにして、表 2.21 のように罫線に色をつけることができます。しかしながら、この方法では、表の最初の行の上の罫線は着色することはできません。

表 2.22 のように行内の縦罫線の色を変更するには、連結列を使う必要があります。まず始めに、列全体に渡らない縦罫線を全て削除します。それから、前例のように、全罫線を以下のコマンドを使って着色します。

```
\\\arrayrulecolor{blue}%
```

最後に、縦罫線の色を変える必要のある行の全てのセルに T_EX コードで下記のコマンドを挿入します。

```
\multicolumn{1}{!{\color{green}\vline}c!{\color{red}\vline}}{
```

2. 表

表 2.21.: 異なる色の横罫線を持つ表

A table with two rows. The top row contains the letter 'Y' on the left and right, with a blue rectangular box between them. The bottom row contains the letter 'B'. A green horizontal line separates the two rows. The entire table is enclosed in a black border.

表 2.22.: 一行中に異なる色の縦罫線を持つ表

A table with two rows. The top row contains the letter 'Y' on the left and right, with a red rectangular box between them. The bottom row contains the letter 'B'. The bottom row is divided into five vertical columns by lines of different colors: green, yellow, blue, cyan, and magenta. The entire table is enclosed in a black border.

これで、セルの両側に縦罫線が追加され、左側の罫線は緑、右側の罫線は赤になります。色を変える必要がなければ、上記コマンド中、`\color` コマンドは要りません。黒の左罫線と青の右罫線を得るには、全罫線がすでに青に設定されていますので、

```
\multicolumn{1}{!{\color{black}\vline}c|}{
```

と書きます。ここで、`|`は縦罫線を生成しますが、`\color` が使用されている時には、`\vline` が同じ働きをします。色を全く変えないで縦罫線を設定するには、

```
\multicolumn{1}{|c|}{
```

とします。このコマンドの後に、セルの内容を通常通り挿入し、`TEX` コードで閉じ括弧}を入れてセルを終わります。

2.12. 表セルの回転

[註] すべての DVI ビューアが回転表示できるわけではありません。

と入れてください。表 2.23 は、3つのセルを 45° 回転させた用例です。

[註] 回転セル中では、改行(ショートカット)はできません。

表 2.23.: 1行目に回転セルのある表

A table with one row and three columns. The first row contains three cells with rotated text: '45', '33', and '21'. Above each cell is a rotated label: '45' is rotated 45 degrees, '33' is rotated 90 degrees, and '21' is rotated 135 degrees. The entire table is enclosed in a black border.

表 2.24.: 複数行セルを回転させた表

	改行を含む 回転	
い		ほ
ろ	は	に

改行を含むセルを回転させたい場合には、セルの内容を `minipage` か `parbox` に置かなくてはなりません⁷。このセルを 90° 回転させると、ボックス幅はセルの高さになります。適切なボックス幅を見積もる手間を避けるには、可変幅を持つミニページを使用することができます。このようなボックスは、文書設定で**可変幅ミニページモジュール**を読み込むことで、挿入することができるようになります。こうすると、**挿入▷特別差込枠▷ミニページ(可変幅)**メニューが使用できるようになります。表 2.24 は、可変幅ミニページを使用し、回転させた複数行セルの例です。

2.13. 表幅の計算

幅広の表では、表がページ余白に突き出てしまわないように調整するのが難しいことがあります。表中の全列に固定幅を設定する方法もありますが、では各列にどれくらいの幅を設ければ良いのでしょうか？

表幅 $W_{表}$ は計算することができます。各列の実際の幅は、指定した列幅 $W_{列}$ に、列の内容と列罫線との間の余白 ($\text{\LaTeX 長}\code{\tabcolsep}・既定値 6 pt) の 2 倍と、罫線幅 ($\text{\LaTeX 長}\code{\arrayrulewidth}・既定値 0.4 pt) を加えたものになります。したがって、もし同じ幅の n 列を作るならば、表幅は$$

$$W_{表} = n(W_{列} + 2\code{\tabcolsep} + \code{\arrayrulewidth}) + \code{\arrayrulewidth} \quad (2.3)$$

となります。表が使用できる幅は、ページ余白の間の幅か、本文段幅(二段組文書の場合)です ($\text{\LaTeX 長}\code{\columnwidth})。表の列をすべて同じ幅にするとすれば、 $W_{表}$ が $1\code{\columnwidth}$ になりますので、 $W_{列}$ は簡単に計算できます。$

計算を行わせる場合には、文書プリアンブル中で \LaTeX パッケージ `calc` を読みこませる必要があります。表ダイアログ中で幅を設定するには、 \LaTeX の引数フィールドに書きこまなくてはなりません。このフィールドのコマンドスキームは、第 2.8.2.2 節で説明したのと同様です。同幅中央揃え 5 列の表ならば、各列の同フィールドに

```
>\{\centering}p{(1\code{\columnwidth}-62.4pt)/5}
```

というコマンドを入力します。5 列の表で、うち 2 列が他の 3 列の幅の 0.75 倍である

⁷これらのボックスについて詳しくは、第 5.4 節および第 5.5 節をご覧ください。

2. 表

ような表の場合には, 計算は

$$\begin{aligned}\backslash\text{columnwidth} &= 3 W_{\text{列}} + 2 \cdot 0.75 W_{\text{列}} + \backslash\text{arrayrulewidth} \\ &\quad + 5 (2 \backslash\text{tabcolsep} + \backslash\text{arrayrulewidth}) \\ &= 4.5 W_{\text{列}} + 62.4 \text{ pt} \\ W_{\text{列}} &= \frac{\backslash\text{columnwidth} - 62.4 \text{ pt}}{4.5}\end{aligned}\tag{2.4}$$

となります. したがって, 表中の広い列には

```
>\centering p{(1\columnwidth-62.4pt)/\real{4.5}}
```

というコマンドを用い, 狭い列には

```
>\centering p{(1\columnwidth-62.4pt)/\real{4.5}*\real{0.75}}
```

というコマンドを用いることにすると, 以下の表が出来上がります.

1	2	3	4	5
111111 1111111		2222 22222		333333 333333
111111		222222		333333 333333
111111111		222222222		3333333 333
11111 11111		2222222 22		
1111		22222		

2.14. 表の調整

2.14.1. 行の余白取り

表の設定ダイアログの罫線タブで, 表の行に縦方向の余白を加えることができます. ここでは三つのことができます.

行上 これは表の行の文字上に余白を追加します. もし表がフォーマル様式の表⁸ならば, LyX は既定値として 0.5 em の余白を挿入します. 通常の表の場合には, 余白を追加すると, 残念ながら以下の表のように縦罫線が破壊されてしまいます.

い
行上に 3 mm の余白
は

したがって, 通常表の行上に余白を加えるのは, 縦罫線がないときのみを使用することができます.

⁸フォーマル様式の表は第 2.9 節で説明されています.

行下 これは表の行の文字下に余白を追加します。もし表がフォーマル様式の表ならば、LyXは既定値として0.5emの余白を挿入し、通常の表ならば既定値として2ptを挿入します。

行間 これは表の現在行と次の行の間に余白を追加します。もし表がフォーマル様式の表ならば、LyXは既定値として0.5emの余白を挿入します。通常の表の場合には、余白を追加すると、残念ながら以下の表のように縦罫線が破壊されてしまいます。

い
↓ 行間に 3 mm の余白 ↓
↑ 行間に 3 mm の余白 ↑

したがって、通常表の行間に余白を加えるのは、縦罫線がないときのみに行間を調整することができます。

すべての表の全セルの高さを上げたい場合には、以下の行をプリアンブルに加えることによって実現できます。

```
\@ifundefined{extrarowheight}
{\usepackage{array}}{}
\setlength{\extrarowheight}{高さ}
```

というコマンドを書き加えて下さい。ここで「高さ」には、付録Aに挙げられている単位を取ることができます。しかしこの方法は、セル内の文章が縦方向に中央揃えにならないという欠点があります。

標準大よりも大きいフォント寸法を使用している場合には、文字が表罫線に近寄りすぎることがよくあります。これは、表の行の最初にTeXコードで`\strut`コマンドを挿入することで補正することができます。表2.25は、この効果を示したものです。

表 2.25.: 大きなフォント寸法の文章の垂直揃え

<p>(a) 通常の表</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td style="padding: 5px;">ふつう, g</td></tr> <tr><td style="padding: 5px;">やや大</td></tr> <tr><td style="padding: 5px;">大</td></tr> <tr><td style="padding: 5px;">極大</td></tr> <tr><td style="padding: 5px;">極々大</td></tr> <tr><td style="padding: 5px;">最大</td></tr> </table>	ふつう, g	やや大	大	極大	極々大	最大	<p>(b) <code>\strut</code> コマンド を使用した表</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td style="padding: 5px;">ふつう, g</td></tr> <tr><td style="padding: 5px;">やや大</td></tr> <tr><td style="padding: 5px;">大</td></tr> <tr><td style="padding: 5px;">極大</td></tr> <tr><td style="padding: 5px;">極々大</td></tr> <tr><td style="padding: 5px;">最大</td></tr> </table>	ふつう, g	やや大	大	極大	極々大	最大
ふつう, g													
やや大													
大													
極大													
極々大													
最大													
ふつう, g													
やや大													
大													
極大													
極々大													
最大													

2. 表

2.14.2. 特殊なセル揃え

ときには表 2.26 の小数点のように、同一列の各セルの内容が、特定の文字で揃えられた方が見栄えの良いことがあります。この揃え方は、当該列にカーソルを置いて、ツールバーボタンを押すか、表設定中で揃えを小数点でに設定します。

表 2.26.: 一列が小数点で揃えられた表セル

見出し
12.6
0.68
-123.0

表 2.27.: 一列が演算子で揃えられた表セル

見出し
12 + 6
0 - 68
-123 / 0

揃え文字を指定するフィールドには、数式を用いることができないので、この方法を使って、数式中の文字で揃えることはできません。このような場合には、表 2.27 のような方法を用いる必要があります。この表は、 4×2 表として作成されており、見出しは中央揃え連結列です。第 1 列には、右揃えで演算子を含む前半部を収め、第 2 列には、左揃えで残りを収めます。演算子の周りに通常加えられる空白を得るために、第 2 列の各セルの冒頭には、小空白が加えられています。通常、表の列間に加えられる余白を削るために、

@{1}

という L^AT_EX 引数を第 2 列に指定します。

表 2.28 は、揃え方の例をいくつか示しています。関係子で揃えるために、表 2.27 の冒頭には、小空白が加えられています。

表 2.28.: 表セルの揃えの例

単位	冪数	関係子
12×24 本の瓶	$10 \cdot 10^{-17}$	$\Gamma(t) \propto \Upsilon(t)$
1024×768 ピクセル	$5.78 \cdot 10^7$	$A \neq B_{\text{red}}$
32×6 cm	$-33.5 \cdot 10^4$	$\sin(\alpha) \geq \sin(\beta)$

2.14.3. 自製のセル書式および列書式

第 2.13 節のようにして通常列の必要幅を計算したり、第 2.8.2.2 節のように連結列の必要幅を計算したりするのは、計算の必要な表が複数あるときには、とても大変です。そこで楽をするために、プリアンブルでセル・列書式を定義してしまえば、それを文書中のすべての表で使うことができます。書式は、以下のように定義します。

`\newcolumnntype{書式名}[引数の数]{コマンド}`

書式名は、一文字だけでも構いません。 *b*・*c*・*l*・*m*・*p*・*r* の文字は定義済みですので使用することができませんが、これらの文字の大文字は使用可能です。

固定幅の縦横中央揃えの連結列セルの場合は、以下のセル書式を定義するとよいでしょう。

`\newcolumnntype{M}[1]{>{\centering\hspace{0pt}}m{#1}}`

ここで `\hspace{0pt}` は、第 2.8.1 節で述べたように、最初の単語がハイフネートされる問題を回避するためのものです。すると連結列を作成する際には、表ダイアログの \LaTeX の引数で

`M{width}`

と入力するだけですみます。

連結列セルにまたがられる各セルについては、以下の書式を定義するとよいでしょう。

`\newcolumnntype{S}[2]{>{\centering\hspace{0pt}}m{(#1+(2\tabcolsep+\arrayrulewidth)*(1-#2))/#2}}`

この書式は、またがられる各セルが同じ幅になる幅を計算するのに、第 (2.1) 式を使用しています。

こうしておけば、当該セルの \LaTeX 引数に

`S{連結列セルの幅}{連結列セルがまたがる列の数}`

と入力するだけですみます。

色付き列については、以下のように定義するとよいでしょう。

`\newcolumnntype{K}[1]{>{\columncolor{#1}\hspace{0pt}}c}`

最後の「c」は、文字列が水平方向中央揃えになった可変幅の列を生成します。こうしておいて、 \LaTeX 引数に

`K{色名}`

と入力します。

表 2.29 を生成するには、第 1 列と連結列に \LaTeX 引数として

`M{2.5cm}`

2. 表

を指定し、最終列に

```
K{red}
```

そして第2列の各セルに

```
S{2.5cm}{2}
```

と指定します。

表 2.29.: ユーザー定義の表書式を使用した表

とても長い表 セル中の単語	複数行の連結 列		は
に	ほ	へ	と
ち	り	ぬ	る

2.14.4. 罫線幅

表中のすべての罫線の罫線幅を調整するには、長さ`\arrayrulewidth`を使用します。たとえば、表 2.30 のように罫線幅を 1.5 pt にするには、表や表フロートの前に $\text{T}_{\text{E}}\text{X}$ コードで

```
\setlength{\arrayrulewidth}{1.5pt}
```

というコマンドを挿入します。こうして変更した罫線幅は、これ以降のすべての表に適用されます。ふたたび既定値を使用するには、表か表フロートの前に $\text{T}_{\text{E}}\text{X}$ コードで `\arrayrulewidth` を 0.4 pt に設定してください。

表 2.31 のように横罫だけを 1.5 pt 幅にするには、表または表フロートの前に $\text{T}_{\text{E}}\text{X}$ コードとして以下のコマンドを挿入してください。

```
\let\myHline\hline  
\renewcommand{\hline}  
{\noalign{\global\arrayrulewidth 1.5pt}  
\myHline\noalign{\global\arrayrulewidth 0.4pt}}
```

既定幅に戻すには、表または表フロートの後に $\text{T}_{\text{E}}\text{X}$ コードとして以下のコマンドを挿入します。

```
\renewcommand{\hline}{\myHline}
```

表 2.30.: 1.5 pt 幅罫線の表

sd		
	sd	
		sd

表 2.31.: 1.5 pt 幅横罫の表

sd		
	sd	
		sd

表 2.32.: 1.5 pt 幅縦罫の表

sd		
	sd	
		sd

縦罫だけを 1.5 pt 幅にするには、第 2.14.3 節の説明にしたがって、文書プリアンブルに以下のような列書式を作成してください。

```
\newcolumntype{V}{!{\vrule width 1.5pt}}
```

表 2.32 を作るには、最右列に L^AT_EX 引数

VcV

を指定し、他の列には

Vc

を指定します。

2.14.5. 点罫

表 2.33.: 点罫を使用した表

い	ろ	は	に	ほ
へ	と	ち	り	ぬ
る	を	わ	か	よ
た	れ	そ	ち	
り	な	ら	む	う

L_AT_EX は点罫を自前ではサポートしません。そこで T_EX コードを使用する必要があります。必須要件として、文書プリアンブル中に

```
\usepackage{arydshln}
```

というコマンドを書き、L^AT_EX パッケージの `arydshln` を読み込んでおく必要があります。縦罫を点罫にするには、表セルダイアログの L^AT_EX 変数として、水平揃え用の文字とともにコロン「:」を入力してください。

