

## 名称

groff – GNU roff 言語の簡潔なリファレンス

## 解説

名前 *groff* は、*GNU roff* を表し、roff 植字システムの自由な実装です。groff システムの概観と背景については、**roff(7)** を参照してください。

この文書は、groff で使用されるように、あらかじめ定義された roff 言語の要素の簡潔な説明のみを与えます。古典的な機能と groff の拡張の両方が提供されます。

歴史的に、roff 言語は、*troff* と呼ばれました。*groff* は、古典的なシステムと互換性があり、適切な拡張を提供しています。それで、GNU では、用語 *roff*, *troff* と *groff* 言語は、同義語として使用されます。しかしながら、*troff* は、わずかに古典的な側面をより参照する傾向があるのに対して、*groff* は、GNU の拡張を強調し、*roff* は、言語のための一般的な用語です。

このファイルは、より詳細で、現実の、簡潔な情報を含んでいる *groff info(1)* ファイルに見つかる完全な文書の唯一の簡潔なバージョンです。

groff の文書を書くための一般的な構文は、比較的容易ですが、roff 言語の拡張を書くことは、少し難しいかもしれません。

roff 言語は、行指向です。ただ 2 種類の行、制御行とテキスト行があります。制御行は、デフォルトでピリオド “.” または単一のクォート “'” である、制御文字で始まります。他のすべて行は、テキスト行です。

**制御行 (control line)** は、コマンドを表し、オプションで引数を付けます。それらには、次の構文があります。先導する制御文字は、コマンド名を続けることができます。もしあるなら、引数は、コマンド名とそれら自体の間を空白によって区切ります、例えば、

```
.command_name arg1 arg2
```

インデントーション (段付け) のために、先導する制御文字とコマンド名の間にいくらかでも空白またはタブ文字を挿入することができますが、制御文字は、行の最初の位置になければなりません。

**テキスト行 (text line)** は、印刷 (表示) される部分を表しています。それらは、先導するバックslash ‘\’ によって認識される、エスケープシーケンスによって修正することができます。これらは、行間または単語間の書式要素または関数です。これらのうちのいくつかは、単一のクォート “'” によって区切られる引数を取り、他は、開き括弧 ‘(’ または角括弧 ‘[’ と ‘]’ によって導入されエンコーディングの長さによって調節されます。

roff 言語は、マクロのような言語の拡張を書くための柔軟な道具を提供しています。マクロの定義を解釈するとき、roff システムは、**コピーモード** と呼ばれる特別な操作モードに入ります。

コピーモードの振る舞いは、まったくトリッキーであるかもしれませんが、安全な使用を保証するいくつかの規則があります。

1. 印刷可能なバックスラッシュは、`\e` として示されなければなりません。もっと正確に言う  
と、`\e` は、現在のエスケープ文字を表しています。バックスラッシュのグリフを得るため  
に、`\(rs` または `\[rs]` を使用します。
2. すべてのバックスラッシュを 2 重にします。
3. 特別な間隔でない文字 `\&` ですべてのテキスト行を始めます。

これは、最も効率的なコードを生成しませんが、それは、最初的手段として動作するべきです。よ  
りよい方策のために、groff 情報ファイルと `groff_tmac(5)` を参照してください。

roff ソースファイルを読むことは、より容易です、ただすべての 2 重のバックスラッシュをすべ  
てのマクロの定義の単一のものに減少します。

## GROFF 要素

roff 言語の要素は、書式情報をテキストファイルに追加します。基本的な要素は、roff を本格的な  
プログラミング言語にする、あらかじめ定義されたコマンドと変数です。

引数を付ける可能性がある、2 種類の roff コマンドがあります。**要求** は、ドット `.'` または `''`  
で始まるそれら自体の行に書かれるのに対して、**エスケープシーケンス** は、バックスラッシュ  
`\` で始まる行間と単語間の書式要素です。

ユーザは、**de** 要求を使用してユーザ自身の書式コマンドを定義することができます。これらのコ  
マンドは、**マクロ** と呼ばれますが、それらは、要求のように正確に使用されます。マクロパッ  
ケージは、groff 言語で書かれたマクロのあらかじめ定義されたセットです。ユーザ自身のエス  
ケープシーケンスを作成するユーザの可能性は、著しく制限されていて、特殊文字のみをマップ  
することができます。

groff 言語は、異なったインタフェースがあるいくつかの種類の変数を提供しています。あらかじ  
め定義された変数がありますが、ユーザは、同様に、ユーザ自身の変数を定義することができま  
す。

**文字列** 変数は、文字シーケンスを格納します。それらは、**ds** 要求で設定され、`\*` エスケープシー  
ケンスによって取り出されます。文字列は、変数を持つことができます。

**レジスタ** 変数は、数値の値、スケール単位がある数値と時々文字列に似のオブジェクトを格納す  
ることができます。それらは、**nr** 要求で設定され、`\n` エスケープシーケンスによって取り出され  
ます。

**環境** によって、ユーザは、後で再利用するために、行長さ、フォントサイズ、などのような、グ  
ローバルな書式パラメータを一時的に格納することができます。これは、**ev** 要求によって行われ  
ます。

**フォント** は、名前または内部の番号によって識別されます。現在のフォントは、**ft** 要求または `\f`  
エスケープシーケンスによって選択されます。各デバイスには、特別なフォントがありますが、

次のフォントは、すべてのデバイスで利用可能です。**R** は、標準のフォントであるローマン体です。**B** は、その **ボールド体** に対応するものです。イタリック体のフォントは、**I** と呼ばれ、どこでも利用可能ですが、テキストデバイスで、それは、下線が付けられたローマン体のフォントとして表示されます。グラフィカルな出力デバイスに対して、そこで存在しなさい これらのフォント、**CR**, **CI** と **CB** の固定幅のペンダント (pendant) が存在します。訳注: pendants は、pendant フォントのことと思われる。テキストデバイスで、すべての文字は、とにかく、固定の幅があります。

さらに、いくつかの高度な roff の要素があります。**ディバージョン** (diversion) は、後で使用するために情報をマクロに格納します。**トラップ** は、ページの先頭から行の特定の数、またはディバージョンで、または入力でのような位置の条件です。条件が満たされているとき、自動的に実行される、いくつかのアクションを指示することができます。

より詳細な情報と例は、groff 情報ファイルを見つけることができます。

## 制御文字

特定の条件で特別な制御タスクがある文字の小さなセットがあります。

- ・ ドットは、行の先頭でのみ、または要求 **if**, **ie**, **el** と **while** の条件の後でのみ特別です。ここでは、要求 (またはマクロ) を導入する制御文字です。**\.** エスケープを使用して、特別な振る舞いを遅延することができます。**cc** 要求を使用することによって、ドット **'.'** を特別でない文字にして、異なる文字に制御文字を設定することができます。

すべての他の位置で、それは、ただのドット文字を意味しています。テキストの段落で、それ自体の行で各文を開始することは有利です。

- ・ シングルクォートには、2 の制御タスクがあります。行の始めと、条件付きの要求で、それは、ブレイクしない制御文字です。それは、ドットのように要求を導入することを意味していますが、この要求が改行 (linebreak) を起こさない追加のプロパティがあります。**c2** 要求を使用することによって、異なる文字にブレイクしない制御文字を設定することができます。

2 番目のタスクとして、それは、いくつかの機能的なエスケープシーケンスの最も共通に使用される引数のセパレータです (が、引数の一部でないあらゆるペアの文字は、動作します)。すべての他の位置で、それは、シングルクォートまたはアポストロフィ文字を示します。groff は、**\(cq** エスケープシーケンスを使用して印刷可能な表現を提供しています。

- " ダブルクォートは、要求、マクロと文字列の引数を囲むために使用されます。**ds** と **as** 要求で、引数の先導するダブルクォートは、(先導する空白類を有効にして) 定義される文字列の後で他のすべてを作成して、取り除かれます。エスケープされたダブルクォート **\"** は、コメントとします。そうでなければ、それは、特別ではありません。groff は、**\(dq** エスケープシーケンスを使用して印刷可能な表現を提供しています。

- \** バックスラッシュは、通常、エスケープシーケンスを開始します (これは、**ec** 要求で変更することができます)。エスケープ文字の印刷 (表示) されたバージョンは、**\e** エスケープです。バックスラッシュのグリフ (glyph) 文字を **\rs** によって取得することができます。
- (** 開き括弧は、ちょうど 2 文字から成るエスケープ名または引数を導入するとき、エスケープシーケンスでのみ特別です。groff で、この振る舞いは、**[ ]** 構成によって置き換えることができます。
- [** 開く角括弧は、groff のエスケープシーケンスでのみ特別です。そこで、それは、長いエスケープ名または長いエスケープ引数を導入するために使用されます。そうでなければ、それは、特別ではありません、例えば、マクロの呼び出しで。
- ]** 閉じ角括弧は、groff のエスケープシーケンスでのみ特別です。そこで、それは長いエスケープ名または長いエスケープ引数を終わらせます。そうでなければ、それは、特別ではありません。
- space* 空白文字は、機能的だけの文字です。それらは、要求、マクロと文字列の引数を分離し、そしてテキスト行の単語を分離します。それらは、groff の水平の間隔の計算に制約されます。定義された空白の幅を取得するために、(これは、空白が続いているエスケープ文字である) **\** ' のようなエスケープシーケンス **\l**, **\i**, または **\h** が使用されるべきです。

#### *newline*

テキストの段落で、改行は、ほとんど空白文字のように振る舞います。エスケープされた改行によって、すなわち、行の最後の文字としてバックスラッシュ **\** を指定して、連続行を指定することができます。

- tab* テキストの間にタブ文字が現れるなら、インタプリタは、次のあらかじめ定義されたタブの位置に水平にジャンプをします。タブの位置を処理するための洗練されたインタフェースがあります。

## 数式

**数値** は、追加された単位指定子あり、またはなしの符号つき整数または符号なし整数、または浮動小数点数です。**単位指定子** は、測定の単位のための 1 文字の短縮形です。単位指定子が続いている数値は、サイズ値を意味しています。デフォルトで、数値は、単位指定子がありません、すなわち、それらは、通常の数値です。

*roff* 言語は、次の単位指定子を定義しています。

<b>c</b>	センチメートル
<b>i</b>	インチ
<b>P</b>	パイカ = 1/6 インチ
<b>p</b>	ポイント = 1/72 インチ

<b>m</b>	エム(em) = ポイントのフォントサイズ (文字 'm' の幅)
<b>M</b>	エム (em) の 100 番目
<b>n</b>	エヌ (en) = エム(em)/2
<b>u</b>	実際の出力デバイスのための基本的な単位
<b>v</b>	基本的な単位の垂直行の間隔スケールポイント = ポイントの 1/サイズスケール (フォント <i>DESC</i> ファイルで定義される)
<b>f</b>	65536 で計ります

**数式** は、伝統的な troff ですすでに定義されていた次の算術演算子で上記に定義された数値の組み合わせです。

<b>+</b>	加算
<b>-</b>	減算
<b>*</b>	乗算
<b>/</b>	除算
<b>%</b>	剰余
<b>=</b>	等しい
<b>==</b>	等しい
<b>&lt;</b>	未満
<b>&gt;</b>	より大きい
<b>&lt;=</b>	以下
<b>&gt;=</b>	以上
<b>&amp;</b>	論理積 (AND)
<b>:</b>	論理和 (OR)
<b>!</b>	論理否定
<b>(</b>	式のグループ化
<b>)</b>	現在のグループ化を閉じる

さらに、*groff* は、数式のための次の演算子を追加しました:

<b><i>e1</i>&gt;<i>?e2</i></b>	<i>e1</i> と <i>e2</i> の最大。
<b><i>e1</i>&lt;<i>?e2</i></b>	<i>e1</i> と <i>e2</i> の最小。
<b><i>(c;e)</i></b>	デフォルトの単位指定子として <i>c</i> を使用して <i>e</i> を評価します。

詳細については、*groff* 情報ファイルを参照してください。

## 条件式

**条件式** は、**if**, **ie** と **while** 要求によって引き起こされたテストに現れます。次のテーブルは、条件式の異なったタイプを特徴づけています。

<i>N</i>	数式 <i>N</i> は、その値が 0 より大きいなら、真をもたらします。
<b>!N</b>	<i>N</i> の値が 0 であるなら、真。
<b>'s1's2'</b>	文字列 <i>s1</i> が、文字列 <i>s2</i> と同一であるなら、真。
<b>!s1's2'</b>	文字列 <i>s1</i> が文字列 <i>s2</i> と同一でないなら、真。
<b>cch</b>	利用可能な文字 <i>ch</i> があるなら、真。
<b>dname</b>	<i>name</i> と呼ばれる、文字列、マクロ、転換 (diversion) または要求があるなら、真。
<b>e</b>	現在のページ数は、偶数です。
<b>o</b>	現在のページ数は、奇数です。
<b>mname</b>	<i>name</i> と呼ばれる色があるなら、真。
<b>n</b>	フォーマッタは、 <b>nroff</b> です。
<b>rreg</b>	<i>reg</i> と名前が付けられたレジスタがあるなら、真。
<b>t</b>	フォーマッタは、 <b>troff</b> です。
<b>Ffont</b>	<i>font</i> 名前が付けられたフォントが存在するなら、真。
<b>Sstyle</b>	<i>style</i> と名前が付けられたスタイルが、登録されているなら、真。

## 要求

このセクションは、あらかじめ定義された要求のための簡潔なリファレンスを提供しています。groff で、要求とマクロの名前は、任意の長さを指定することができます。括弧で囲むか、または長い名前のマークが、必要とされません。

ほとんどの要求は、1 つ以上の引数を取ります。引数は、空白文字によって区切られます (タブでなく!)。それらの長さまたは数に対して固有の制限はありません。引数は、ペアのダブルクォートによって囲むことができます。引数が空白文字を含んでいるなら、これは、とても便利です、例えば、"*arg with space*" は、単一の引数を示します。