2. 表

横点罫を得るには、表中当該行の最初のセルの冒頭に、 $\text{T}_{\text{E}}\text{X}$ コードで

```
\hdashline
```

というコマンドを加えてください。

連結列に点罫を書くには、表中当該行の最初のセルの冒頭に、 $\text{T}_{\text{E}}\text{X}$ コードで

```
\cdashline{列番号}
```

なるコマンドを使用してください。たとえば、第2列から第4列にわたる連結列があって、その上に点罫が欲しい場合には、連結列のある行の最初のセルの冒頭に

```
\cdashline{2-4}
```

なるコマンドを置いてください。

表 2.33 は、第3列の $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 変数に「:c」を指定して作られています。第3行の最初のセルには、 $\text{T}_{\text{E}}\text{X}$ コードのコマンド `\hdashline` が挿れられており、第4行の最初のセルには、 $\text{T}_{\text{E}}\text{X}$ コードのコマンド

```
\cdashline{4-5}
```

が挿れられています。

点罫を持つ表も、表 2.34 のように色付けをすることができます。

表 2.34.: 点罫線の色付き表

い	ろ	は	に	ほ
へ	と	ち	り	ぬ
る	を	わ	か	よ
た	れ	そ	つ	
ね	な	ら	む	う

行や列、罫線の着色は、第 2.11 節に述べられているように動作します。水平線の着色も、`\myHlineC` コマンドを `\myHdashline` に置き換え、`\hline` を `\hdashline` に置き換えれば、そこで述べられているのと同様に機能します。したがって、以下のようなコマンドを使います。

```
\let\myHdashline\hdashline
\renewcommand{\hdashline}
{\arrayrulecolor{red}\myHdashline\arrayrulecolor{black}}
```

`\cdashline` に色をつけるならば、コマンドは

```
\let\myCdashline\cdashline
\renewcommand{\cdashline}[1]
{\arrayrulecolor{red}\myCdashline{#1}\arrayrulecolor{black}}
```

のようになります。黒の罫線に戻すには、以下のコマンドを表の後ろに $\text{T}_{\text{E}}\text{X}$ コードで挿入する必要があります。

`\hdashline` の場合は

```
\renewcommand{\hdashline}{\myHdashline}
```

とし、`\cdashline` の場合は

```
\renewcommand{\cdashline}{\myCdashline}
```

としてください。点線の間余白に色をつけるには、表の前に $\text{T}_{\text{E}}\text{X}$ コードで

```
\dashgapcolor{色}
```

とすれば良いです。ここで「色」は、定義済みの色名です。

```
\nodashgapcolor
```

というコマンドは、`\dashgapcolor` が行った変更を元に戻します。

[註] 1つのセル中に複数のコマンドがある場合、`arydshln` のコマンドが、そのセル中で最初に来る必要があります。したがって、表 2.34 のセル k では、`\hdashline` コマンドが `\rowcolor` の前に来ています。

[註] $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ パッケージ `arydshln` は、文書プリアンブル中で $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ パッケージ `colortbl` よりも前に読み込まれている必要があります。

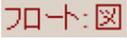
点罫について詳しい情報は、`arydshln` の取扱説明書 [4] をご覧ください。

3. フロート

3.1. はじめに

フロートは、何かしらのラベルに関連付けられた文章の塊で、固定された居場所を持ちません。フロートはもっとも適した場所を求めて、1 頁ないし 2 頁、前後にただよう（フロート）ことができます。脚註や傍註も、当該ページに註が多すぎる場合には次頁にただよって行きますので、フロートの一種です。

フロートを使うと、質の良いレイアウトを組むことができます。図表が均等に各ページに割り当てられるので、文章のない空白やページが出ることを避けることができます。このようなフローティングは、文章と図表のあいだの連関を破壊してしまうことがあるので、各フロートは文中から参照することができるようになっています。このため、フロートには連番が割り当てられます。参照については、第 3.4 節で述べられています。

フロートを挿入するには、**挿入▷フロート**メニュー（またはツールバーボタン）を使えば、ラベルのついたボックスすなわちキャプション差込枠が挿入されます。このラベルは、出力の際に自動的に文書言語に翻訳されます。ラベルの後ろには、キャプションの文章を入れ、画像ないし表は、フロート中、キャプションの上ないし下に別の段落として挿入します。キャプションの配置に関して詳しくは、第 3.10 節に述べられています。また、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 文書を読みやすくするために、ボックスラベルを左クリックすれば、フロートボックスを広げたり畳んだりすることができるようになっています。畳んだフロートボックスは、のように、灰色ボタンに赤字ラベルで表示されます。

周囲の文章に特殊な整形がなされているときには $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ エラーが発生し得るので、これを避けるために、フロートは独立した段落として挿入することをお勧めします。

既存の図表は、選択した後に新規フロートのツールバーボタンを押せば、フロート中の図表にすることができます。

3.2. フロート型

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ には、第 1.2 節および第 2.5 節で各々説明されている図フロートと表フロート以外に、**アルゴリズム型**と**折返し型**のフロート型があります。後者は、第 6.1 節に説明があります。

3. フロート

3.2.1. アルゴリズムフロート

アルゴリズム 3.1 アルゴリズムフロートの例

```
for I in 1..N loop
  Sum:= Sum + A(I); /*comment*/
end loop
```

このフロート型は、**挿入▷フロート▷アルゴリズム** メニューを使うと挿入することができます。これは、プログラムコードやアルゴリズムの記述に用いられ、第8章で説明されているプログラムコードリストの代わりとみなすことができます。『LyX ユーザーの手引き』に述べられているように、アルゴリズム向けの環境としては、LyX コードがあります。アルゴリズム 3.1 は、アルゴリズムフロートの一例であり、下の罫線が最後の文字行のちょうど下に来るように、フロートの終わりに-4 mm の垂直空白が入れてあります。

フロートラベルは、自動的に文書言語に翻訳されるわけではありません。使用中の文書が英語でない場合には、文書プリアンブルに

`\floatname{algorithm}{付けたい名称}`

という行を手動で加える必要があります。ここで**付けたい名称**には、お使いの言語で「*algorithm*」に対応する単語を指定します。

アルゴリズム一覧を挿入するには、文書言語が LyX のメニューに表示されている言語と同じであるならば、**挿入▷一覧/目次▷アルゴリズム一覧** メニューを使用します。それ以外の場合には、代わりに

`\listof{algorithm}{付けたい名称}`

というコマンドを TeX コードとして用います。ここで**付けたい名称**とは、「*List of Algorithms*」という単語に相当する、ご使用になりたい言語での単語です。

アルゴリズムフロートは、多くの文書クラスにおける図表フロートとは違って、「chapter.algorithm」スキームでは既定で付番されていません。このスキームでアルゴリズムフロートに連番を振るには、使用している文書のプリアンブルに

`\numberwithin{algorithm}{chapter}`

というコマンドを追加してください。 `\numberwithin` というコマンドを使用するには、文書設定の数式オプションタブで AMS math パッケージを使用するオプションを有効にしてください。

3.3. フロートの連番

フロートは通常、フロートのある節とは関係なく番号が振られるか、あるいは「章. 番号」や「節. 番号」といった形で番号が付けられます。これは、使用している文書クラスによって変わります。

節から独立して付けられている連番を変更するには、文書プリアンブルで

```
\renewcommand{\thetable}{\roman{table}}
```

というコマンドを使うと良いでしょう。 `\thetable` は表番号を印字するコマンドで、図フロートの場合であれば、代わりに `\thefigure` コマンドを使用します。上のコマンドで使用されている `\roman` は、表番号を小文字のローマ数字で印字するコマンドです。

連番の付け方をたとえば「小節. 番号」に変更するには、プリアンブルで

```
\numberwithin{table}{subsection}
```

というコマンドを使用してください。 `\numberwithin` コマンドを使用可能にするためには、文書設定の **数式オプション** タブで、AMS math 数式パッケージを使用するを使用します。

連番コマンドについての詳細と大切な注意点については、第 4.2.1 節もご覧ください。

3.4. フロートの参照

フロートを参照するには、**挿入**▷**ラベル**メニューかツールバーボタンを使用して、フロートのキャプションにラベルを挿入してください。 **enu.第2項目** のような灰色のラベルボックスが挿入され、ラベルウィンドウがポップアップして、ラベルの字句を入力するよう促します。LyX は、キャプションの最初の数単語に前置句をつけたものを候補として示します。前置句はフロート型に依存し、たとえば図フロートには「fig:」という前置句がつけられます。

ラベルは、参照のための名称兼位置標として用いられます。 **挿入**▷**相互参照**メニューかツールバーボタンを使えば、ラベルを参照することができます。 **参照:enu.第2項目** のような灰色の相互参照ボックスが挿入され、文書中のすべてのラベルを表示した相互参照ウィンドウが現れます。複数の LyX 文書が開かれている場合には、ダイアログの上方にあるドロップリストから作業中の文書を選択してください。ここでラベルを整理してから選ぶこともできます。出力中では、相互参照ボックスの位置にはフロート番号が表示されます。

相互参照名と数字の間に改行が入るのを防ぐために、両者の間の空白には非改行空白を用いた方が良いでしょう。もし相互参照が存在しないラベルを参照している場合には、出力では、参照の代わりに二つの疑問符が表示されることになります。

ラベルは、ラベルボックスをクリックすることによって、いつでも変更することができます。変更されたラベルへの参照は、自動的に新しいラベル文へのリンクに変更されますので、リンクの更新に手間を割く必要はありません。

相互参照ウィンドウの **ラベルへ移動** ボタンは、参照されているラベルの前にカーソルを移動します。すると、ボタンの文字が戻るに変わるので、相互参照にカーソルを戻すのに使用することができます。

3. フロート

3.4.1. 相互参照の書式

相互参照には、以下の6種類があります。

<参照>: これはフロート番号を表示します。これが既定値です。例：1.3

(<参照>): これはフロート番号を2つのかっこの中に表示します。これはふつう数式を、とくに参照名「Equation」を省略して参照する際に用いられる様式です。例：(2.1)

<参照ページ>: これはページ番号を表示します。例：Page 4

on page <参照ページ>: これは「on page」という文とページ番号を表示します。例：4 ページ

<参照> on page <参照ページ>: これはフロート番号と「on page」という文とページ番号を表示します。例：1.3 4 ページ

整形された参照: これは自製の相互参照書式で出力します。

[註] この機能は、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ パッケージ `prettyref` または `refstyle` が導入されている場合のみ使用することができます。

文書▷設定▷文書クラスメニューでオプション**相互参照**に(`prettyref`ではなく)`refstyle`を使用を設定すると、この機能で使用する $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ パッケージを選択することができます。書式は、文書プリアンプルに`\newrefformat`コマンド(`prettyref`の場合)あるいは`\newref`コマンド(`refstyle`の場合)を書き加えることによって指定することができます。たとえば、図への参照(これは短絡ラベル「fig」を持ちます)をすべて定義し直すには、

```
\newref{fig}{refcmd={Image on page \pageref{#1}}}
```

というコマンドを用います。書式についての詳細は、パッケージの取扱説明書[18, 20]をご覧ください。

複数形と大文字化のオプションは、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ パッケージ `refstyle` を使用している時のみ、利用可能になります。複数形は、例えば「images」のように単語に *s* を付けるものであり、**大文字化**は「Image」のように単語を大文字にするものですので、日本語文書には有用ではありません。

名称参照: これは、下のような参照のキャプションを出力します。

二つの像。両画像とも「distorted」画像設定グループに属しています。

ラベルのみ これは、fig: 歪められた二つの像のように参照の $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ラベルのみ出力します。

これによって、ラベルを $\text{T}_{\text{E}}\text{X}$ コード中でしようにできるようになります。**接頭辞なし**オプションを使えば、ラベル名の接頭辞を歪められた二つの像のように取り除くことができます。

<参照ページ>は、ラベルが同一あるいは前後ページにある場合には、ページ番号を出力しません。その代わりに「on this page」のような文章が表示されます。様式<参照> on page <参照ページ>では、ラベルが同一ページにある場合には、ページに関することは出力されません。

出力中で参照されている頁番号と文書中の位置は、 \LaTeX が自動的に計算します。参照の種類は、相互参照ボックスをクリックすると現れる相互参照ウィンドウ中の書式フィールドで選ぶことができます。

【註】 \LaTeX パッケージ `prettyref` は、 LyX の取り得るラベルショートカット¹を全部は解釈できず、国際化もされていないので、`refstyle` を使用することを推奨します。

3.4.2. 参照名の自動命名

文書設定ダイアログの PDF 特性で有効にされた `hyperref` \LaTeX パッケージは、相互参照が参照されたフロート（あるいは節のような文章中の箇所）の名称を自動的に取り入れてくれるという、たいへん便利な機能を提供します。つまり、図への参照毎に前に「図」と書き込む必要はなくなるのです。この機能を使用するには、`hyperref` を有効にして、

```
\AtBeginDocument{\renewcommand{\ref}[1]{\mbox{\autoref{#1}}}}
```

という行を \LaTeX プリアンブルに挿入してください。たとえば「section」の代わりに「sec.」としたいなど、既定の参照名以外のものが欲しい場合には、

```
\addto\extrasjapanese{\renewcommand{\sectionautorefname}
{sec.\negthinspace}}
```

のようにプリアンブルに挿入することで参照名を再定義することができます。英語以外の文書言語を使用している際には、

`\extrasjapanese` を `\extras***` に置き換えてください（***は使用中の言語名）。

自動命名を行いたいけれども、ある参照型以外、たとえば数式以外に対して行いたいときには、以下のようなプリアンブルコードを書いてください。

```
\addto\extrasjapanese{\renewcommand*{\equationautorefname}[1]{}}
```

この話題に関する詳細は、`hyperref` の説明書 [11] をお読みください。

【註】 参照の自動命名は、相互参照を整形参照様式（第 3.4.1 節）で使用しているときには、使用することができません。

LyX の取扱説明書『数式篇』では、参照名の自動命名が使用されています。

3.4.3. 参照位置

文書設定ダイアログの PDF 特性で、`hyperref` を使用して出力中で相互参照をリンクするようにした場合には、画像フロート参照をクリックすると、画像ラベルに移動できるようになります。キャプションは、スクリーン上のテキストの最初の部分なので、スクロールしないと画像を見ることができません。これは、参照リンク標がラベルの位置に置かれるからです。 \LaTeX パッケージの `hyperref` の一部である `hyperref`

¹例えば、「fig:」は図キャプションを表す LyX ショートカットです。

3. フロート

パッケージを使うと、リンク標はフロートの最初に置かれます。この機能を図フロートに対して使用するには、文書プリアンブルに

```
\usepackage[figure]{hycap}
```

という行を書いて、hycapを読み込んでください。hycapをすべてのフロート型に対して使用することもできますが、安定性の問題から推奨できません。詳しくは、hycapの説明書 [10] をご覧ください。

[註] hycap は内部図への参照には効力がありません。

3.5. フロートの設定

LyXにおけるフロート設定は、2箇所で行うことができます。文書▷設定...▷フロートの設定では、フロートのローカルな設定を変えない限り、文書中の全フロートの既定値として用いられるオプション（ページ上の配置および内容の揃え）を設定することができます。

もう一つの方法は、フロートボックスを右クリックして、設定...をクリックする方法です。すると、ダイアログが開いて、対象となるフロートの大域的な配置や揃えその他の選択ができます。文書▷設定...▷フロートの設定とローカルなフロートの設定両方について、これらのオプションを説明していきます。