いくつかの要求は、異なる振る舞いで、オプションの引数があります。これらの詳細は、すべてがここで説明されるわけではありません。すべての詳細については、groff 情報ファイルと **groff\_diff(7)** を参照してください。

次の要求の仕様で、ほとんどの引数名は、説明的であるように選択されました。次の明示的な意味だけは、明確化を必要とします。

<i>c</i>	単一の文字を示します。
<i>font</i>	フォント名またはフォント番号として指定されるいずれかのフォント。
<i>anything</i>	行の終りまで、または \{ と \} 内のすべての文字。
<i>n</i>	は、整数値に評価する数式です。
<i>N</i>	は、符号付き、または符号なしの任意の数式です。

$\pm N$  は、以下に説明されている、その符号に依存している 3 つの意味があります。

$\pm N$  のように定義される式が '+' 符号で始まるなら、式の結果の値は、関連した要求のためにすでに存在する固有の値に加算されます、例えば、数値レジスタに加算します。式が '-' で始まるなら、式の値は、要求値から減算されます。

符号なしで、 $N$  は、既存の値に直接置き換えます。負の数値に割り付けるためには、0 を先頭に付けるか、または括弧内で負の数を囲みます。

## 要求の簡潔なリファレンス

. 空行、無視されます。文書を構造化するために役に立ちます。

.\" anything

行全体がコメントです。

.ab string

標準エラーに *string* を印刷 (表示) し、プログラムを終了します。

.ad 現在の調整 (adjust) モードの出力行のための行の調整を開始します。

.ad *c* モード *c* (*c*=l, r, b, n) で行の調整を開始します。

.af register *c*

形式 *c* をレジスタ (*c*=l, i, I, a, A) に割り当てます。

.aln alias register

register のためのエイリアス名を作成します。

.als alias object

要求、文字列、マクロまたは転換 (diversion) *object* のためのエイリアス名を作成します。

.am macro

.. に遭遇するまで、*macro* に追加します。

.am macro end

.end が呼び出されるまで、*macro* に追加します。

.am1 macro

.am と同じですが、マクロ展開の間にスイッチがオフにされた互換モードがあります。

.am1 macro end

.am と同じですが、マクロ展開の間にスイッチがオフにされた互換モードがあります。

.ami macro

.. に遭遇するまで、名前が文字列レジスタ *macro* に含まれているマクロに追加します。

.ami macro end

間接的にマクロに追加します。*macro* と *end* は、内容が、それぞれ、マクロの名前と終了のマクロのために差し挟まれる文字列レジスタです。

**.ami1 macro**

**.ami** と同じですが、マクロ展開の間にスイッチがオフにされた互換モードがあります。

**.ami1 macro end**

**.ami** と同じですが、マクロ展開の間にスイッチがオフにされた互換モードがあります。

**.as stringvar anything**

*anything* を *stringvar* に追加します。

**.as1 stringvar anything**

**.as** と同じですが、マクロ展開の間にスイッチがオフにされた互換モードがあります。

**.asciify diversion**

転換 (diversion) の ASCII 文字、空白といくつかのエスケープシーケンスを未書式化 (unformat) します。

**.backtrace**

標準エラーに入力のバックトレース (backtrace) を印刷 (表示) します。

**.bd font N**

$N-1$  単位によって *font* を強調表示 (embolden) します。

**.bd S font N**

現在のフォントが *font* であるとき、特別のフォント *S* を強調表示 (embolden) します。

**.blm** 空行のマクロを未設定とします。**.blm macro**

空行マクロを *macro* に設定します。

**.box** 現在の転換 (diversion) を終了します。**.box macro**

部分的に満たされた行を省略して *macro* に転換 (divert) します。

**.boxa** 現在の転換 (diversion) を終了します。**.boxa macro**

部分的に満たされた行を省略して *macro* に転換 (divert) して、追加します。

**.bp** 現在のページを排出 (eject) し、新しいページを開始します。**.bp ±N** 現在のページを排出 (eject) します。次のページ番号  $\pm N$ 。**.br** 改行。**.brp** 改行し、出力行を拡張します。**\p** と同じ。**.break** while ループを脱出します。**.c2** 非改行制御文字 “*’*” をリセットします。**.c2 c** 非改行制御文字を *c* に設定します。



- .cc** 制御文字を ‘.’ にリセットします。
- .cc c** 制御文字を *c* に設定します。
- .ce** 次の入力行をセンタリング (中央揃え) します。
- .ce N** 次の *N* 行の入力行をセンタリング (中央揃え) します。
- .cf filename**  
処理されていないファイル *filename* の内容を標準出力または転換 (diversion) にコピーします。
- .cflags mode c1 c2...**  
*mode* 番号にしたがって文字 *c1*, *c2*, ... を扱います。
- .ch trap N**  
*trap* 位置を *N* に変更します。
- .char c anything**  
文字列 *anything* として文字 *c* を定義します。
- .chop object**  
マクロ、文字列または転換 (diversion) *object* の最後の文字を切り落とします。
- .close stream**  
*stream* をクローズします。
- .color** 色を有効にします。
- .color N**  
*N* が 0 であるなら、色を無効にし、そうでなければ、それらを有効にします。
- .composite from to**  
合成のグリフ名を構成する間にグリフ名 *from* から グリフ名 *to* にマップします。
- .continue**  
while ループの現在の繰り返しを終了します。
- .cp** 互換モードを有効にします。
- .cp N** *N* が 0 であるなら、互換モードを無効にし、そうでなければ、それを有効にします。
- .cs font N M**  
*font* のための定数文字幅を em *M* で *N*/36 em に設定します。
- .cu N** troff の **.ul** のように nroff の下線を継続します。
- .da** 現在の転換 (diversion) を終了します。
- .da macro**  
*macro* を転換 (divert) し、追加します。
- .de macro**  
.. に遭遇するまで、マクロを定義するか、または再定義します。

**.de** *macro end*

**.end** が呼ばれるまで、マクロを定義するか、または再定義します。

**.dei** *macro*

**.de** と同じですが、マクロ展開の間にスイッチがオフにされた互換モードがあります。

**.dei** *macro end*

**.de** と同じですが、マクロ展開の間にスイッチがオフにされた互換モードがあります。

**.defcolor** *color scheme component*

名前 *color* で色を定義するか、または再定義します。*scheme* は、**rgb**, **cym**, **cymk**, **gray** または **grey** を指定することができます。*component* は、0 から 1 までの範囲 (デフォルトの単位指定子は、**f** です) の分数として、先導する # で 2 桁の 16 進数の色構成要素の文字列として、または 2 つの先導する # で 4 桁の 16 進数の構成要素の文字列として指定される単一の構成要素を指定することができます。色 **default** を再定義することができません。

**.dei** *macro*

..*に遭遇するまで、名前が文字列レジスタ *macro* に含まれているマクロを定義するか、または再定義します。*

**.dei** *macro end*

マクロを間接的に定義するか、または再定義します。*macro* と *end* は、内容が、それぞれ、マクロの名前と終わりのマクロのために差し挟まれる文字列レジスタです。

**.dei1** *macro*

**.dei** と同じですが、マクロ展開の間にスイッチがオフにされた互換モードがあります。

**.dei1** *macro end*

**.dei** と同じですが、マクロ展開の間にスイッチがオフにされた互換モードがあります。

**.di** 現在の転換 (diversion) を終了します。

**.di** *macro*

*macro.* に転換 (divert) します。

**.do** *name*

無効にされた互換モードで **.name** を解釈します。

**.ds** *stringvar anything*

*stringvar* を *anything* に設定します。

**.ds1** *stringvar anything*

**.ds** と同じですが、マクロ展開の間にスイッチがオフにされた互換モードがあります。

**.dt** *N trap*

転換 (diversion) トラップを位置 *N* に設定します (デフォルトの単位指定子は、**v** です)。

- .ec** エスケープ文字を ‘\’ にリセットします。
- .ec c** エスケープ文字を *c* に設定します。
- .ecr** **.ecs** で保存されたエスケープ文字を復元します。
- .ecs** 現在のエスケープ文字を保存します。
- .el anything**  
if-else (**ie**) 要求のための else 部分。
- .em macro**  
*macro* は、入力の終わりの後に実行されます。
- .eo** エスケープ文字のメカニズムをオフに切り替えます。
- .ev** 以前の環境に切り換えます。
- .ev env** 環境の数値または名前 *env* をプッシュし、それに切り換えます。
- .evc env** 環境 *env* の内容を現在の環境にコピーします。プッシュまたはポップを行いません。
- .ex** roff の処理を終了します。
- .fam** 前のフォントファミリに戻ります。
- .fam name**  
現在のフォントファミリを *name* に設定します。
- .fc** フィールドメカニズムを無効にします。
- .fc a** フィールドデリミタを *a* に設定し、埋め込み文字を空白にします。
- .fc a b** フィールドデリミタを *a* に設定し、埋め込み文字を *b* にします。
- .fchar c anything**  
何でも文字列にすると、フォールバック (fallback) 文字 *c* を文字列 *anything* として定義します。
- .fcolor** 塗りつぶし色を、以前の塗りつぶし色に設定します。
- .fcolor c**  
塗りつぶし色を *c* に設定します。
- .fi** 出力行を満たします。
- .fl** 出力バッファをフラッシュします。
- .fp n font**  
位置 *n* で *font* を埋め込みます。
- .fp n internal external**  
位置 *n* で長い *external* 名を短い *internal* 名でフォントを埋め込みます。
- .fschar f c anything**  
文字列 *anything* としてフォント *f* のためにフォールバック文字 *c* を定義します。
- .fspecial font**  
*font* が空となるように、特別なフォントのリストをリセットします。

**.fspecial** *font s1 s2 ...*

現在のフォントが *font* であるとき、フォント *s1*, *s2*, ... は、特別となります。

**.ft** 前のフォントに戻ります。**VF** または **VP** と同じです。

**.ft font** フォントの名前または番号 *font* に変更します。**VF[font]** エスケープシーケンスと同じです。

**.ftr font1 font2**

*font1* を *font2* に変換します。

**.gcolor** グリフ色を前のグリフ色に設定します。

**.gcolor c**

グリフ色を *c* に設定します。

**.hc** 追加のハイフネーション指定子文字を削除します。

**.hc c** 追加のハイフネーション指定子文字 *c* を設定します。

**.hcode c1 code1 c2 code2 ...**

文字 *c1* のハイフネーションコードを *code1* に設定し、*c2* を *code2* に、等、設定します。

**.hla lang**

現在のハイフネーション言語を *lang* に設定します。

**.hlm n** 連続してハイフンが付けられた行の最大の数 *n* に設定します。

**.hpf file** *file* からハイフネーションのパターンを読み込みます。

**.hpfa file**

*file* からハイフネーションのパターンを追加します。

**.hpfc code file**

**.hpf** のために入力マッピングを設定します。

**.hw words**

例外的なハイフネーションがある *words* のリスト。

**.hy N** ハイフネーションモード *N* に切り換えます。

**.hym n** ハイフネーションのマージンを *n* に設定します (デフォルトの単位指定子は、**m** です)。

**.hys n** ハイフネーションの空白を *n* に設定します。

**.ie cond anything**

*cond* であるなら、*anything* であり、そうでなければ、**.el** に行きます。

**.if cond anything**

*cond* であるなら、*anything* です。そうでなければ、何も行いません。

**.ig** .. に遭遇するまで、テキストを無視します。

**.ig end** **.end** までテキストを無視します。

**.in** 前のインデント値に変更します。

- .in**  $\pm N$   $\pm N$  にしたがってインデントを変更します (デフォルトの単位指定子は、**m** です)。
- .it**  $N$  *trap*  
次の  $N$  行のための入力行のカウントトラップを設定します。
- .itc**  $N$  *trap*  
**.it** と同じですが、1 行として **\c** で割り込まれたカウント行です。
- .kern** ペアワイズ (pairwise; 対の) のカーニングを有効にします。
- .kern**  $n$   $n \neq 0$  であるなら、ペアワイズのカーニングを無効にし、そうでなければ、それを有効にします。
- .lc** リーダ (leader; 先導する) 繰り返し文字を削除します。
- .lc**  $c$  リーダ繰り返し文字を  $c$  に設定します。
- .length** *register anything*  
*register* の文字列 *anything* の長さに書き込みます。
- .linetabs**  
行タブモードを有効にします (すなわち、出力行と関連するタブ位置を計算します)。
- .linetabs**  $n$   
 $n \neq 0$  であるなら、行タブモードを無効にし、そうでなければ、それを有効にします。
- .lf**  $N$  入力の行番号を  $N$  に設定します。
- .lf**  $N$  *file*  
入力の行番号を  $N$  に設定し、ファイル名を *file* に設定します。
- .lg**  $N$   $N > 0$  であるなら、合字 (ligature) モードをオンにします。
- .ll** 前の行の長さに変更します。
- .ll**  $\pm N$   $\pm N$  にしたがって行の長さを設定します (デフォルトのサイズは、**6.5i** で、デフォルトの単位指定子は、**m** です)。
- .ls** 追加の行内のスキップの前の値に変更します。
- .ls**  $N$  追加の行内のスキップ値を  $N$  に設定します、すなわち、 $N-1$  の空白行は、各テキスト出力行の後に挿入されます。
- .lt**  $\pm N$  タイトルの長さ (デフォルトの単位指定子は、**m** です)。
- .mc** マージン文字をオフにします。
- .mc**  $c$  右のマージンからの実際の距離で各テキスト行の後に文字  $c$  を印刷 (表示) します。
- .mc**  $c$   $N$  マージン文字を  $c$  に設定し、右のマージンからの距離を  $N$  に設定します (デフォルトの単位指定子は、**m** です)。
- .mk** *register*  
*register* の現在の垂直位置をマークします。
- .mso** *file* *file* が tmac ディレクトリで検索されることを除いて、.so 要求と同じです。
- .na** 出力行の調整を行いません。