3.5.1. フロートの配置

文書▷設定...▷フロートの設定（すべてのフロート対象）とローカルなフロートの設定（特定のフロート対象）の両方において、フロートをページ上に配置するのにL^AT_EXが用いる配置アルゴリズムを修正することができます。これは配置の設定で行います。既定では、これは文書▷設定...▷フロートの設定では文書クラスの既定値になっており（つまり文書クラスの既定配置アルゴリズムが使用されます）、ローカルのフロートの設定ダイアログでは文書の既定値になっています（すなわち文書▷設定...▷フロートの設定の設定——既定では文書クラスの既定値——が使用されます）。これを変更したい場合には、配置の設定を詳細設定にしてください。これによって、下記から1つ以上選択できるようになります。

可能ならば現在位置に： フロートを挿入位置に配置することを試みます

なんとしても現在位置に： フロートの配置を挿入した位置に強制します

ページ上部： フロートを現在ページの上部に配置することを試みます

ページ下部： フロートを現在ページの下部に配置することを試みます

フロートを独立したページに： フロートを独立したページに配置することを試みます

L^AT_EX は、つねに上記のオプション順を用います。すなわち、既定の配置を使用すると、L^AT_EX はまず**可能ならば現在位置に**を試み、次に**ページの上**部、その後他のオプションを試みます。既定値を使わないときには、L^AT_EX は有効化されたオプションのみを使用しますが、やはり同じ順序でこれらを試みます。これら四つの配置がいずれも無理な場合は、フロートを内部的に次のページにおいてみて同様の手続きがうまくいくかどうかを試みます。

既定では、各オプションはそれぞれ以下の規則を持っています。

ページ上部 は、フロートのページに占める割合が7割以下の場合のみ、ページ上部に配置します (`\topfraction`)。

ページ下部 は、フロートのページに占める割合が3割以下の場合のみ、ページ下部に配置します (`\bottomfraction`)。

フロートを独立したページ には、フロートのページに占める割合が5割以下の場合のみ、複数のフロートを同一ページに配置します (`\floatpagefraction`)。

これらの規則に従いたくないときは、L^AT_EX の規則を無視するオプションを追加することによって、これらは無視することができます。

上記説明の後ろに括弧でくくってある L^AT_EX コマンドを使うと、これらの規則を定義しなおすこともできます。たとえば、往々にして小さすぎる、ページ下部規則の既定値を、ページの5割に変えるには、文書プリアンブルに

```
\renewcommand{\bottomfraction}{0.5}
```

という行を加えてください。

ときにはフロートを、挿入位置に正確に、しかもかならず配置させたい場合があるかもしれません。このような場合には、**何としても現在位置に置く**オプションを使用します。このオプションはできる限り使わないようにし、使う場合でも印刷する直前にまで文書が仕上がってから、使用するようにしてください。なぜならば、このオプションを使うことによって、フロートはもはや「漂う」ことができなくなりますから、文書に変更を加えてしまうと、ページのレイアウトが往々にして破壊されてしまうからです。

折返しフロートは、つねにどれかの段落の本文で囲まれますので、配置オプションはありません。

対応する節がページの中程から始まるのというのに、フロートがページの先頭に置かれてしまうために、読者がフロートを前節の一部であると勘違いしてしまうという問題に直面することがあります。この問題を避けるには、L^AT_EX コマンドの`\suppressfloats`を使用することができます。これは、これが挿入されたページの特定のフロートの配置を抑制するので、節が始まる前にフロートが設定されてしまうのを避けるのに使用することができます。これを用いるには、文書プリアンブルに以下のコマンドを加えてください。

```
\let\mySection\section
\renewcommand{\section}{\suppressfloats[t]\mySection}
```

3. フロート

これと同じものは、章や小節など、すべての節見出しに用いることができます。この定義は、小々節のような小さい文字の部分では、 \LaTeX が適当な配置を見つけるのに失敗することがあるので、これらに対する定義はお勧めしません。

場合によっては、すべての図表を文末に置く必要があることがあります。この目的のために、 \LaTeX パッケージ `endfloat` が開発されました。このパッケージは、すべての図表を文末に独立した節として置きます。元フロート位置には、“[Figure 3.2 about here.]” というようなヒント文が挿入されます。`endfloat` パッケージは、プリアンブルに

```
\usepackage[options]{endfloat}
```

という行を加えることで読み込まれます。このようにして生成された図表の節を整形するために、さまざまなパッケージオプションがあります。詳細は、`endfloat` 取扱説明書 [8] をご参照ください。

[註] `endfloat` は、ヒント文の自動翻訳を行わないので、手動で行わなくてはなりません。[8] の第4節をご参照ください。

[註] 現在のところ、`endfloat` にはキャプションにドイツ語の「B」が含まれているときにうまく動かないバグがあります。この場合には、「B」の代わりに \TeX コードで `\ss` コマンドを使用してください。

フロートの割付に関する詳細は、 \LaTeX 関連書 [1, 2, 3] をご覧ください。

3.5.2. フロートの中身の揃え

このオプションリストでは、フロート（あるいは **文書▷設定** から呼び出したのであれば、すべてのフロートの既定値）の中身をどう揃えるかを定義します。**文書クラス** の既定値では、そのクラスの揃え、すなわち \LaTeX の既定の揃えが使用されます。**文書の既定値**（フロートの設定ダイアログでのみ表示され、**文書▷設定** では表示されません）は、**文書▷設定▷フロートの設定** で指定した設定が使われます。他の3つの選択肢、**左揃え・中央揃え・右揃え** は自明でしょう。**文書▷設定▷フロートの設定** でこれらが選択されれば、文書中のすべてのフロートがこの設定を持ちます（すると、たとえばすべてのフロートを中央揃えにするのは極めて簡単です）。

3.5.3. 他のフロートオプション

これらの2つのオプションは、**フロートの設定ダイアログ**でのみ表示され、**文書▷設定**では表示されません。

段をまたぐオプションは、二段組文書でのみ有効です。これを選択すると、フロートは段の中のみならず、ページの間段にまたがって広がります。

横倒しに回転オプションは、フロートを回転するのに用います。第3.6節をご参照ください。

3.6. 回転フロート

特に横長の表などの場合、フロートを回転させたいことがあります。キャプションを含むフロート全体を回転させたいときは、フロートボックスを右クリックして、横倒しに回転オプションを有効にします。

回転フロートは、つねに独立したページ（二段組文書のときは独立した段）に配置されます。フロート設定の段をまたぐオプションを有効にすれば、複数の段にまたがるようにすることができます。また、回転フロートは、外側余白方向から読むことができるような形に回転されます。全ページにわたって回転方向を強制したい場合には、文書クラスオプションに `figuresleft` オプションあるいは `figuresright` オプションを加えることができます。

回転フロートへの相互参照は、通常のコロットの場合同じです。キャプションの形式も同じです。表 3.1 が回転表フロートの例です。

[註] すべての DVI ビューアが回転フロートを表示できるわけではありません。

3.7. 従属フロート

従属フロートは、図が複数の画像から成るような場合に使用します。これは、フロートの中に新たなフロートを挿入することで作ることができます。従属フロートの揃えは、表 3.2 と表 3.3 に示されているように段落と同様に指定することができます。

従属フロートの参照は、通常のコロットの参照と同様にすることができます。例：表 3.2a と表 3.2b は表 3.2 の従属フロートです。

3.8. 横並びのフロート

図 3.1 や図 3.2 のようにフロートを横並びに配置するには、**ひとつのフロートのみ**を使用し、その中に二つの `minipage` ボックスを挿入します²。幅は 45-50 列幅%とし、各 `minipage` のボックス配置を下に設定します。 `minipage` ボックスには、フロートと同じように画像とキャプションを入れます。フロートと唯一違う点は、画像単位列幅%がここでは `minipage` ボックスの幅によって計算されることです。

2つのフロートを横に並べるには、可変幅ミニページを使うと簡単にできます(第 5.4 節参照)。可変幅ミニページは、図の幅に自動的に調整されます。明示的に幅を指定する必要もなく、図の水平方向の配置も調整してくれます。ファイル `examples/varwidth-floats-side-by-side` に、この方法の例があります。

²Minipage は第 5.4 節に説明があります。

3. フロート

表 3.1.: 回転した表

試験	甲	乙	丙	丁
----	---	---	---	---

表 3.2.: 従属表を横に並べたもの

(a) これは従属表 a です.

試験	甲	乙	丙	丁
----	---	---	---	---

(b) これは従属表 b です.

丁	丙	乙	甲	試験
---	---	---	---	----

表 3.3.: 従属表を縦に並べたもの. (a) は 4セルから成る表, (b) は 5セルから成る表.

(a)

試験	試験	試験	試験
----	----	----	----

(b)

甲	乙	丙	丁	戊
---	---	---	---	---

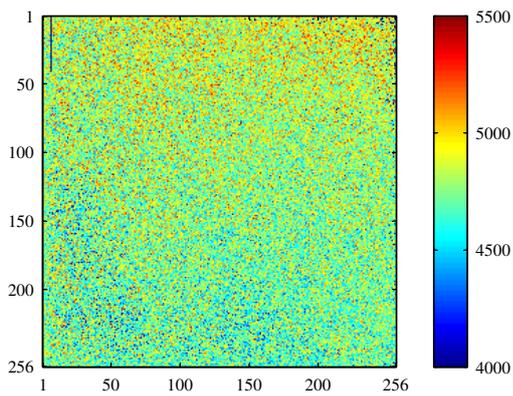


図 3.1.: 左側のフロート

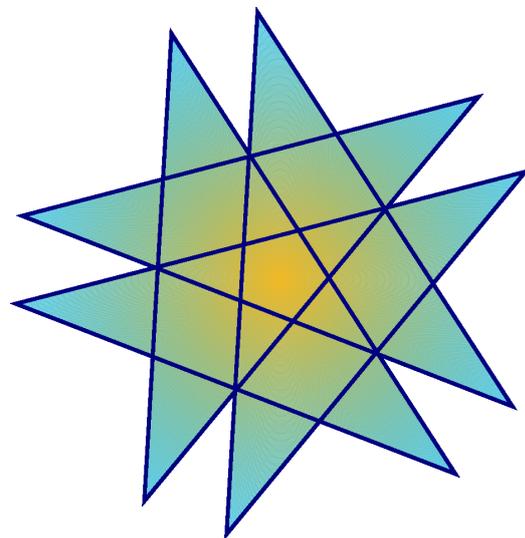


図 3.2.: 右側のフロート

3.9. キャプションの書式設定

キャプション環境は、フロートの既定の段落環境です。キャプションは、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 画面上では、例えば「図#:」のようなラベルの後に、キャプション文を伴って表示されます。ここで「#」には、実際には参照番号が入ります。既定の設定では、ラベルと番号はキャプション文と同じフォントになっており、ラベルと文を分けるために、番号の後にコロンが入ります。このキャプション書式は、すべての文書書式に適しているわけではありません。

既定のキャプション書式を変更するには、文書プリアンブルに

```
\usepackage[format definition]{caption}
```

という行を加えて、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ パッケージ `caption` を読み込んでください。たとえば、ラベルと番号をサンセリフのボールド体にして、表キャプションを、この文書のように常に表の上に置くようにするには、

```
\usepackage[labelfont={bf,sf}, tableposition=top]{caption}
```

というコマンドを使用してください。また、各フロート型毎に別々のキャプション書式を定義することもできます。この場合には、`caption` パッケージを書式関連オプション抜きで読み込み、文書プリアンブルに

```
\captionsetup[フロート型]{書式定義}
```

というコマンドを使用して、個別の書式を指定してください。たとえば、図 3.3 と表 3.4 に使われているキャプション書式は、文書プリアンブルに以下のコマンドを指定することで作り出すことができます。

```
\usepackage[tableposition=top]{caption}
\captionsetup[figure]{labelfont={tt}, textfont=it, indentation=1cm,%
                    labelsep=period}
\captionsetup[table]{labelfont={bf,sf}}
```

[註] オプションのうち、`tableposition=top` は、文書クラスに **KOMA-Script** が使われているときには、無効になります。この場合には、文書クラスオプション `captions=tableheading` を指定しなくてはなりません。

`caption` パッケージについての詳細は、その取扱説明書 [6] をご覧ください。

ラベル名を、たとえば「図」から「画像」に変更するには、以下のプリアンブルコマンドを使用してください。

```
\renewcommand{\fnum@figure}{画像~\thefigure}
```

なお、`\thefigure` は表番号を挿入し、「~」は非改行空白を入れます。

KOMA-script 文書クラス (**KOMA-Script Article**・**KOMA-Script Book**・**KOMA-Script Letter**・**KOMA-Script Report**) を使用している場合には、`caption` パッケージの代わりに **KOMA-script** のビルトインコマンド `etkomafont` を使用することもできます。たとえば、キャプションラベルをボールド体にするには、以下のコマンドを文書プリアンブルに加えます。

```
\setkomafont{captionlabel}{\bfseries}
```

`\setkomafont` についての詳細は、KOMA-Script の取扱説明書 [12] をご覧ください。

3.10. キャプションの配置

通常、キャプションの配置原則は、

図の場合: キャプションを図の下に置く

表の場合: キャプションを表の上に置く

という風になっています。L^AT_EX の標準クラスでは、キャプションを表の上に置くことは、残念ながらサポートされていません。したがって、`article`・`book`・`letter`・`report` といった文書クラスを使用している時には、キャプションと表の間に余白が置かれなくなってしまいます。本来あるべき余白を挿入するには、お使いの文書プリアンブルで LaTeX パッケージ `caption` を読み込むコマンドに

```
tableposition=top
```

というオプションを追加してください³。KOMA-Script 文書クラス (KOMA-Script Article・KOMA-Script Book・KOMA-Script Letter・KOMA-Script Report) をお使いの場合には、`caption` パッケージにオプションを与える代わりに、文書クラスオプション

```
captions=tableheading
```

を設定することもできます⁴。

KOMA-Script 文書クラスの文書では、キャプションを右クリックして、文脈依存メニューから、これが図表の上または下に来るべきキャプションかを指定することもできます。これによって、文書全体のキャプション配置指定にかかわらず、特定のキャプションの配置を変えることができます。表 3.5 は、キャプションは、L^AT_EX 中では上に置かれているけれども、表の下に来るように指定されている例です。比較のために、表 3.6 のキャプションでは、指定が為されていません。

また、キャプションは、図表の横に置くこともできます。これを実現するには、文書プリアンブルに

```
\usepackage[option]{sidecap}
```

という行を入れて、L^AT_EX パッケージ `sidecap` を読み込まなくてはなりません。オプションを与えなければ、ページの外側余白の側に—偶数ページでは右側、奇数頁では左側に—キャプションが置かれます。`innercaption` オプションを与えれば、内側余白側に配置を変更することができます。配置をつねに右側ないし左側に来るように強制するには、`rightcaption` ないし `leftcaption` オプションを用います。

本節の残りの部分を出力で表示するには、L^AT_EX パッケージ `sidecap` を導入する必要があります。

³`caption` パッケージについての詳細な情報は、第 3.9 節をご覧ください。

⁴このオプションは本文書で使用されています。

3. フロート

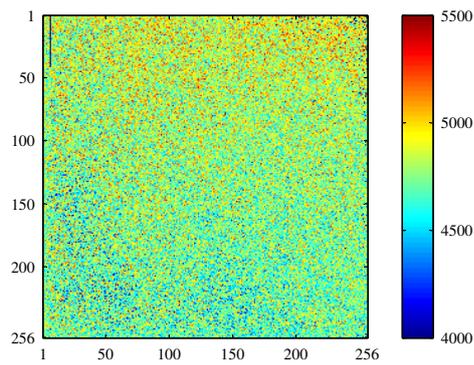


図 3.3.. これは、別のキャプション書式を例示するための、一行よりも長い図キャプションです。ここでは、自製キャプション書式が使用されています。

表 3.4.: これは、別のキャプション書式を例示するための、一行よりも長い表キャプションです。ここでは、この文書での表キャプションの標準書式が使用されています。