- .ne** 1 行の垂直の空白を必要とします。
- .ne N** *N* の垂直空白を必要とします (デフォルトで、単位指定子は、**v** です)。
- .nf** 出力行を満たすことか、または調整を行いません。
- .nh** ハイフネーションを行いません。
- .nm** 数値モードをオフにします。
- .nm ±N [M [S [I]]]**  
行番号モードで、数値、複数、間隔とインデントを設定します。
- .nn** 次の行に番号付けを行いません。
- .nn N** 次の *N* 行に番号付けを行いません。
- .nop anything**  
常に、*anything* を実行します。
- .nr register ±N M**  
自動的な増加量 *M* で ±*N* を使用して *register* を定義するか、または修正します。
- .nroff** 組み込みの条件 **n** を真にし、**t** を偽にします。
- .ns** 空白なしモードを音オンに切り替えます。
- .nx** 直ちに、現在のファイルの終わりにジャンプします。
- .nx filename**  
次のファイル。
- .open stream filename**  
書き込みのために レジスタ **filename** をオープンし、レジスタ **stream** で指定されたストリームとそれを関連付けます。
- .opena stream filename**  
**.open** に似ていますが、それに追加します。
- .os sv** 要求によって保存された垂直距離を出力します。
- .output string**  
*string* が (取り除かれる) " で始まるなら、先導する空白類を許可して、中間の出力に *string* を直接出力します。
- .pc** ページ番号の文字を '**%**' にリセットします。
- .pc c** ページ番号文字。
- .pi program**  
*program* への出力をパイプに送ります (nroff のみ)。
- .pl** ページの長さをデフォルトの 11**i** 設定します。現在のページの長さは、レジスタ **.p** に格納されます。
- .pl ±N** ページの長さを ±*N* に変更します (デフォルトの単位指定子は、**v** です)。
- .pm** マクロの名前とサイズを印刷 (表示) します (128 バイトのブロック数)。

- .pm t**      マクロのサイズの合計だけを印刷 (表示) します (128 バイトのブロック数)。
- .pn ±N**    次のページ番号 *N*。
- .pnr**      すべての現在定義された数値レジスタの名前と内容を標準エラーに印刷 (表示) します。
- .po**        前のページのオフセットに変更します。現在のページのオフセットは、レジスタ **.o** で利用可能です。
- .po ±N**    ページのオフセット *N*。
- .ps**        前のポイントサイズに戻ります。
- .ps ±N**    ポイントサイズ。 **\s[±N]** と同じです。
- .psbb filename**  
PostScript イメージ *filename* の境界ボックスを取得します。
- .pso command**  
この振る舞いは、入力 *command* の標準出力から来ることを除いて、**so** 要求に似ています。
- .ptr**      標準エラーで、(入力行のトラップと転換 (diversion) トラップを含まない) すべてのトラップの名前と位置を印刷 (表示) します。
- .pvs**      前の後方の垂直行 (post-vertical line) の間隔を変更します。
- .pvs ±N**   *±N* にしたがって後方の垂直行 (post-vertical line) の間隔を変更します (デフォルトの単位指定子は、**p** です)。
- .rchar c1 c2...**  
文字 *c1, c2, ...* の定義を削除します。
- .rd prompt**  
挿入を読み込みます。
- .return**   マクロから返ります。
- .return anything**  
2 度返ります、すなわち、現在のレベルのマクロから返り、マクロの 1 つ高いレベルから返ります。
- .rfschar fc1 c2...**  
フォント *f* のための文字 *c1, c2, ...* の定義を削除します。
- .rj n**      次の *n* 入力行を右寄せします。
- .rm name**  
要求、マクロまたは文字列 *name* を削除します。
- .rn old new**  
要求、マクロまたは文字列 *old* から *new* に名前を変更します。

- .rnn** *reg1 reg2*  
レジスタ *reg1* から *reg2* に名前を変更します。
- .rr** *register*  
*register* を削除します。
- .rs**  
間隔を復元します。空白なしモードをオフに切り替えます。
- .rt**  $\pm N$  マークされた垂直位置に (上方向のみ) 戻ります (デフォルトの単位指定子は、**v** です)。
- .schar** *c anything*  
文字列 *anything* のようにグローバルなフォールバック文字 *c* を定義します。
- .shc**  
ソフトハイフン文字を **\hy** にリセットします。
- .shc** *c* ソフトハイフン文字を *c* に設定します。
- .shift** *n*  
マクロで、*n* 位置によって引数をシフトします。
- .sizes** *s1 s2 ... sn [0]*  
利用可能なフォントサイズを **DESC** ファイルの **sizes** コマンドと同様に設定します。
- .so** *filename*  
ソースファイルをインクルードします。
- .sp**  
1 つの行を垂直にスキップします。
- .sp** *N* *N* の符号にしたがって垂直の距離 *N* を上または下に間隔をあけます (デフォルトの単位指定子は、**v** です)。
- .special**  
空となるように特別なフォントのグローバルなリストをリセットします。
- .special** *s1 s2 ...*  
フォント *s1*, *s2*, などは、特別で、現在のフォントでない文字を検索します。
- .spreadwarn**  
その値を変更せずに広がり警告をオンとオフに切り替えます。
- .spreadwarn** *limit*  
出力行の各空白が *limit* 以上に広げられるなら、警告を発行します (デフォルトの単位指定子は、**m** です)。
- .ss** *N* 空白文字のサイズは、現在のフォントの空白の幅の *N*/12 に設定しました。
- .ss** *N M* 空白文字のサイズは、*N*/12 に設定され、文の空白のサイズは、現在のフォントの空白の幅の *M*/12 に設定されます (=1/3 em)。
- .sty** *n style*  
フォント位置 *n* で *style* に結び付けます。
- .substring** *xx n1 n2*  
*xx* で指定された文字列をインデックス *n1* と *n2* によって定義された部分文字列で置き換えます。