い	ろ	は	に	ほ
---	---	---	---	---

表 3.5.: 表下に来るように指定されたキャプション

1	2	3
森嶋	宇沢	二階堂
$\int x^2 dx$	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$	$1 + 1 = 2$

表 3.6.: 標準の表キャプション

1	2	3
森嶋	宇沢	二階堂
$\int x^2 dx$	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$	$1 + 1 = 2$

3.11. フロート一覧

文書中の各節が列挙されたものである目次と同様、文書中の図など、すべてのフロート型に一覧があります。これらは、**挿入▷一覧/目次**メニューで挿入することができます。

一覧の見出しは、フロートのキャプションないし短縮タイトル、フロート番号、これらが文書中で現れるページ番号で構成されます。

本文書の終わりに、図一覧と表一覧があります。

4. 註釈

4.1. LyX 註釈

註釈は、ツールバーボタンか、**挿入**▷**註釈**メニューで挿入することができます。註釈には、以下の三つの型があります。

LyX 註釈 この註釈型は、内部用途の註釈で出力には現れません。註釈ボックスは、以下のように表示されます。

註釈 これは、出力には現れない註釈ボックスに書かれた文章です。

コメント この註釈も出力には現れませんが、**ファイル**▷**書き出し**▷**LaTeX (pdflatex) / (平文)**メニューを使って文書を \LaTeX にエクスポートする際には、 \LaTeX コメントとして出力されます。註釈ボックスは、以下のように表示されます。

コメント これは、LaTeXファイルのコメントのみに現れる註釈ボックスに書かれた文章です。

淡色表示 この註釈は、薄灰色の文章として出力に現れます。註釈ボックスは、以下のように表示されます。

淡色表示 これは、出力に薄灰色の文章として現れるコメントに書かれた文章 **脚注 1**です。

これは、出力に薄灰色の文章として現れるコメントに書かれた文章¹です。

この例から分かるように、淡色註釈の最初の行は少し字下げされるほか、淡色註釈には脚註を入れることができます。

註釈を挿入するのにツールバーボタンを使用すると、**LyX 註釈**が挿入されます。註釈ボックスを右クリックすれば、五つの註釈型を切り替えることができます。既存の平文を註釈に変えるには、平文を選択して、註釈ツールバーボタンをクリックしてください。註釈を平文に変更するには、カーソルを註釈の先頭に置いてBackspaceキーを押すか、カーソルを註釈の最後に置いてDeleteキーを押してください。

淡色註釈の文字色は、**文書**▷**設定**▷**色**メニューで変更することができます。

¹これは、淡色註釈内に脚註を用いた例です。

4. 註釈

4.2. 脚註

脚註は、ツールバーボタンか、**挿入**▷**脚註**メニューを使って挿入することができます。すると、脚註文を挿入したところ **脚註 1** のような脚註ボックスが現れます。もし既存の文章を脚註に変えたいならば、その文章を選択して、脚註ツールバーボタンをクリックしてください。脚註を平文に変えるには、カーソルが脚註の先頭にある時に Backspace キーを押すか、カーソルが脚註の最後にある時に Delete キーを押します。これが脚註の例です²。

出力に於いては、脚註は、文中の脚註ボックスを置いた位置に上付き数字として表示され、脚註文は現ページ底部に置かれます。脚註番号は L^AT_EX が計算を行い、連番が付けられます。脚註番号が章毎にリセットされるか否かは、使用する文書クラスに依ります。

脚註もフロート同様、参照することができます。脚註にラベルを挿入し、第 3.4 節に説明されているようにして、本文中でこのラベルを相互参照してください。これは脚註 2 の相互参照です。

表題環境における脚註は、通常、記号付きで付番はされず、複数の段落を含むことはできません。たとえば、この文書の表題の脚註をご覧ください。

多頁表中の脚註は、第 2.6.1 節で述べられています。

ミニページ中の脚註は出力されますが、ミニボックスは頁中の頁のようなものなので、ボックス内に独立した連番として出力されます（ミニページについての詳細は、第 5.4 節を参照）。通常の脚註のように頁底部に出力されるミニページを得るには、しかし、それを行うには他の方法があり、まず脚註の代わりに T_EX コードで `\footnotemark{}` コマンドを入れてください。脚註の本文は、ミニページの後で T_EX コード `\footnotetext` コマンドの引数として入力します。

以下の表ではコマンド

```
\footnotetext{ミニページ外の脚註です。}
```

を用いています minipage.

これはミニページ中の例文です。^a

これはミニページ中の例文です。³

^aミニページ中の脚註です。

毎回脚註本文を出力せずに、同じ脚註を何度か参照したい場合には、T_EX コードでコマンド `\footnotemark[番号]` を使用してください。

文章を書いている途中には、繰り返す脚註の数はわからないので、その番号は保存しておかなくてはなりません。以下の脚註例では、脚註 2 の後に、T_EX コードで

```
\newcounter{MyRepeatFoot}  
\setcounter{MyRepeatFoot}{\thefootnote}
```

²これは脚註の一例です。

³ミニページ外の脚註です。

というコマンドを挿入しています。そして脚註マークは、

```
\footnotemark[\theMyRepeatFoot]
```

で作成します。

これは繰り返し脚註の例です。²

直前の脚註を繰り返すには、脚註番号の保存を行わないで、直ちに

```
\footnotemark[\thefootnote]
```

を使用してください。

これは繰り返し脚註の例です。³

4.2.1. 脚註の付番

各節毎に脚註番号を 1 にリセットするには、文書プリアンブルに

```
\@addtoreset{footnote}{section}
```

というコマンドを加えてください。

以下のプリアンブルコマンドを使うと、脚註の付番様式を小文字ローマ数字に変更することができます。

```
\renewcommand{\thefootnote}{\roman{footnote}}
```

これは、ローマ数字を使った脚註です^{iv}。

付番様式を大文字ローマ数字に変更するには、上記コマンドの `\roman` を `\Roman` に置き換えてください。脚註を大文字または小文字ラテン「文字」で付番するには、それぞれ `\Alph` または `\alph` を使用してください。記号を使って付「番」するには、`\fnsymbol` を使ってください。

【註】 この付番方法には 1 文字しか使えないので、ラテン文字を使うと 26 個の脚註しか番号を振ることができません。

【註】 記号を使うと 9 個の脚註しか番号を振ることができません。

付番様式を変更した後、既定様式に戻すには、上記コマンドの `\roman` を `\arabic` に置き換えて使用してください。

脚註を「章番号. 脚註番号」スキームで付番したいときには、使用中文書のプリアンブルに

```
\numberwithin{footnote}{chapter}
```

というコマンドを加えてください。`\numberwithin` コマンドを使用できるようにするためには、文書設定の**数式オプション**タブで AMS math パッケージを使うオプションを有効にしてください。

^{iv}これは、ローマ数字を使った脚註の例です。

4. 註釈

これは、脚註のもう一つの例です^{4.5}。

[註] `\numberwithin` は、脚註番号を常にアラビア数字で出力します。それよりも前にアラビア数字以外に再定義されたものは無効となります。

また、例えば「章番号.`\Roman{footnote}`」スキームを得るには、`\numberwithin` の代わりに以下のコマンドを使用してください。

```
\renewcommand{\thefootnote}{\thechapter.\Roman{footnote}}
```

4.2.2. 脚註の配置

同一ページに複数の脚註がある場合、それらはページ底部に縦方向の余白を空けることなく、並べられます。これを読みやすくするには、例えば以下のようなコマンドをプリアンブルに足して、1.5 mm の余白を追加することもできます。

```
\let\myFoot\footnote
\renewcommand{\footnote}[1]{\myFoot{#1\vspace{1.5mm}}}
```

二段組文書では、脚註は各段底部に表示されます (図 4.1)。図 4.2 のように、脚註を右段底部にのみ表示するようにするには、文書プリアンブルに

```
\usepackage{ftnright}
```

というコマンドを書いて、`LaTeX` パッケージ `ftnright` を読み込みます。

Sei nun S unser normiertes Ausgangssignal und P die Phasenverteilungsfunktion, so ergibt sich die Beziehung	Das Spektrum wird Fouriertransformiert. Die Fouriertransformation wird verwendet, um die überlagerten Signale (Netzwerk, Lösungsmittel) zu trennen. Nachdem wir die Phasenverschiebung bestimmen konnten, interessiert uns nun das Aussehen des Ausgangssignals. Im Experiment haben wir es mit sehr vielen Teilchen zu tun, so dass
$S(t) = S_0(t) \int_{-\infty}^{\infty} P(\phi, t) e^{i\phi} d\phi \quad (2)$	
wobei S_0 das Signal ohne Gradient ist und die Normierungsbedingung	
¹ Fourier transformation	³ Fourier transformation
² Phase distribution function	⁴ Phase distribution function
1	

図 4.1.: 二段組文書の標準的な脚註配置

科学的文献においては、図 4.3 のように、節の末尾に別段落を設けて、脚註をまとめて出力することも普通に行われます。これは「後註」と呼ばれるものです。文書中で

^{4.5}これは「章番号.脚註番号」スキームで番号を振られた脚註です。

man über alle Phasen integrieren muss. Sei nun S unser normiertes Ausgangssignal und P die Phasenverteilungsfunktion, so ergibt sich die Beziehung

$$S(t) = S_0(t) \int_{-\infty}^{\infty} P(\phi, t) e^{i\phi} d\phi \quad (2)$$

wobei S_0 das Signal ohne Gradient ist und die Normierungsbedingung $\int_{-\infty}^{\infty} P(\phi, t) d\phi = 1$ gilt. Nun dürfen

Das Spektrum wird fouriertransformiert. Die Fouriertransformation wird verwendet, um die überlagerten Signale (Netzwerk, Lösungsmittel) zu trennen. Nachdem wir die Phasenverschiebung bestimmen konnten, interessiert uns nun das Aussehen des

1. Fourier transformation
2. Phase distribution function
3. Fourier transformation
4. Phase distribution function

1

図 4.2.: L^AT_EX パッケージ `ftnright` を使用したときの二段組文書の脚註配置

man über alle Phasen integrieren muss. Sei nun S unser normiertes Ausgangssignal und P die Phasenverteilungsfunktion, so ergibt sich die Beziehung

$$S(t) = S_0(t) \int_{-\infty}^{\infty} P(\phi, t) e^{i\phi} d\phi \quad (2)$$

wobei S_0 das Signal ohne Gradient ist und die Normierungsbedingung $\int_{-\infty}^{\infty} P(\phi, t) d\phi = 1$ gilt. Nun dürfen

Notes

- ¹Fourier transformation
- ²Phase distribution function
- ³Fourier transformation
- ⁴Phase distribution function

図 4.3.: 後註 – 注を節ないし章末に別段落として印字します。

脚註の代わりに後註を使用するには、**文書**▷**設定**▷**モジュール**メニューで脚註から後註へを読み込んでください。まとめた注を出力するには、節ないし章末に T_EX コードで

`\theendnotes`

というコマンドを挿入してください。

後註の段落見出しは、文書言語に自動的に翻訳されることはありませんので、手動で翻訳しなくてはなりません。プリアンブルで

`\renewcommand{\notesname}{Anmerkungen}`

というコマンドを使えば、既定の英語名「Notes」を独語訳「Anmerkungen」に変えることができます。

第 4.2.1 節で述べた脚註の付番と同様に、後註の付番も変更することができます。 `\the-footnote` コマンドの代わりに `\theendnote` を用いてください。後註番号をリセット

4. 註釈

するには、第 4.2.1 節で述べたコマンド `\@addtoreset` の、パラメータ `footnote` を `endnote` に変えて、使用してください。

後註の印だけを作成するには、第 4.2 節で説明した `\footnotemark` コマンドの代わりに、`\endnotemark[番号]` コマンドを使用してください。

脚註はページ余白に置くこともでき、脚註文の揃えも変更することができます。詳細は、`LATEX` パッケージ `footmisc`[9] を参照してください。

脚註の整形にまつわるその他多くの話題が、`LATEX` 関連書籍 [1, 2, 3] に載っています。`KOMA-Script` 文書クラスを使用している場合には、すべての脚註設定を、クラス固有のコマンドで調節することができます ([12] 参照)。

4.3. 傍註

`LATEX` 内において、傍註註釈は、脚註のような外見と挙動を示します。これは **挿入**▷**傍註** メニューかツールバーボタンで挿入することができます。すると、紫字で「傍註」と書かれた灰色のボックスが現れて、そこに傍註文を入れることができます。

これは傍註文
です。

横にあるのが傍註です。

片面文書においては、傍註は右側に表示され、両面文書においては外側余白—偶数ページでは左側、奇数ページでは右側—に表示されます。傍註文は、外側余白の反対側に—注が左余白に現れるときには右揃えに—寄せられます。傍註文の一行目は、傍註が挿入された文書箇所と同じ行に来るように配置されます。

傍註を内側余白に置くには、傍註の前に `TEX` コードで

```
\reversemarginpar
```

というコマンドを書き加えてください。この新しい配置は、以後のすべての傍註に対して効力があります。

[註] 内側余白には十分な空白がないことがよくあり、出力で傍註が正しく表示されないことがあります。

これは内側
白の傍註
です。

既定の配置に戻すには、`TEX` コードで

```
\normalmarginpar
```

というコマンドを書き加えてください。**[註]** `\reversemarginpar` コマンドが既に挿入されている段落内では、このコマンドは無視されます。

第 2.8.1 節で述べられている事例と同様に、長い単語が傍註文の最初の単語として来るとハイフネートが行われません。これを回避するには、その単語の前に `0pt` 水平空白を挿入してください。

[註] 通常、傍註は表やフロート、脚註の中では使用することができません。

AVeryLong-
MarginPar-
Word (とっ
ても長い傍註
段落中の単
語) がハイフ
ネートされ
る例.

この制約は、 \LaTeX パッケージ `marginnote` を使用することで回避することができます。以下の二行を使用中の文書プリアンブルに書き加えると、 \LaTeX が傍註に使うコマンドが再定義されて、`marginnote` パッケージが提供するコマンドを使うようになります。

```
\usepackage{marginnote}
\let\marginpar\marginnote
```

`marginnote` には他にも便利な機能があるので、この方法が本文書でも使われています。というのは、`marginnote` を使うと、傍註文に垂直オフセットを設定することができるからです。これは、あまりにもたくさんの傍註が非常に近いところに集まっていたり、もっとよいページレイアウトが望まれるときなどに、よく必要とされます。オフセットは、 \LaTeX 中の傍註の後ろに \TeX コードで直接

[オフセット]

という形で書くことで設定することができます。ここで「オフセット」には、表 A.1 に挙げてある単位の一つを使用します。負の値を与えると註釈は上にずれ、正の値だと下にずれます。たとえば、この本文行の横にある傍註は、 \TeX コードコマンド「`[-1.5cm]`」を使用して、1.5 cm 上にずらしてあります。

この傍註は、本来の位置よりも 1.5 cm 上にずらしてあります。

`marginnote` を使うと、傍註の揃えを変更することもできます。たとえば、

```
\renewcommand*{\raggedleftmarginnote}{\centering}
\renewcommand*{\raggedrightmarginnote}{\centering}
```

というコマンドを使うと、中央揃えになります。`\raggedleftmarginnote` とは、左側に表示されている傍註のことを指します。既定の設定は、

この傍註の文章は中央揃えになっています。

```
\renewcommand*{\raggedleftmarginnote}{\raggedleft}
\renewcommand*{\raggedrightmarginnote}{\raggedright}
```

です。

`marginnote` の他の機能については、[16] の取扱説明書をご参照ください。

傍註のレイアウトは、傍註の定義を変更することによって変えることができます。たとえば、すべての傍註に、下線を引いたサンセリフ体の太字で「**注意!**」というヘッダ文をつけるには、以下を使用している文書プリアンブルに書き加えてください。

```
\let\myMarginpar\marginpar
\renewcommand{\marginpar}[1]{\myMarginpar{%
\hspace{0pt}\textsf{\textbf{\underbar{注意!}}}}%
\vspace{1.5mm}\#1}}
```

注意!