- .sv** 垂直の空白の **1v** を保存します。
- .sv N os** 要求で後の出力のために垂直の距離 *N* を保存します。
- .sy command-line**  
プログラム *command-line* を実行します。
- .ta T N** セットは、*N* の倍数であるすべての位置の後にタブを設定します (デフォルトの単位指定子は、**m** です)。
- .ta n1 n2 ... nn T r1 r2 ... rn**  
位置 *n1*, *n2*, ..., *nn* でタブを設定し、次に *nn+r1*, *nn+r2*, ..., *nn+rn* でタブを設定し、次に *nn+rn+r1*, *nn+rn+r2*, ..., *nn+rn+rn* などのように設定します。
- .tc** タブ繰り返し文字を削除します。
- .tc c** タブ繰り返し文字を *c* に設定します。
- .ti ±N** 次の行を一時的にインデント (段付け) します (デフォルトの位指定子は、**m** です)。
- .tkf font s1 n1 s2 n2**  
*font* のためのトラックカーニングを有効にします。
- .tl 'left' center' right'**  
3つの部分のタイトル。
- .tm anything**  
端末で *anything* (何でも) 印刷 (表示) します (UNIX 標準のメッセージ出力)。
- .tm1 anything**  
端末で *anything* (何でも) 印刷 (表示) します (UNIX 標準のメッセージ出力)、*anything* (何か) が (取り除かれる) " で始まるなら、先導する空白類を許可します。
- .tmc anything**  
最終の改行を出力しない **.tm1** と同様です。
- .tr abcd...**  
出力で、*a* から *b* に、*c* から *d* に変換します。
- .trf filename**  
ファイル *filename* の内容を透過的に出力します。
- .trin abcd...**  
これは、文字変換の前に **asciify** 要求が (もしあるなら) 文字コードを使用するを除いて、**tr** 要求と同じです。
- .trnt abcd...**  
変換が、**\!** でディバージョン (diversion) への透過的なスループットであるテキストに適用しないことを除いて、これは、**tr** 要求と同じです。
- .troff** 組み込み条件式 **t** を真に **n** を偽にします。
- .uf font** 下線フォントを *font* に設定します (**.ul** によって切り換えられます)。

- .ul** *N*     *N* 入力行を (troff でイタリック体の) 下線を引きます。
- .unformat** *diversion*  
      *diversion* のフォント情報を保存して、空白文字とタブを書式化解除 (unformat) します。
- .vpt** *n*     *n* が 0 でなければ、垂直位置のトラップを有効にし、そうでなければ、それらを無効にします。
- .vs**        前の垂直の基準行の間隔に変更します。
- .vs**  $\pm N$     $\pm N$  にしたがって垂直の基準行の間隔を設定します (デフォルトの単位指定子は、**p** です)。デフォルト値は、12**p** です。
- .warn** *n*    警告コードを *n* に設定します。
- .warnscale** *si*  
      警告で使用する単位指定子を *si* に設定します。
- .wh** *N*     位置 *N* で (最初の) トラップを削除します。
- .wh** *N trap*  
      位置トラップを設定します。負は、ページの底からを意味します。
- .while** *cond anything*  
      条件式 *cond* が真の間に、入力として *anything* (何でも) 受け付けます。
- .write** *stream anything*  
      *anything* (何でも) *stream* で指定されたストリームに書き込みます。
- .writec** *stream anything*  
      最終の改行を出力しない **.write** と同様です。
- .writem** *stream xx*  
      マクロまたは文字列 *xx* の内容を *stream* で指定されたストリームに書き込みます。

これらの標準の groff 要求に加えて、さらなるマクロの呼び出しがあるかもしれません。それらは、マクロのパッケージ (要約については、**roff**(7) を参照) またはプリプロセッサに由来するかもしれません。

プリプロセッサのマクロを認識することは容易です。それらは、それらのコードをペアの特有のマクロで囲みます。

プリプロセッサ	開始マクロ	終了マクロ
<b>eqn</b>	<b>.PS</b>	<b>.PE</b>
<b>grap</b>	<b>.G1</b>	<b>.G2</b>
<b>grn</b>	<b>.GS</b>	<b>.GE</b>
<b>pic</b>	<b>.PS</b>	<b>.PE</b>
<b>refer</b>	<b>.R1</b>	<b>.R2</b>
<b>soelim</b>	なし	なし
<b>tbl</b>	<b>.TS</b>	<b>.TE</b>

## エスケープシーケンス

エスケープシーケンスは、通常、バックスラッシュ ‘\’ によって導入されたインラン言語要素であり、エスケープ名と、ときどき必要な引数が続きます。入力処理は、介在する区切り文字なしで、エスケープされた文字または引数 *resp.* の直後に続けられます。それで、エスケープ名と終わりと引数の終わりを決定するための方法がなければなりません。

これは、角括弧のペア *[name]* (変数名から成るエスケープ名と引数) で名前を囲み、‘*constant*’ のようなアポストロフィ (ASCII 0x27) によって (数値表現と文字) 定数引数を囲むことによって行われます。

短い名前のための省略形があります。閉じ括弧なしで *\xy* のような開き括弧によって、2つの文字エスケープ名を指定することができます。そして、形式 *\c* のマーカなしで、特殊文字 ‘*[*’ と ‘*(*’ と異なるすべての 1 文字名を指定することさえできます。

長さ 1 の定数引数は、マーカのアポストロフィも省略することができますが、2つの文字に類似点は、ありません。

1 文字のエスケープシーケンスがインライン関数とシステム関連のタスクのために主に使用される間に、*\* 構成が続いている 2 文字の名前は、roff システムによってあらかじめ定義された特殊文字のために使用されます。2 を超える名前があるエスケープシーケンス *\[name]* は、ユーザ定義の名前が付けられた文字を示します (**char** 要求を参照)。

### 単一の文字のエスケープ

- \"* コメントの開始。行の終わりまでのすべてが無視されます。
- \#* 次の改行を含んですべてが、無視されます。これは、コピーモードで解釈されます。これは、終端の改行が同様に無視されることを除いて、*\"* に似ています。
- \\*s* 1 文字の名前 *s* がある文字列変数に格納される文字列。
- \\*(st* 2 文字の名前 *st* がある文字列変数に格納される文字列。
- \\*[stringvar arg1 arg2 ...]*  
文字列は、引数として *arg1, arg2, ...* を取り、任意の長さの名前 *stringvar* で文字列変数に格納された文字列。

- \\$0** 現在のマクロが呼び出された名前。**als** 要求は、1 つ以上の名前があるマクロを作成することができます。
- \\$x** 1 つ場所の数値  $x$  があるマクロまたは文字列引数、ここで、 $x$  は、1 から 9 までの数値です。
- \\$(xy** 2 桁の数値  $xy$  があるマクロまたは文字列引数。
- \\$[nexp]**  
数値  $nexp$  があるマクロまたは文字列引数、ここで、 $nexp$  は、整数  $\geq 1$  に評価している数値式です。
- \\$\*** マクロまたは文字列において、空白によって区切られたすべての引数の連結。
- \\$@** マクロまたは文字列において、ダブルクォートによってそれぞれ囲まれ、空白によって区切られる、すべての引数の連結。
- \** 単一のバックスラッシュになりますコピーモードのエスケープ文字としてその解釈を遅延するために役に立ちます。印刷可能なバックスラッシュのために、現在のエスケープ文字から独立している **\e** またはずっと素晴らしい **\rs** を使用します。
- \'** 揚音アクセント符号 **\'** (**\aa** と同じです。エスケープされない: アポストロフィ、右の引用符、シングルクォート (ASCII 0x27)。
- \`** グレーブアクセント **\`** (**\ga** と同じ。エスケープされない: 左クォート、バッククォート (ASCII 0x60)。
- \-** 現在のフォントの **-** 符号。
- \.** 行の開始のでさえ、解釈されないドット (ピリオド)。
- \%** デフォルトのオプションのハイフネーション文字。
- \!** 透過的な行指定子。
- \?anything?**  
ディバージョンで、これは、ディバージョンの *anything* (何でも) 透過的に組み込みます。*anything* (何でも) コピーモードで読み込まれます。エスケープシーケンス **\!** と **\?** も参照してください。
- \space** パディングされない、空白サイズの空白文字 (改行なし)。
- \0** 数値の幅。
- \|** 1/6 em の狭い空白文字。nroff で 0 の幅。
- \^** 1/12 em 0.5 半分の狭い空白文字。nroff で 0 の幅。
- \&** 印刷可能でない、0 の幅の文字。
- \)** 文の認識の終わりの目的のために透過的である **cflags** 要求で宣言される文字のように振る舞うことを除いて、**\&** に似ています。
- \/** 次の文字がロマン体の文字であるなら、その文字と次の文字の間隔が修正することができますように、先行する文字の幅を増加します。