これは、定義したヘッダをつけた傍註です。

5. ボックス

5.1. はじめに

ボックスは、一塊の文章を整形するのに使用します。ボックスは、多言語文書を書くのに使用したり（第 5.4 節参照）、文章を囲ったり（第 5.2.3 節参照）、単語がハイフネートされるのを予防したり（第 5.6.1 節参照）、文章を揃えたり（第 5.6.2 節参照）、あるいは文章の背景色を設定したり（第 5.7 節参照）するのに使用します。

ボックスは、挿入▷ボックスメニューかツールバーボタンを使って挿入することができます。すると、**ボックス (ミニページ)** のようにボックス (minipage 環境) というラベルのついた灰色の箱形が挿入されます。ボックス型は、この箱形を右クリックすると出てくるメニューで指定することができます。設定メニューを選択すると現れるボックスダイアログでは、parbox コマンドと minipage 環境の二つの**内部ボックス型**を選択することができます。minipage 型は、新規ボックスの既定値となっており、第 5.4 節で説明します。parbox 型については第 5.5 節、makebox 型については第 5.6.1 節で述べます。

ボックスは付番されないので、フロートや脚註のように参照することはできません。

【註】 ボックスは、**箇条書き (記号) 環境**や**箇条書き (記述) 環境**の単独要素とすべきではありません。

【註】 理由は不明ですが、**枠付きボックス**を使用している場合には、**内部ボックス型**はなしにしか設定することができません。**内部ボックス型**や**枠のないボックス**については、第 5.6.1 節で説明します。

5.2. ボックスダイアログ

5.2.1. 寸法

ボックスダイアログでは、幅と高さの各フィールドでボックスのジオメトリを調整することができます。ジオメトリに使用することのできる単位は、表 A.1 に説明されています。高さフィールドには、加えて以下の寸法を使用することができます。

階層 これは平文の「高さ」です。ボックス中に複数行がある場合には、全体としての階層は無視されます。

5. ボックス

高さを1
階層に設
定した
ボックス

高さ これはボックス内部にある文章の高さです。この寸法でたとえば2の値を与え

ると、ボックスの高さを文章の2倍の高さに設定します。

高さを2高さに
設定したボッ
クス

全高 これは「高さ」+「階層」です。

高さを1全高に
設定したボッ
クス

幅 これはボックスの幅を高さに設定します。

高さを1
幅に設定
したボッ
クス

5.2.2. 文字揃え

内部ボックスを選んだ場合には、ボックスの垂直揃えを以下のいずれかに設定できます。

上 これは例示のための本文行です。

このボッ
クスは上
揃えです。

これは例示のための本文行です。

中央 これは例示のための本文行です。

このボッ
クスは中
央揃え
です。

これは例示のための本文行です。

下 これは例示のための本文行です。

このボッ
クスは下
揃えです。

これは例示のための本文行です。

[註] 同一行に二つのボックスがあり、一方だけに影付きなどが設定されている場合には、ボックスの垂直文字揃えが出力中で無効となる場合があります。

ボックスを独立した段落においた場合には、ボックスの水平文字揃えは、L_AT_EXの段落ダイアログから設定できます。

内部ボックスを設定したときには、ボックスの内容を以下のように垂直揃えを行うことができます。

上 これは例示のための本文行です。 このボック
スの文
章は上揃
えです。 これは例示のための本文行です。

中央 これは例示のための本文行です。 このボッ
クスの文
章は中央
揃えです。 これは例示のための本文行です。

下 これは例示のための本文行です。 このボッ
クスの文
章は下揃
えです。 これは例示のための本文行です。

広げる これは例示のための本文行です。 このボック
スの
文章は広げて
あります。 これは例示のための本文行
です。

ボックスの内容を広げるには、ボックスに複数の段落がなくはなりません。上述の例では、文章の各行を独立の段落に設定してあります。

内部ボックスを設定しているときに、ボックスの内容を水平方向に揃えるには、 L_YX の段落ダイアログを用います。

このボックスの

文章は広げて

あります。

内部ボックスを設定していないときには、ボックスの内容は、ボックスダイアログの中で水平揃えの設定をすることができます。

このボックスの文章は水平方向に広げてあります。

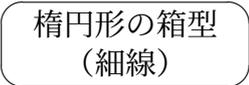
5. ボックス

5.2.3. 装飾

ボックス型は、ボックスダイアログ中の装飾ドロップダウン式一覧から指定することができます。以下の型を指定することができます。

枠なし 枠を描きません。

簡素な長方形の枠 これはボックスの周りに長方形の枠を描きます。枠線の太さを指定することができます。  長方形の箱型

楕円形の枠（細線） これはボックスの周りに楕円形の枠を描きます。枠線の太さは 0.4 pt です。  楕円形の箱型
(細線)

楕円形の枠（太線） これはボックスの周りに楕円形の枠を描きます。枠線の太さは 0.8 pt です。  楕円形の箱型
(太線)

影付き これはボックスの周りに影付きの長方形の枠を描きます。枠線の太さと影を指定することができます。  影付きの箱型

影の背景 これは背景色が赤のボックスを描きます。色付きボックスとは違って¹、これは列幅全体を使用し、ボックスは独立した段落に設定されます。

影付き背景の箱型

二重線の長方形の枠 これはボックスの周りに二重線の長方形の枠を描きます。内側の枠の太さは、線幅の 0.75 倍であり、外側の枠の太さは、線幅の 1.5 倍です。枠同士の間隔は、線幅の 1.5 倍 + 0.5 pt です。  二重の長方形の
箱型

LyX のボックスラベルには、使用中の枠の型が表示されます。すべての型を使用可能にするには、`LaTeX` パッケージ `fancybox` が導入済みでなくてはなりません。

簡素な長方形の枠の装飾を使って内部ボックスを使用しないとき、ボックス内で改頁を許可することができます。その場合、他の枠付きボックスと違って、この枠は、つねに段幅の全幅を使用し、ボックスは独立の段落として設定されることに注意してください。

改行可能な箱型

¹第 5.7 節を参照のこと。

5.3. ボックスの詳細な設定

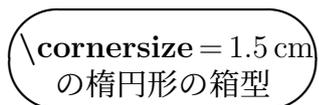
楕円形ボックスの丸い角の直径は、コマンド`\cornersize`で設定することができます。

```
\cornersize*{1cm}
```

というコマンドは、直径を 1 cm に設定します。

```
\cornersize{数値}
```

というコマンドは、直径を **数値 × (ボックスの幅と高さの小さい方)** に設定します。既定値は`\cornersize{0.5}`です。



```
\cornersize = 1.5 cm  
の楕円形の箱型
```

影付き背景ボックスの赤色の既定背景色は、`\definecolor{shadebox}` コマンドで局所的に変更することもできますし、**ツール▷設定▷色▷影付きボックス** メニューからグローバルに変更することもできます。`\definecolor` コマンドの用法は第 2.11 節²で説明されています。

たとえば、下記の影付き背景ボックスの外見は

```
\definecolor{shadecolor}{cmyk}{0.5,0,1,0.5}
```

という TeX コードコマンドを使って設定してあります。

これは、濃緑色の背景の影付き背景ボックスに入れた黄色の文章です。

手動で変更した寸法と色は、それを変更したコマンド以降のすべてのボックスに対して有効です。

5.4. ミニページ

ミニページは、L^AT_EX によってページの中のページとして取り扱われるので、たとえばそれ自身で脚註を持つことができます。

ミニページは、複数の言語を含む文書を執筆する際に便利です。

下掲は、二つの例示用ミニページを横に並べたものです。これらの幅は 45 列幅%に設定され、挿入▷整形▷水平方向の空白メニューで挿入した水平フィルで隔てられています。

²`\definecolor` は、L^AT_EX パッケージの `color` をプリアンブルに読み込む必要があります。第 5.7 節をご参照ください。..

5. ボックス

Dies ist ein deutscher Text. Dies ist ein deutscher Text.^a. Dies ist ein deutscher Text. Dies ist ein deutscher Text.

^aDies ist eine deutsche Fußnote.

This is an English Text. This is an English Text.^a This is an English Text.

^aThis is an English footnote.

ミニページ内では、文書全体の段落設定は無視されます。つまり、たとえば文書設定で中スキップを設定していたとしても、ミニページ内の段落間には余白が置かれません。

ミニページは、文章部に背景色を与えるのに使用することもできます。第5.7節をご参照ください。

[註] ミニページ内にフロートや傍註を使用することはできませんが、ミニページを表・フロート・他のボックス内で使用することはできます。

LyXは、**可変幅ミニページ**モジュールを経由して、内容の幅によって自動的に幅が調整されるミニページを一部サポートしています。LyX例示ファイル *varwidth-floats-side-by-side* では、2つのフロートを並べて配置するのに可変幅ミニページを使用する例を取り扱っています。

5.5. 段落ボックス (parbox)

段落ボックス (parbox) はミニページに非常によく似ていますが、脚註を付けることができない点が違います。

これは段落ボックス内の文章です。これは段落ボックス内の文章です。右の脚註は表示されません。

3

5.6. 単語や文字のボックス

5.6.1. ハイフネーションの回避

単語や文章がハイフネートされるのを防ぐには、`makebox` 型ボックスを使うことができます。

以下が例示用の文章です。

This line is an example to show how you can prevent the hyphenation of “veryvery-longword”.

単語「veryverylongword」がハイフネートされるのを防ぐには、単語をボックスの中に入れて、**内部ボックス**を `makebox` コマンドに設定し、幅は指定しません。

以下がその結果です。

This line is an example to show how you can prevent the hyphenation of “veryverylongword”.

別の方法としては、以下のように、単語の前に直接 `TeX` コードで「`\-`」コマンドを設定する方法もあります。

This line is an example to show how you can prevent the hyphenation of “veryverylongword”.

もちろん、こうすることでこの単語は右余白に突き出てしまいます。これを回避するには、以下のように **挿入▷整形▷整形なし改行**メニュー（ショートカット）から改行を単語の前に入れてください。

This line is an example to show how you can prevent the hyphenation of “veryverylongword”.

5.6.2. 垂直揃え

`\raisebox` コマンドを使うと、単語や文字や他のボックスを、周囲の文章に対して垂直方向に整列させることができます。`\raisebox` は以下の書式で使います。

`\raisebox{持ち上げ幅}[高さ][深さ]{ボックスの内容}`

ここで持ち上げ幅は、正の値ならばボックスを持ち上げ、負の値ならばボックスを引き下げます。

たとえば、「preventing」という単語を、そのもっとも「深い」文字「p」の底部がベースラインに来るように揃えるには、単語の前に `TeX` コードで

`\raisebox{\depth}{`

というコマンドを挿入してください。単語の後には、閉じ括弧「`}`」を `TeX` コードで挿入してください。

以下がその出力です。

This is a text line with the word “preventing” as raised word.

5. ボックス

行中の文字を上げ下げすると、以下のように行間が広がります。

This is a text line with the word “preventing” as lowered word.
“testing”

This is a text line with the word as raised word.

何らかの理由でこれを抑制したい場合には、ボックスの高さを零に設定してください。たとえば、

```
\raisebox{-\depth}[0pt]{
```

とすると、以下のようになります。

This is a text line with the word “preventing” as lowered word.
“testing”

This is a text line with the word as raised word.

5.7. 色付きボックス

ボックスの背景に色を付けるには、ボックスダイアログで好みの色を選択するだけです。これは、橙色の背景を持つボックスの例です：**橙色背景のボックス**

装飾を簡素な長方形の枠にした場合には、枠の色を設定できます。この場合、背景色も指定しなくてはなりません (LyX は白を既定色として設定します)。

例：**ティール色の枠と黄色の背景を持つボックス**

もちろん、色付きボックスの中に色付きの文を入れることもできます：

これは着色した色付きボックス中の色付き文章です。

事前に設定された色以外の色が欲しいときは、第 2.11 節に述べられているように、独自色を定義することができます。独自色を使う場合には、以下のように、TeX コードを使ってボックスを組版しなくてはなりません。

枠なしの色付きボックスは、`\colorbox` コマンドで作ります。これは、以下のような書式で使われます。

```
\colorbox{色}{ボックスの内容}
```

ボックスの内容には、ボックスが入ることもできますし、色付きボックスを他のボックスに入れることもできます。たとえば、単語の背景を濃緑にするには、単語の前に TeX コードで

```
\colorbox{darkgreen}{
```

というコマンドを入れてください。単語の後には、TeX コードで閉じ括弧「}」を入れてください。

以下がその結果です。

この行では、「**注意!**」という単語が濃緑の背景になっています。

枠付き色付きのボックスは、下記のように `\fcolorbox` コマンドを使って作ります。

```
\fcolorbox{枠の色}{ボックスの色}{ボックスの内容}
```

5.8. ボックスの伸縮回転

本節に述べられているボックスを使うには、画像ボックスモジュールを文書設定で読み込んでおく必要があります。

[註] DVIビューアによっては、回転伸縮したものの表示ができないものがあります。

[註] フロートは回転伸縮したボックスの中に入れることはできません。

5.8.1. ボックスの回転

ものを回転するならば、**回転ボックス**に入れることができます。このボックスは、**挿入▷拡張差込枠▷回転ボックス**で挿入することができます。

回転の原点は、原点差込枠 (**挿入▷原点**メニュー) に **origin=位置** の形で指定することができます。位置として指定可能なのは、*c* (中央揃え)・*l* (左揃え)・*r* (右揃え)・*b* (下揃え)・*t* (上揃え) と、これら五つの適切な組み合わせです。たとえば、*lt* と指定した場合には、回転の原点をボックスの左上隅にすることです。回転原点が指定されなかった場合には、位置として *l* が使用されます。回転角は、角度を指定する数字で負の値もとることができます。回転の方向は、反時計回りです。

下の例では、**原点差込枠**に **origin=c**、**回転差込枠**に **60** と入れています。

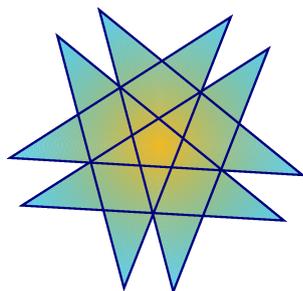
これは、 文章を含む行です。

ボックスの中身には、以下のように、もう一つボックスを入れたり行内数式を入れたりすることもできます。

これは、 数式を含む行です。

あるいは、以下のように図や表を入れることもできます。

5. ボックス



これは、

回転した画像と表を含む行です。

e	b
r	M

[註] 角度差込枠が空であったり、数字以外の文字が入っていたりした場合、文書を閲覧したり書き出したりする際に、 \LaTeX エラーが発生します。

5.8.2. ボックスの伸縮

ものを伸縮するならば、**伸縮ボックス**に入れることができます。このボックスは、**挿入**▷**拡張差込枠**▷**伸縮ボックス**で挿入することができます。

伸縮ボックスには、必須差込枠「**水平因子**」と、非必須差込枠「**垂直因子**」(**挿入**▷**垂直因子**メニューで挿入可)がありますので、そこに水平因子と垂直因子を入力してください。**垂直因子**を使わない場合には、水平因子が垂直因子としても使われます。

たとえば、文書の本文の大きさに比べて、文を2倍の大きさにしたい場合、**水平因子**差込枠に2と入力してください。例：**こんにちは**

水平因子を2、**垂直因子**を1にすると、このようにゆがみます：**こんにちは**

伸縮因子を負に設定した場合、ボックスの内容は鏡像になります。鏡文字を作りたい場合には、**水平因子**を-1、**垂直因子**を1にします：**おきこひこ**

水平因子を1、**垂直因子**を-1にすると、ベースラインでひっくり返した形になります：**おきこひこ**

反転ボックス(**挿入**▷**拡張差込枠**▷**反転ボックス**メニュー)は、**伸縮ボックス**で**水平因子**を-1、**垂直因子**を1にしたものと同じです。

[註] **水平因子**差込枠が空であったり、数字以外の文字が入っていたりした場合、文書を閲覧したり書き出したりする際に、 \LaTeX エラーが発生します。

中身を決まった幅と高さに伸縮したい場合には、**寸法変更ボックス**(**挿入**▷**拡張差込枠**▷**寸法変更ボックス**メニュー)を使うことができます。**寸法変更ボックス**には、必須差込枠「**幅**」と、非必須差込枠「**高さ**」(**挿入**▷**高さ**メニューで挿入可)がありますので、そこに幅と高さを単位付きの値で入力してください。A.1に使用できる単位が列挙してあります。2つの差込枠のいずれかに感嘆符!があると、ボックス内容のアスペクト比が保たれるように寸法が設定されます。**高さ**差込枠を使わない場合には、高さには内部的に感嘆符が使用されます。

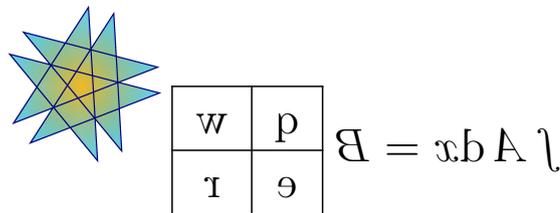
2cm を幅に, 1cm を高さを使用するとこうなります: こんにちは
 2cm を幅に, !を高さを使用すると(あるいは高さなし)こうなります: こんにちは

[註] 幅差込枠が空であったり, 無効な単位が入っていたりした場合, 文書を閲覧したり書き出したりする際に, L^AT_EX エラーが発生します.

ボックスは好きな順序で組み合わせることができます. 回転ボックスの中に寸法変更ボックスが入り, さらにその中に反転ボックスが入っている例です:

こんにちは

という出力になります. ボックスの内容には, 以下のように, 画像や表や行内数式を入れてもかまいません.



文書で大域的に数式様式 `fleqn` を使用している場合⁴には, 別行建て数式も伸縮させることができます.

⁴つまり, 文書クラスオプションに「`fleqn`」を書き加えた場合.

6. 本文回り込みオブジェクト

6.1. 折返しフロート

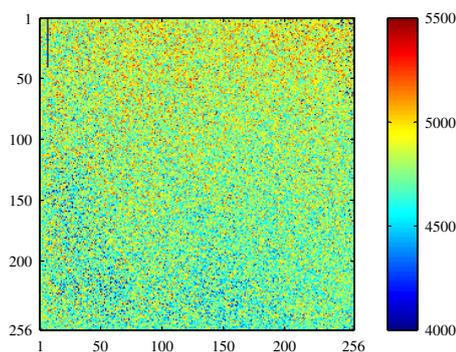


図 6.1.: これは折返し図フロートです。

このフロート型は、本文が図の周りに折返して入ることによって、図が段幅の一部しか占拠しないようにしたいときに用います。折返しフロートは、 \LaTeX パッケージの `wrapfig` を導入済みであれば、挿入▷フロート▷折返し図・折返し表メニューで挿入することができます¹。フロートの設定は、フロートボックスを右クリックすることで変更することができます。必須の設定項目は、フロートの配置と幅です。非必須項目として、ぶら下げと行幅があります。前者は、フロートがどれくらい段落ないしページ余白に入り込むかを指定するものであり、後者は、フロートが必要とするおおよその本文行数を指定するものです。行幅は、推測するのが難しいことがよくあるので、フロートの配置に問題が生じたときのみ使用するようにした方が良いでしょう。

さらに、フロートを段落内でフロートするようになるか、周囲の段落までフロートすることを \LaTeX に許可するかを指定することができます。第 6.1 図は、40 列幅%の幅と 1 cm のぶら下げを持ち、左揃えの折返しフロートを例示してあります。

【註】 折返しフロートは壊れやすいことがあります！例えば、図をページの下すぎる位置に置いてしまうと、ごちゃごちゃになって、フロートが出力に現れなかったり、他の本文の上に上書きされて出力されたりします。

原則として、

- 折返しフロートは、改頁に掛かってしまう段落に置くべきではありません。つまり、折返しフロートは、文書がほぼ完成する段階になって、どこに改頁が現れるかを予測できるようになってから、正確な位置に挿入しなくてはなりません。
- 折返しフロートは、入れ込む段落の前に単独の段落を設けて挿入するか、その段落自体の中に挿入しなくてはなりません。
- 折返しフロートを二つの段落に連続して入れると問題を引き起こす可能性があるため、二つの折返しフロートのあいだには本文のみの段落が入るようにしてください。

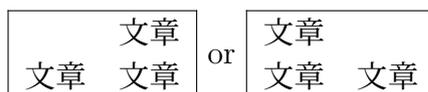
¹ \LaTeX パッケージの導入方法は、取扱説明書の『 \LaTeX の設定』に説明されています。

6. 本文回り込みオブジェクト

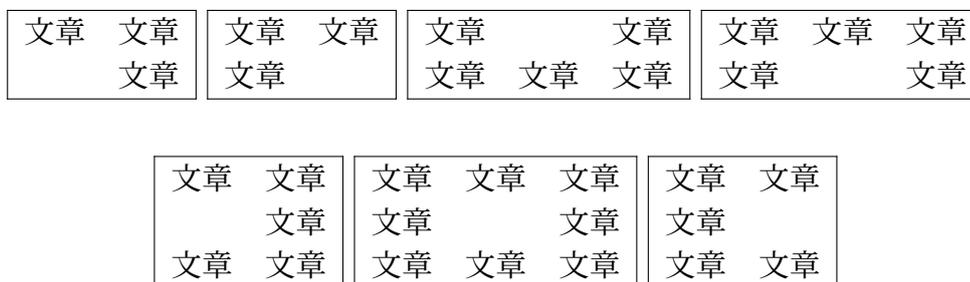
- 折返しフロートは、節見出しや表中に入れることは認められていません。
- 連番や参照は、第3章で述べられているように動作します。

6.2. 固定回り込みオブジェクト

第6.1節の折返しフロートは、段落中で、以下のような配置がされます。

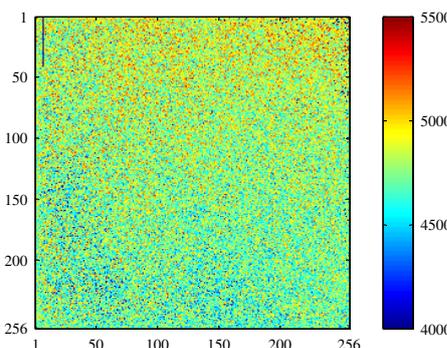


以下に示すいずれかのように、



オブジェクトを挿入した場所に正確に配置したい時には、`picinpar`LaTeX パッケージを使うことができます。下記はその用例です。

It was in the spring of the year 1894 that all London was interested, and the fashionable world dismayed, by the murder of the Honourable Ronald Adair under most unusual and inexplicable circumstances. The public has already learned those particulars of the crime which² came out in the police investigation, but a good deal was suppressed upon that occasion, since the case for the prosecuting was so overwhelmingly strong that it was not necessary to bring forward all the facts. Only now, at the end of nearly ten years, am I allowed to supply those missing links of which make up the whole of that remarkable chain.



固定回り込みオブジェクトは、オブジェクトの周りに本文を流しこむのに用いられ、段落中のオブジェクトの位置は、オブジェクトの上の行数を指定することで決定します。フロートとは違って、連番のキャプションをつけることはできません。

固定オブジェクトは、段落冒頭に TeX コードで

²固定オブジェクトのある段落の脚註

```
\begin{window}[2,c,{
```

というコマンドを書くことによって挿入されます。角括弧中の最初のパラメータ(ここでは「2」)は、オブジェクト上に来る行数を指定し、二番目のパラメータ(ここでは「c」)はオブジェクトの位置を示します。位置としては、「l」「c」「r」が可能です(それぞれ左・中央・右)。このコマンドの後にオブジェクトを挿入します。この例では、段幅の40%の画像が挿入されています。オブジェクトは、表やボックス、数式でも構いません。オブジェクトの後には、

```
}, 説明文]
{}
```

というコマンドが挿入されています。2行目に「{ }」が入っていることに注意してください。説明文は非必須です。オブジェクトに回り込む文章は、オブジェクトと同じ段落になくてもなりませんので、{ }の直後から書き始めてください。段落末には、

```
\end{window}
```

というコマンドを挿入してください。固定オブジェクトのある段落では、通常のように脚註を挿入することはできません。第4.2節にあるように、代わりに、段落内で`\footnotemark` コマンドを使用し、段落外に`\footnotetext` コマンドを用いてください。

6.3. 頭文字

頭文字を用いるには、 \LaTeX パッケージが導入済みでなくてはならず、文書中で**頭文字**モジュールを使用する必要があります(文書▷設定▷モジュールメニュー)。頭文字を挿入するには、**頭文字**様式を使用してください。この様式には、1つの必須引数と2つの非必須引数があります。引数は、**挿入**メニューから挿入できます。必須引数**頭文字**には、頭文字にする文字を入れ、引数**頭文字の残り**には、頭文字の単語若しくは文字列の残りが入ります。引数**オプション**には、頭文字を修正するパラメータが入ります。使うことができるパラメータについての詳細は、**lettrine**の取扱説明書[13]をご覧ください。この様式の段落そのものに、頭文字に回り込む文章も入れます。通常、この様式では、最初の文字を頭文字とする単語全体が、スモールキャップ体(SMALL CAPS)で印字されます。

下記がその例です。

1. 非必須引数なし。頭文字は、数式フォント \mathfrak{I} の「I」。

\mathfrak{I} T was in the spring of the year 1894 that all London was interested, and the fashionable world dismayed, by the murder of the Honourable Ronald Adair under most unusual and inexplicable circumstances. The public has already learned those particulars of the crime which came out in the police investigation, but a good deal was suppressed upon that occasion, since the case for the prosecution was so overwhelmingly strong that it was not necessary to bring forward all the facts.

6. 本文回り込みオブジェクト

2. 頭文字に回り込む行は1行のみ。頭文字は標準フォントで2文字。スモールキャップ体で印字する文字はなし。

It was in the spring of the year 1894 that all London was interested, and the fashionable world dismayed, by the murder of the Honourable Ronald Adair under most unusual and inexplicable circumstances. The public has already learned those particulars of the crime which came out in the police investigation, but a good deal was suppressed upon that occasion, since the case for the prosecution was so overwhelmingly strong that it was not necessary to bring forward all the facts.

3. ぶら下がり頭文字は、数式フォント *CALIGRAPHIC* で3行にかかり、マゼンタ色。「t was in the spring」という文章をスモールキャップ体。頭文字に色をつけるには、頭文字の前に

```
\renewcommand{\LettrineFontHook}{\color{色}}
```

というコマンドを挿入しなくてはなりません。ここで、色は「red」・「blue」・「green」・「yellow」・「cyan」・「magenta」・「black」・「white」のいずれかです。また、文書プリアンブルに

```
\@ifundefined{textcolor}
```

```
{\usepackage{color}}{}
```

というコマンドを書きこんでおかなくてはなりません。

IT WAS IN THE SPRING of the year 1894 that all London was interested, and the fashionable world dismayed, by the murder of the Honourable Ronald Adair under most unusual and inexplicable circumstances. The public has already learned those particulars of the crime which came out in the police investigation, but a good deal was suppressed upon that occasion, since the case for the prosecution was so overwhelmingly strong that it was not necessary to bring forward all the facts.

7. 外部文書構成要素

挿入▷ファイルメニューを使うと、使用中の文書に外部素材を挿入することができます。挿入できるのは、

LyX 文書 別の LyX 文書。その内容は直接使用中の文書に挿入されます。

平文 テキスト文書。そのテキスト行各々が、使用中の文書に独立した段落として挿入されます。

平文 (行を連結して) テキスト文書。各テキスト行がその形のまま挿入されます。空の行があると、使用中の文書に新規段落を生成します。

外部素材 様々な書式のファイル。

子文書 LyX あるいは L^AT_EX 文書。

7.1. 外部素材

外部素材機能は、文書の出力書式が読むことのできる形式に前もって変換しなくても、LyXが必要な変換を行うことで、使用中の文書にファイルを挿入することができるようにするものです。これは、画像が様々な画像形式のまま LyX 文書に挿入することができるのと同様の働きをします。LyX 設定の **操作性▷表示** で自動プレビューを有効にすると、外部素材の Dia 形式と Xfig 形式は、直接 LyX 中に表示されるようになります。

外部素材は、**挿入▷ファイル▷外部素材** メニューで開くことのできる **外部素材** ダイアログから挿入することができます。目下、以下のファイル形式 (**ひな型**) が使用できます。

チェス棋譜 このひな型は、**XBoard** プログラムで作成されたチェス棋譜をサポートします。

Dia ダイアグラム このひな型は、**Dia** プログラムで作成したダイアグラムをサポートします。

Inkscape  このひな型は、スケーラブル・ベクトル・グラフィックス (SVG) から派生した特定のスケーラブル・イメージをサポートしており、画像に含まれるテキストを L^AT_EX で (したがって現文書のフォントで) 組んだバージョンに置き換えることができます。これは、与えられた SVG ファイルから特殊な PDF や EPS および L^AT_EX ファイルを生成することができる、プログラム **Inkscape** の機能を使用しています (INKSCAPE 自身では、描画を PDF や EPS で保存する

7. 外部文書構成要素

際、オプション PDF のテキストを除外して LaTeX ファイルを生成を用いてこの機能を使用することができます)。このようなファイルの利点は、全文書を通じて統一されたテキスト様式を持たせることができることです。

この外部ひな型は、SVG ファイルを読み込み、変換プログラムを使用して (Inkscape 経由で) PDF/EPS ファイルと L^AT_EX ファイルを同時並行で生成しますので、これらのファイルを書き出す必要がありません。そうではなく、*.pdf と *.pdf_tex もしくは *.eps と *.eps_tex を書き出して使用したい場合には、このひな型を使用せず、子文書ダイアログ (第 7.2 節参照) から *.pdf_tex または *.eps_tex を入力してください。

SVG のテキスト中では、\Large や \small のような T_EX コードコマンドを使用することができます。フォントの大きさは、L_YX 文書の設定から採用されます。詳しい情報と用例については [22] を参照してください。

Lilypond 組版楽譜 このひな型は、LilyPond プログラムで作成した楽譜の組版に使用されます。L_YX における LilyPond 拡張サポートについては、L_YX 用例ファイル *lilypond.lyx* をご覧ください。

PDF ページ このひな型を使えば、PDF 文書を使用中の文書に挿入することができます。PDF の一部頁ないし全頁を挿入するには、ダイアログ中のひな型の解説に沿って、L^AT_EX と L_YX オプションタブ中のオプションフィールドに *pages* オプションを指定してください。このオプションの引数は、コンマ区切りのリストで、ページ番号 (オプションを `pages={3,5,6,8}` の形で入れる) やページ範囲 (`pages={4-9}`)、あるいはこれらの組み合わせで指定できます。空のページを入れるには、`{}` を指定してください。例えば、`pages={3, {}, 8-11, 15}` は、3 ページ目と空のページ、8・9・10・11・15 ページ目を挿入します。ページ範囲は、**開始頁-終了頁** の文法で指定します。これは、**開始頁** と **終了頁** の間のすべてのページを選択します。**開始頁** を省略すると、1 ページ目を指定したのと同じ事になり、**終了頁** を省略すると、文書の最終ページを指定したのと同じ事になります。最終ページは、`last` というキーワードを使って指定することもできます (これはページ範囲指定のみで有効です)。したがって、`pages=-` は文書の全ページを挿入することになり、`pages=last-1` は全ページを逆順で挿入します。`pages` オプションを指定しないと、PDF の最初のページのみが挿入されます。`noautoscale` オプションは、各ページを元々の寸法で挿入します。もしこのオプションが指定されていないならば、各ページは文書のページ幅に合うように大きさが変更されます。寸法と回転タブでは、取り込むページの縦横の大きさと回転角を指定することができます。100 縮尺%以外の幅が指定されると、`noautoscale` オプションは無視されます。詳しい情報と取り得るオプション値については、L^AT_EX パッケージ `pdfpages` の取扱説明書 [17] をご参照ください。

このひな型を Beamer プレゼンテーションで使用するには、まず **フレーム終了** 環境を挿入して、続く段落で T_EX コードボックスを挿入し、`\setbeamertemplate{background canvas}{}` というコマンドを入力してください。すると、PDF ページのひな型を含んだ段落を好きなだけ入れることができるようになります。取り込んだ PDF ページの後で、新しいフレームを開始するか、新しい節を開始することによって、Beamer プレゼンテーションに復帰することができます。PDF ページの前に空白ページができてしまうのを防ぐために、ひな型を

含む段落には、既定配置を用いるようにしてください。
 下記は、-5度回転させたPDFを取り込んだものです。

 This is an abstract. As you can see, it is
 printed in a smaller font size than the other
 paragraph types.