- \,** 先行する文字がロマン体の文字であるなら、その文字と先行する文字の間隔を訂正することができるように、次の文字の間隔を修正します。
- \~** 行が調整されるとき、通常の相互の単語空白のように引き延ばされる改行されない空白。
- \:** 0 の幅のブレイクポイントを挿入します (**\%** に似ていますが、ソフトハイフン文字なしです)。
- \newline**  
行を連続するための、無視される改行。
- \{** 条件入力を開始します。
- \}** 条件入力を終了します。
- \(sc** 2 文字の名前 *sc* がある特殊文字、セクション **特殊文字** を参照してください。
- \[name]**  
任意の長さの名前 *name* がある名前付き文字 (というより、グリフ文字)。
- \[comp1 comp2 ...]**  
構成要素 *comp1, comp2, ...* がある複合のグリフ文字。
- \a** 解釈されないリーダ文字。
- \A'anything'**  
*anything* (何でも) が、文字列、マクロ、ディバージョン、レジスタ、環境、またはフォントの名前として受け付け可能であるなら、1 に展開され、そうでなければ、0 に展開します。
- \b'abc...'**  
角括弧を構築している関数。
- \B'anything'**  
*anything* (何でも) が、有効な数値表現として受け付け可能であるなら、1 に展開し、そうでなければ、0 に展開します。
- \c** テキスト処理を割り込みます。
- \C'char'**  
*char* と呼ばれる文字。 **\[char]** と同じですが、他の roff バージョンと互換性があります。
- \d** 1/2 em 垂直単位前方 (下方) に進みます (nroff の 1/2 行)。
- \D'charseq'**  
*charseq* の文字によって定義されたグラフィカルな要素を描画します。詳細については、groff 情報ファイルを参照してください。
- \e** 現在のエスケープ文字の印刷可能なバージョン。
- \E** エスケープ文字と同等ですが、コピーモードで解釈されません。
- \fF** 1 文字の名前または 1 桁の数値 *F* があるフォントに変更します。
- \fP** 以前のフォントに切り替えて戻します。
- \f(fo** 2 文字の名前または 2 桁の数値 *fo* があるフォントに変更します。

**\f** [*font*]

任意の長さの名前または数表現 *font* があるフォントに変更します。

**\f** [] 以前のフォントに切り替えて戻します。

**\F** *f* 1 文字の名前 *f* があるフォントファミリに変更します。

**\F** (*fm* 2 文字の名前 *fm* があるフォントファミリに変更します。

**\F** [*fam*]

任意の長さの名前 *fam* があるフォントファミリに変更します。

**\F** [] 以前のフォントファミリに切り替えて戻します。

**\g** [*reg*]

**.af** に適切な名前 *reg* があるレジスタの形式を返します。形式 **\g**(*xy* と **\gx** を代わりに使用します。

**\h** '*N*' ローカルな水平の動作。右に *N* 移動します (負であるなら左に移動します)。

**\H** '*N*' 現在のフォントの高さを *N* に設定します。

**\k** [*reg*]

任意の長さの名前 *reg* があるレジスタの水平の入力位置をマークします。形式 **\k**(*xy* と **\kx** を代わりに使用します。

**\l** '*Nc*'

水平の行を描画する関数 (オプションで、文字 *c* を使用します)。

**\L** '*Nc*'

垂直の行を描画する関数 (オプションで、文字 *c* を使用します)。

**\m** [*color*]

色 *color* に変更します。形式 **\m**(*co* と **\mc** を代わりに使用します。

**\m** [] 以前の色に切り替えて戻します。

**\M** [*color*]

閉じた描画のオブジェクトのための塗りつぶし色を色 *color* に変更します。形式 **\M**(*co* と **\Mc** を代わりに使用します。

**\M** [] 以前の塗りつぶし色に切り替えて戻します。

**\nr** 1 文字の名前 *r* があるレジスタ変数に格納される数値。

**\n** (*re* 2 文字の名前 *re* があるレジスタ変数に格納される数値。

**\n** [*reg*]

任意の長さの名前 *reg* があるレジスタ変数に格納される数値。

**\N** '*n*' 現在のフォントのコード *n* がある文字をタイプセットします、特別なフォントは、検索されません。**char** 要求を使用して文字をフォントに追加するために役に立ちます。

**\o** '*abc...*'

重ね打ちする文字 *a*, *b*, *c* など。

- \o0**    グリフ文字出力を無効にします。主に内部の使用のためです。
- \o1**    グリフ文字出力を有効にします。主に内部の使用のためです。
- \p**    改行し、出力行を引き延ばします。
- \r**    1 em 垂直動作を反転します (nroff で逆の行)。
- \R'name ±n'**  
       **.nr name ±n** と同じです。
- \s [±N]**  
       ポイントサイズを  $N$  スケールポイントに設定します。形式 **\s[±N]**, **\s'±N'**, **\s±'N'**, **\s(±xy,**  
**\s±(xy, \s±x** を代わりに使用することに注意してください。ps 要求と同じです。
- \s'N'**    出力を  $N$  度傾けます。
- \t**    解釈されない水平タブ。
- \u**    1/2 em 垂直動作を反転 (上方) します (nroff で 1/2 行)。
- \v'N'**    ローカルな垂直の動作。  $N$  下方に移動します (負であるなら、上方に移動します)。
- \V[env]**  
       環境変数 *env* の内容。形式 **\V(xy** と **\Vx** を代わりに使用します。
- \w'string'**  
       文字シーケンス *string* の幅。
- \x'N'**    特別な行間隔関数 (負であるなら前に、正であるなら後に)。
- \X'string'**  
       デバイス制御関数として *string* を出力します。
- \Y[name]**  
       デバイス制御関数として解釈されていない文字列変数またはマクロ *name* を出力します。  
       形式 **\Y(xy** と **\Yx** を代わりに使用します。
- \zc**    0 の幅 (間隔なし) で *c* を印刷 (表示) します
- \Z'anything'**  
       *anything* (何でも) 印刷 (表示) し、次に、水平位置と垂直位置を復元します。*anything* (何でも) は、タブまたはリーダを含めません。

エスケープシーケンス **\e**, **\l**, **\'**, **\\$**, **\\***, **\a**, **\n**, **\t**, **\g**, と **\newline** は、コピーモードで解釈されます。