Also several paragraphs are possible in the
 abstract.

This is a “Standard” paragraph to visualize the differ-
 ences in the font size.

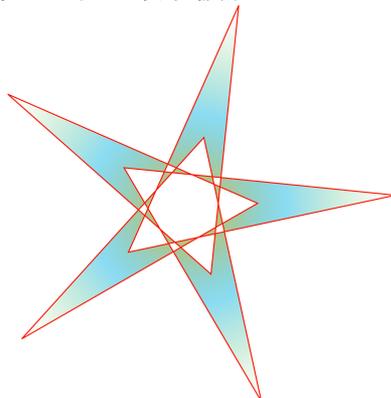
【訳註】 L^AT_EX パッケージ `pdfpages` は、`pdflatex` に基づいており、L^AT_EX では組版ができない (L^AT_EX パッケージ `pdfpages` 取扱説明書 [17] 参照) ため、この外部素材は pL^AT_EX では使用することができません。英文版の取扱説明書では、上記のPDFは、挿入▷ファイル▷外部素材でひな型にPDFページを指定して挿入されていますが、ここでは挿入▷画像を使用して挿入しています。

ラスター画像 これはビットマップ画像に使用します。外部素材ダイアログを使って入れる画像は、第 1.1 節にあるように**画像**ダイアログを使って挿入される画像と同じように取り扱うことができます。右記は、ラスター画像を挿入した例です。



表計算表 このひな型では、**Gnumeric**・OpenDocument・Microsoft Office XML・Office Open XML 形式の表計算表を挿入することができます。この機能を使用するには、L^yX の初期設定中に `ssconvert` プログラムが検出される必要があります。このプログラムは Gnumeric の一部ですので、Gnumeric 以外の形式の表を読み込む場合には、Gnumeric が導入済みである必要があります。用例と制約については、L^yX 用例ファイル `spreadsheet.lyx` をご覧ください。

ベクトル画像 これは、ベクトル画像に使用することができます。外部素材ダイアログを使って入れる画像は、第 1.1 節にあるように**画像**ダイアログを使って挿入される画像と同じように取り扱うことができます。右記は、ベクトル画像を挿



入した例です。

XFig このひな型は、**Xfig** プログラムで作成した画像をサポートします。

7. 外部文書構成要素

外部素材ダイアログのファイルタブ中で下書きオプションを指定すると、出力中には挿入したファイルのパスだけが表示されます。

LyX 中での外部素材は、ダイアログ内の LaTeX と LyX のオプションタブにある LyX 内に表示オプションの指定に従って、**表計算表: Table.ods** のようなボックスとして表示されるか、画像として表示されます。

取扱説明書**カスタマイズ**篇には、自製のひな型を定義する方法が説明されています。

7.2. 子文書

子文書は、大きな部や節を複数含む長大な文書を作成するときに使われます。メンテナンス上の理由から、文書をいくつかのファイルに分割して、個別に修正できるようにした方が便利なことはよくあり、場合によってはそうすることが必要な場合さえあります。こうした個別文書がいわゆる「子文書」であり、親文書はこれら子文書を連結して、文書全体やその一部を印刷できるようにします。子文書は、L^AT_EX プリアンブルや書誌情報や相互参照用ラベルなどの親文書の内容を引き継ぎます。

親文書を開くことなく子文書で作業できるようにするには、子文書の **文書▷設定▷文書クラス** メニューで親文書を指定してください。すると、子文書を編集する際には、LyX がバックグラウンドでこの親文書を扱うようになります。

取り込んだ文書は、LyX 中で **内包(include): DummyDocument1.lyx** のようなボックスとして表示されます。子文書を親文書に取り込むには、**挿入▷ファイル▷子文書** メニューを使用します。するとダイアログが現れ、以下のように4つの取り込み方法を選択することができます。

内包 (include) LyX 文書と L^AT_EX 文書は、内包することができます。子文書ダイアログの編集ボタンを押すと、内包された文書が、LyX 中に新規ファイルタブとして開き、修正ができるようになります。

以下は、**内包 (include)** を使用して挿入した子文書です。

7.2.1. 外部文書の第1小節

これは、ファイルが別の文書にどのように挿入されるかを示すための、短いダミー子文書です。

7. 外部文書構成要素

取り込んだファイルの節番は、各節を親文書に挿入した順序にしたがって取り込まれます。たとえば、上で取り込まんだ例示文書には、この節の小節として番号を振られた小節があります。取り込んだ文書のラベルは、第 7.2.1 小節、のように参照することができます。

子文書のプリアンブルは無視され、親文書のプリアンブルだけが使用されます。親文書に同じ名称の派生枝がない限り、親文書は子文書の派生枝を無視します。取り込まれた文書は、新規頁から始まるように挿入され、改頁して終わります。

[註] L_YX または L^AT_EX ファイルを内包する際、子文書が親文書と異なる文書クラスを使用している場合には、予期せぬ出力を避けるために、文書を書き出したり閲覧する際に警告が発せられます。

入力 (input) この方法は、内包による方法に非常によく似ています。異なっている点は、

- 入力ファイルは、新規頁として始まることはなく、終わりに改頁されることもありません。
- 入力ファイルは、L_YX 設定の **操作性▷表示** の下にある **画像を表示** を有効にすると、L_YX 中でプレビューすることができます。
- **文書▷設定▷子文書** メニューから入力ファイルを除外することはできません。

右は、**入力 (input)** を使って挿入した子文書です。

7.2.2. 外部文書の第 2 小節

これは、ファイルが別の文書にどのように挿入されるかを示すための、もう一つの短いダミー子文書です。

原文儘 (verbatim) この方法では、テキストファイルなら何でも取り込むことができます。出力中では、ファイルはそのソースコードの通りに表示され、テキスト中に用いられているコマンドは実行されません。**出力の空白に印付けをする** オプションを有効にすると、ソースコード中のすべての空白文字に「」文字を表示させることができます。**挿入▷ファイル▷平文** メニューを使う方法との違いは、こちらは文書の内容が L_YX 中に表示されないことです。

以下は、原文儘として挿入された子文書です。

```
This is a small dummy child text document to show how files can be inserted into a
```

以下は、**出力の空白に印付けをする** オプションを指定して、原文儘として挿入された子文書です。

```
This is a small dummy child text document to show how files can be inserted into a
```

【註】上記の例でわかるように、原文儘 (verbatim) として取り込んだ文書の文章は、文書行の終わりで改行されません。

【訳註】原文儘で日本語のファイルを読み込むときは、子文書のエンコーディングを文書エンコーディングに合わせてください。さもないと出力が文字化けします。

プログラムリスト この型については、第8章に説明されています。

【註】同一文書中に、同じ文書を異なる方法で二度取り込むと、 \LaTeX で問題が発生する可能性があります。

コンパイル時間を短縮したり、執筆中の book の特定の章のみを配布したいなどの理由で、親文書内の特定の子文書のみを表示させたり書きだしたりしたい場合には、子文書の一部を出力から排除することができます。これは、**文書**▷**設定**▷**子文書**メニューから行うことができます。その一覧中の各子文書の**出力に含める**列をダブルクリックすることによって、その子文書を排除したり取り込んだりすることができます(これは「入力」ではなく「内包」で埋めこまれた文書に対してのみ機能することにご注意ください)。

文書全体のカウンタ及び参照のセクションでは、ページ番号や参照、節カウンタ等をどう取り扱うか、設定することができます。

- オプション**厳密に管理**では、 \LaTeX は、すべてのページ番号と参照等が、文書全体を出力した時と同じように正しくなるようにします。これは、選択した部分文書が、文書全体のコンテキストの中で表現されるのと全く同じようにしておきたい時に便利です。しかしながら、この機能を実現するには、 \LaTeX はバックグラウンドで文書全体を処理しなくてはならず、長い時間がかかります。
- オプション**管理しない**では、 \LaTeX は親文書と内包された文書のみコンパイルします。したがって、カウンタは完全な文書とは異なり、除外された子文書への参照は解決されません。もしコンパイルの時間を節約し、カウンタや参照が必ずしも正しくなくてもいい場合には、このオプションが最速ですので、これを選択してください。
- オプション**ほぼ管理**はいわば上記2つの中間です。これが選択されると、 \LaTeX は最初の出力では全文書をコンパイルし、カウンタと参照を正しく設定します。その後の出力では、外部子文書に変更が加えられた場合のみ、全文書をコンパイルします。内包されたファイルを編集する限りは、このアプローチは、カウンタと参照をほぼ正しく保ちつつ、2番目の選択肢と同じ速度を持ちます。これがほぼ正しいとしかいえないのは、内包文書の大きさが変われば、明らかに除外された文書のページ割りも変わるはずですが、この方法ではその変化までは追跡できないからです。

あるいは、取り込んだファイルの各々を派生枝の中に入れる方法があります¹。派生枝を有効化したり無効化したりすることによって、取り込んだファイルのどれを出力に表示させるかを定めることができます。この方法は、入力で取り込んだ部分文書についても動作するという利点があります。ただし、カウンタと参照は、この方法では維持されるとは限りません。

¹派生枝は、 \LaTeX ユーザーの手引きの派生枝の節に説明されています。

8. プログラムコードリスト

プログラムコードを取り込んで組版するには、**挿入▷プログラムリスト**メニューを使用して挿入されるプログラムリスト差込枠を使用することができます。ここで使用される L^AT_EX パッケージ **listings** または **minted** は、使用中の文書にプログラムソースコードを挿入する強力な柔軟な方法を提供します。一文書中では、これらのパッケージのうち一つしか使うことができません。既定値では、**listings** が用いられます。

プログラムリスト差込枠を右クリックすると、文脈依存メニューが開き、リスト書式を設定することができます。

既定では、出力におけるプログラムリストは、新しい段落として始まります。配置オプションで**行内リスト**を指定すると、リストは `int a=5;` のように行内に印字されます。

フロートオプションは、プログラムリストフロートを生成するものですが、ここで、第 3.5 節に説明されているフロート配置オプションに従って、「h」「t」「b」「p」の配置オプションを指定することができます。配置オプションは組み合わせて使用することができ、「htbp」のように区切りを入れることなく入力することができます。「h」オプションは効かない場合がありますが、フロートでないプログラムリストにもキャプションを入れることができ、また参照もできるので、この場合には**フロートオプション**を使用しなければ大丈夫です。

挿入▷キャプションメニューを使えば、行中でないプログラムリストには、キャプションを付けることができます。プログラムリストは、フロート同様、リスト 8.1 のように参照することができます。キャプションのついたプログラムリストの全一覧を、**挿入▷一覧/目次▷プログラムリスト一覧**で作成することができます。一覧の項目には、プログラムリストのキャプションとプログラムリスト番号が入ります。

プログラムリスト 8.1: Example Listing Float

```
# Example listing float
def func(param):
    'this is a python function'
    pass
```

[註] **listings** パッケージは、既定ではプログラムリストの前にキャプションを置きます。

minted パッケージは、フロートの場合、既定ではプログラムリストの後にキャプションを置きます。フロートでない場合には、キャプションは、一行目に入れられた場合にはプログラムリストの前に組版され、そうでなければリストの後に置かれます。常にプログラムリストの前に置くように強制するためには、プレアンブルの最後に下記

8. プログラムコードリスト

の行を加えてください。

```
\@ifundefined{newfloat}
{\usepackage{float}}{}
\floatstyle{plaintop}
```

プログラムリストダイアログでプログラミング言語を指定すると、その言語のキーワードが識別されて、出力中で特別に組版されます。上の例示プログラムリストでは、Pythonのキーワード「def」が識別され、ボールド体で出力されます。

[註] タイプライタフォントを使用している際にキーワードがボールド体にならない場合には、おそらくお使いの `typewriter` フォントがボールド体を提供していません。この場合には、文書▷設定▷フォントメニューで別のフォントを選択してください (*LuxiMono*・*BeraMono*・*Courier* の各フォントはボールド体を提供します)。

[訳註] プログラムリスト差込枠中では、日本語のエンコーディングを表示できないため、日本語は使用できません。プログラムリスト差込枠中のキャプションでも同様です。

プログラムリストダイアログの**行番号**のセクションでは、行番号の様式を指定することができます。行間フィールドでは、どの行に番号を振るかを指定する数字を入れることができます。たとえば、「3」と入れると、3行ごとに数字が振られます。

さらに行の範囲を指定することもでき、この場合にはこれらの行のみが出力に表示されます。**拡張文字コード表を使う**オプションは、`listings` パッケージにのみ有効であり、プログラムリスト中でドイツ語のウムラウトのような国字を使用する場合に使用します。

[訳註] これは、`pLATEX` ではうまく動作しません。

以下は、左側の行番号・行間「3」・言語「Python」・「拡張文字コード表を使う」オプション・「空白を記号で表示」オプション・行範囲「3-8」で設定したプログラムリストの例です。

```
□□□□pass
2 def func (param):
  'This is a German word: Tschuss'
  pass
5 def func (param):
  'this is a python function'
```

プログラムリスト中でタブュレータを使う場合は、タブュレータが確保する文字の数を**タブュレータの大きさ**フィールドで指定することができます。

[註] `listings` パッケージにはバグがあり、前にプログラムリストがある毎に、行番号が一行ずつずれます。上記プログラムリストで3行目と6行目に番号が振られる代わりに、2行目と5行目に振られているのはこのためです。

また、ファイルの各行をプログラムリストとして印字することも可能です。そのためには、挿入▷ファイル▷子文書メニューを使用して**プログラムリスト**型を選択してく

ださい¹. 子文書ダイアログでは、テキストボックスにプログラムリストのパラメータを指定することができます. 使用できる全パラメータの一覧を見たい場合は、テキストボックス中で疑問符「?」を打鍵してください.

子文書プログラムリストを参照したい場合には、子文書ダイアログの対応するフィールドにラベル文を書き込んでください. すると、このラベルは挿入▷相互参照メニューを使って参照できるようになります.

プログラムリスト 8.2は、ファイルを使ったプログラムリストの一例です. ここでは、このLyX ファイルの 10 行目から 15 行目が表示されています.

プログラムリスト 8.2: Lines 10 - 15 of this LyX file

```
% Correction for PDF bookmarks
%\usepackage[dvipdfm,bookmarks=true,bookmarksnumbered=true,
  bookmarkstype=toc]{hyperref}
\ifpdf
\ifnum 42146=\euc"A4A2 \AtBeginDvi{\special{pdf:tounicode EUC
-UCS2}}\else
\AtBeginDvi{\special{pdf:tounicode 90ms-RKSJ-UCS2}}\fi
\fi
```

リストに数式を入れるには、以下のようにします.

1. リストダイアログを開き、詳細タブのフィールド中に `mathescape=true` と書き加えます.
2. リスト外で数式を作成します.
3. 数式の内容 (数式差込枠ではなく) をクリップボードにコピーします (短縮キー).
4. リストに「`$$`」と書き、クリップボードの内容をドルマークの間に貼り付けます (短縮キー).

プログラムリスト 8.3: Example Listing float containing a formula

```
# Example listing float
def func(param):
  #  $\prod_{n=1}^{\infty} \frac{1}{x^2}$ 
  pass
```

大域的なプログラムリストの設定は、文書▷設定▷リストダイアログで設定することができます. ここで利用できるオプションの一覧を得るには、プログラムリスト設定フィールドで疑問符「?」を打鍵してください.

`listings` 及び `minted` パッケージについての詳しい情報については、取扱説明書 [14, 15] をご参照ください.

¹他の子文書型については、第 section 7.2 節に説明があります.

A. LyXで利用できる単位

この文書で用いられる単位の理解のために、表 A.1 に、LyX で使うことのできる単位をまとめておきます。

表 A.1.: 単位

単位	名称／摘要	L ^A T _E X コマンド
bp	ビッグポイント (72 bp = 1 in)	-
cc	シゼロ (1 cc = 12 dd)	-
cm	センチメートル	-
dd	ディドー (72 dd ≈ 37.6 mm)	-
em	現在のフォントの文字「M」の幅	-
ex	現在のフォントの文字「x」の高さ	-
in	インチ	-
mm	ミリメートル	-
mu	数式単位 (1 mu = 1/18 em)	-
pt	ポイント (72.27 pt = 1 in)	-
pc	パイカ (1 pc = 12 pt)	-
sp	スケールポイント (65536 sp = 1 pt)	-
列幅%	組段幅の%	<code>\columnwidth</code>
行幅%	連続する本文 2 行のベースライン間の高さの%	<code>\baselineskip</code>
行幅%	行幅の%	<code>\linewidth</code>
ページ高%	用紙高さの%	<code>\paperheight</code>
ページ幅%	用紙幅の%	<code>\paperwidth</code>
画像の縮尺%	元の画像幅の%	
本文高%	テキスト高さの%	<code>\textheight</code>
本文幅%	テキスト幅の%	<code>\textwidth</code>

B. 画像を含む場合のファイル出力形式

B.1. DVI

このファイル型の拡張子は「.dvi」です。このファイル型は完全に可搬性があるので「device-independent (DVI: デバイス独立型)」の名称がついています。これはあるコンピュータから別のコンピュータへ、何らの変換も行うことなしに移動することができます。このファイル書式が開発された当時には、これは当然のことでは全くありませんでした。DVIは手っ取り早くプレビューしたり、PostScriptなどの他の出力形式への中間段階として使用されます。

【註】 DVI ファイルには画像は含まれていません。画像はリンクされているだけです。ですから、お使いの.dvi ファイルを別のコンピュータに移すときには、このことを忘れてはいけません。この性質は、DVIを閲覧する際にお使いのコンピュータを遅くしてしまうことがあります。DVIビューアは、DVIをスクロールする際、画像を表示できるようにバックグラウンドで変換しなくてはならないためです。ですから、画像のたくさんあるファイルの場合は、PDFを使用することをお勧めします。

使用中の文書は、**ファイル**▷**書き出し**▷**DVI**メニューを使用して、DVIに書き出すことができます。表示メニューから、文書をDVIとして表示させることもできます。

B.2. PostScript

このファイル型の拡張子は「.ps」です。PostScriptは、Adobe社によってプリンタ言語として開発されました。したがって、このファイルにはプリンタがファイルを印字するコマンドが書かれています。PostScriptは、計算を行ったり図表や画像を描いたりすることもできるので、「プログラミング言語」と見なすことができます¹。この機能のためにPostScriptはPDFよりも大きくなりがちです。

PostScriptは、「Encapsulated PostScript (EPS: ファイル拡張子「.eps」)」形式の画像のみを含めることができます。LyXでは、既知の画像形式であればどれでも文書中で使用することができるようになっているため、LyXはバックグラウンドで画像をEPSに変換しなくてはなりません。文書中に、仮に50の画像があれば、文書を表示したり書き出ししたりするたびに、LyXは50もの変換を行わなくてはなりません。すると、LyXで行う作業手順が、著しく遅れてしまいます。したがって、もし

¹これについてももう少し学ぶ気があれば、L^AT_EX パッケージ **PSTricks**[19] を見てみてください。

B. 画像を含む場合のファイル出力形式

PostScript を使用することを考えているならば、この問題を回避するために、画像を直接 EPS として入れておく方がよいでしょう。

文書は、**ファイル**▷**書き出し**▷**Postscript** メニューを使用して、PostScript に書き出すことができます。使用中の文書は、**表示メニュー**か**ツールバーボタン**を使用することで PostScript として表示させることができます。

B.3. PDF

このファイル型の拡張子は「.pdf」です。この「Portable Document Format (PDF)」は、Adobe 社によって PostScript の派生として開発されました。PDF は、PostScript よりも圧縮されており、かなり少ないコマンドしか使いません。「portable (可搬的)」の名称が示すとおり、この形式は、どのコンピュータシステムでも処理することができます。印字出力は全く同じように出すことができます。

PDF は画像を、同じ PDF 形式か、「Joint Photographic Experts Group (JPG : ファイル拡張子「.jpg」または「.jpeg」)」形式か、「Portable Network Graphics (PNG : ファイル拡張子「.png」)」として含めることができます。それにもかかわらず、LyX は画像をこれらの形式のいずれかにバックグラウンドで変換を行うので、他の画像形式も自由に使用することができます。しかし、PostScript の節で説明したように、この画像変換は作業手順を遅らせます。したがって、上記で述べた3つの画像形式のうちいずれかを使用することをお勧めします。

使用中の文書は、**ファイル**▷**書き出し**メニューを使用して、以下の3つの方法のうちいずれかで PDF に書き出すことができます。

PDF (ps2pdf) これは ps2pdf プログラムを使って、現在のファイルの PostScript 版から PDF を生成します。この PostScript 版は、中間段階に DVI 版を使用する dvips プログラムによって生成されます。つまり、この書き出し方法は、3段階の変換を伴います。

PDF (dvi2pdf) これは dvi2pdf プログラムを使って、現在のファイルをバックグラウンドで DVI に変換し、2段階目で PDF に変換します。

PDF (pdflatex) これは pdflatex プログラムを使って、現在のファイルを直接 PDF に変換します。

pdflatex は現在の PDF 版のすべての機能をサポートし、高速で問題なく安定して動作するので、PDF (pdflatex) を使用することが推奨されます。dvi2pdf プログラムは開発が停止しており、若干時代遅れになっています。

使用中の文書を PDF として表示するには、**表示メニュー**か**ツールバーボタン**（これは PDF (pdflatex) を使用します）を用います。

C. dvips 色名一覽

色	色名	色	色名	色	色名	色	色名
	Apricot		Aquamarine		Bittersweet		BlueGreen
	BlueViolet		BrickRed		Brown		BurntOrange
	CadetBlue		CarnationPink		Cerulean		CornflowerBlue
	Dandelion		DarkOrchid		Emerald		ForestGreen
	Fuchsia		Goldenrod		Gray		GreenYellow
	JungleGreen		Lavender		LimeGreen		Mahogany
	Mahogany		Maroon		Melon		MidnightBlue
	Mulberry		NavyBlue		OliveGreen		Orange
	OrangeRed		Orchid		Peach		Periwinkle
	PineGreen		Plum		ProcessBlue		Purple
	RawSienna		RedOrange		RedViolet		Rhodamine
	RoyalBlue		RoyalPurple		RubineRed		Salmon
	SeaGreen		Sepia		SkyBlue		SpringGreen
	Tan		TealBlue		Thistle		Turquoise
	Violet		VioletRed		WildStrawberry		YellowGreen
	YellowOrange						

D. 第(2.1)式の説明

n 個の表セルの全幅 $W_{全:n}$ は

$$W_{全:n} = n \cdot (W_{指:n} + 2 \cdot \backslash\text{tabcolsep}) + (n + 1) \cdot \backslash\text{arrayrulewidth} \quad (\text{D.1})$$

で計算できます. ここで $W_{指:n}$ は, すべてのセルが持つ固定幅です. $\backslash\text{tabcolsep}$ は, セルテキストとセル罫線のあいだを指す L^AT_EX 長であり, 既定値は 6 pt です. $\backslash\text{arrayrulewidth}$ は, セル罫線幅であり, 既定値は 0.4 pt です.

第 (D.1) 式にしたがえば, 連結列の全幅 $W_{全:連結列}$ は

$$W_{全:連結列} = W_{指:連結列} + 2 \cdot \backslash\text{tabcolsep} + 2 \cdot \backslash\text{arrayrulewidth} \quad (\text{D.2})$$

となります. n 列を連結する時には各列の全幅は $W_{全:連結列}/n$ となるので, 第 (D.1) 式と第 (D.2) 式が等しいものと置けば, 必要となる固定幅 $W_{指:n}$ を計算することができて,

$$W_{指:n} = \frac{W_{指:連結列} + (1 - n) \cdot (2 \cdot \backslash\text{tabcolsep} + \backslash\text{arrayrulewidth})}{n} \quad (\text{D.3})$$

となります.

参考文献

- [1] Frank Mittelbach and Michel Goossens: *The L^AT_EX Companion Second Edition*. Addison-Wesley, 2004
- [2] Helmut Kopka and Patrick W. Daly: *A Guide to L^AT_EX Fourth Edition*. Addison-Wesley, 2003
- [3] Leslie Lamport: *L^AT_EX: A Document Preparation System*. Addison-Wesley, second edition, 1994
- [4] L^AT_EX パッケージ [arydshln](#)
- [5] L^AT_EX パッケージ [booktabs](#) の取扱説明書
- [6] L^AT_EX パッケージ [caption](#) の取扱説明書
- [7] L^AT_EX パッケージ [diagbox](#)
- [8] L^AT_EX パッケージ [endfloat](#) の取扱説明書
- [9] L^AT_EX パッケージ [footmisc](#) の取扱説明書
- [10] L^AT_EX パッケージ [hypcap](#) の取扱説明書
- [11] L^AT_EX パッケージ [hyperref](#) の取扱説明書
- [12] L^AT_EX パッケージ [KOMA-Script](#) の取扱説明書
- [13] L^AT_EX パッケージ [lettrine](#) 及び用例
- [14] L^AT_EX パッケージ [listings](#) の取扱説明書
- [15] L^AT_EX パッケージ [minted](#) の取扱説明書
- [16] L^AT_EX パッケージ [marginnote](#) の取扱説明書
- [17] L^AT_EX パッケージ [pdfpages](#) の取扱説明書
- [18] L^AT_EX パッケージ [prettyref](#) の取扱説明書
- [19] L^AT_EX パッケージ [PSTricks](#) のウェブページ
- [20] L^AT_EX パッケージ [refstyle](#) の取扱説明書
- [21] L^AT_EX パッケージ [sidecap](#) の取扱説明書
- [22] [INKSCAPE](#) プログラムの [Omit text in PDF](#) オプションの取扱説明書

索引

DVI, → ファイル書式の項 参照

EPS, → 画像形式の項 参照

GIF, → 画像形式の項 参照

JPG, → 画像形式の項 参照

L^AT_EX パッケージ

arydshln, 43, 107

booktabs, 30, 107

calc, 24, 27, 37

caption, 10, 15, 16, 58, 107

colortbl, 31, 33, 45

diagbox, 25, 107

endfloat, 54, 107

fancybox, 74

footmisc, 68, 107

ftnright, 66

hyperref, 14, 51, 107

KOMA-Script, 58, 59, 68, 107

lettrine, 85, 107

listings, 95, 107

marginnote, 69, 107

minted, 95, 107

pdfpages, 88, 107

picinpar, 84

prettyref, 50, 51, 107

PSTricks, 107

refstyle, 50, 51, 107

sidecap, 59, 107

wrapfig, 83

xcolor, 33

PDF, 5, 102

PNG, → 画像画像形式の項 参照

PostScript, → ファイル書式の項 参照

SVG, → 画像形式の項 参照

色付き

表罫線部, 34

表セル部, 31

表の行, 33

回転

素材の——, 36

素材の——, 79

外部文書構成要素, 87

外部素材, 87

子文書, 90

頭文字, 85

画像, → 図の項 参照

画像形式, 5

脚註, 64

配置, 66

付番, 65

キャプション

書式設定, 58

配置, 59

後註, 66

参照

参照位置, 51

自動命名, 51

書式, 50

図への——, 3

表への——, 10

フロートに対する, 49

索引

伸縮

素材の——, 80

図, 1

画像形式, 5
画像ダイアログ, 1
設定グループ, 2
——の回転, 1, 79
——の伸縮, 1, 80
フロート, 3

多頁表, 10

脚註, 13
キャプション, 14
キャプションの幅, 15
強制改頁, 18
計算, 18
参照, 14
縦横揃え, 13
フロート, 18
ページごとに異なるキャプション,
16

単位, 99

註釈

L_AT_EX ノート, 63
脚註, 64
傍註, 68

表, 7

色付き, 31
改行, 22
セルの回転, 36
揃え, 30
ダイアログ, 7
多頁——, 10
調整, 38
ツールバー, 8
はじめに, 7
フォーマル様式, 28
フロート, 10
編集メニュー, 9
連結行, 24
連結列, 23

表セル内の複数行, 22

表の色

行全体, 33
罫線部, 34
セル部, 31
表の調整, 38
回転, 79
行の余白取り, 38
罫線幅, 42
伸縮, 80
セルと列の書式, 41
点罫, 43
特殊なセル揃え, 40

ファイル

インクルード, 90

ファイル書式

DVI, 101
PDF, 102
PostScript, 101

プログラムコード, 95

フロート, 47

アルゴリズム, 48
折返しフロート, 83
回転, 55
キャプションの書式設定, 58
キャプションの配置, 59
参照, 49
従属フロート, 55
図, 3
設定, 52
配置, 52
はじめに, 47
表, 10
フロート一覧, 61
横並び, 55
リスト, 95
連番, 48

ボックス

Raiseboxes, 77
色付き, 78
回転した——, 79
詳細な設定, 75
伸縮させた——, 80

- 垂直揃えの仕方, 77
- 寸法, 71
- 装飾, 74
- 段落——(parbox), 76
- ハイフネーションの回避, 77
- はじめに, 71
- ボックスダイアログ, 71
- ミニページ, 75
- 文字揃え, 72
- 文字用, 77
- 本文回り込み, 83
 - 折返しフロート, 83
 - 頭文字, 85
 - 固定オブジェクト, 84
- リスト, 95
- 連結行, 24
- 連結列, 23
 - 計算, 23