**\(** または **\[** で始まるエスケープシーケンスは、単一の文字のエスケープシーケンスを表していませんが、2 つ以上の文字でエスケープ名を持ち込みます。

バックスラッシュに、定義されたエスケープシーケンスを構成しない文字が続いているなら、バックスラッシュは、静かに無視され、文字は、それ自体にマップします。

## 特殊文字

共通の特殊文字は、文字  $x$  と  $y$  で形式 **\(xy** のエスケープシーケンスによってあらかじめ定義されます。これらのいくつかは、それらのほとんどが特別なフォントでのみ利用可能な間に、通常の

フォントに存在します。以下で、利用者は、最も重要なグリフ文字の選択を見つけます。完全なリストは、**groff\_char(7)**で見つけることができます。

**\(bu** 中黒符号。  
**\(co** 著作権 (Copyright)。  
**\(ct** セント。  
**\(dd** ダブルダガー (2 重短剣符)。  
**\(de** 度。  
**\(dg** ダガー (短剣符)。  
**\(rs** 印刷可能なダブルクォート。  
**\(em** 全角ダッシュ。  
**\(hy** ハイフン。  
**\(rg** 登録記号。  
**\(rs** 印刷可能なバックスラッシュ文字。  
**\(sc** セクション記号。  
**\(ul** 下線文字。  
**\(==** 等しい。  
**\(>=** 以上。  
**\(<=** 以下。  
**\(!=** 等しくない。  
**\(->** 右矢印。  
**\(<-** 左矢印。  
**\(+-** プラスマイナス記号。

## 文字列

文字列は、**ds** 要求によって定義され、**\\*** エスケープシーケンスによって取り出すことができます。

文字列は、それらの名前空間とマクロを共有します。それで、文字列と引数なしのマクロは、おおよそ同等です。マクロのような文字列を呼び出すことは、逆もまた同様に可能ですが、これは、しばしば予測不可能な結果を導きます。次の文字列は、**groff** であらかじめ定義されています。

**\\*[.T]**      **-T** コマンド行オプションによって指定されるような現在の出力デバイスの名前。

## レジスタ

レジスタは、値を格納する変数です。**groff** では、ほとんどのレジスタは、数値 (上記のセクション **数式** を参照) を格納しますが、いくつかは文字列値も保持することができます。

各レジスタは、名前を与えられます。任意のレジスタは、要求 **nr register** (レジスタ) で定義して設定することができます。



`\n` によって持ち込まれたエスケープシーケンスによってレジスタに格納された値を取り出すことができます。

最も役に立つことは、あらかじめ定義されたレジスタです。次の表記で、*name* は、レジスタに関して話すことを明らかにするためにレジスタ **name** を呼び出すレジスタを参照するために使用されます。`\n[]` 修飾は、レジスタ名の一部ではないことを心に留めてください。

### 読み込み専用レジスタ

次のレジスタは、ユーザによって修正されるべきではない値をあらかじめ定義しています (通常、読み込み専用のドットで始まるレジスタ)。たいてい、それらは、要求呼び出しから現在の設定または格納の結果についての情報を提供しています。

- `\n[. $]`      現在のマクロまたは文字列の引数の数。
- `\n[. a]`      `\x'N'` を使用して最も最近利用するポスト行 (post-line) の特別の行の間隔。
- `\n[. A]`      オプション `-A` が使用されるなら、**troff** で 1 に設定します。常に、**nroff** で 1 です。
- `\n[. c]`      現在の入力行番号。
- `\n[. C]`      互換モードが有効であるなら、1、そうでなければ、0 です。
- `\n[. cdp]`    現在の環境に追加された最後の文字の深さ。文字が、基準行の下に引き延ばされるなら、それは、正です。
- `\n[. ce]`      **ce** 要求によって設定されたように、センタリングされる残りの行数。
- `\n[. cht]`    現在の環境に追加された最後の文字の高さ。文字が、基準行の上に引き延ばされるなら、それは、正です。
- `\n[. color]`  
色の有効にされるなら、1、そうでなければ、0 です。
- `\n[. csk]`    現在の環境に追加された最後の文字の傾き。文字の傾きは、文字が置かれるべきアクセントの中央が文字の中央の右にどのくらい離れているかです。
- `\n[. d]`      現在のディバージョンの現在の垂直の位置。レジスタ **nl** に同等です。
- `\n[. ev]`      現在の環境 (文字列の値) の名前または数値。
- `\n[. f]`      現在のフォント番号。
- `\n[. fam]`    現在のフォントファミリー (文字列の値)。
- `\n[. fn]`      現在の (内部の) 実際のフォント名 (文字列の値)。
- `\n[. fp]`      次の空いているフォント位置の番号。
- `\n[. g]`      常に、GNU **troff** で 1 です。マクロは、**groff** の下で実行しているなら、テストするために、それを使用すべきです。
- `\n[. h]`      現在のページまたはディバージョンでテキストの基準行の最高水位線。
- `\n[. H]`      基本的な単位で利用可能な水平の解像度。

`\n[.height]`

`\H` で設定されるような現在のフォントの高さ。

`\n[.hla]` `.hla` 要求によって設定されるような現在のハイフネーション言語。

`\n[.hlc]` 直前の連続してハイフネーションされた行の数。

`\n[.hlm]` `hlm` 要求によって設定されるような連続してハイフネーションされた行の最大の許可された数。

`\n[.hy]` (`hy` 要求によって設定されたような) 現在のハイフネーションのフラグ。

`\n[.hym]` (`hym` 要求によって設定されたような) 現在のハイフネーションのマージン。

`\n[.hys]` (`hys` 要求によって設定されたような) 現在のハイフネーションの空白。

`\n[.i]` 現在のインデント。

`\n[.in]` 現在の出力行に適用するインデント。

`\n[.int]` 最後の出力行が `\c` を含むなら、正です。

`\n[.kern]` ペアのカーニングが有効であるなら、1、そうでなければ、0 です。

`\n[.l]` 現在の行の長さ。

`\n[.lg]` (`lg` 要求によって設定されたような) 現在の合字モード。

`\n[.linetabs]`

(`linetabs` 要求によって設定されたような) 現在の行タブモード。

`\n[.ll]` 現在の出力行に適用する行の長さ。

`\n[.lt]` (`lt` 要求によって設定されたような) タイトルの長さ。

`\n[.m]` 現在の描画色 (文字列の値)。

`\n[.M]` 現在の背景色 (文字列の値)。

`\n[.n]` 前の出力行のテキスト部分の長さ。

`\n[.ne]` 飛躍するトラップを引き起こす、最後の `ne` 要求で必要とされた空白の量。レジスタ `.trunc` とともに役に立ちます。

`\n[.ns]` 1 空白なしモードで、1、そうでなければ、0 です。

`\n[.o]` 現在のページのオフセット。

`\n[.p]` 現在のページの長さ。

`\n[.pe]` ページの排出の間は、1、そうでなければ、0 です。

`\n[.pn]` 次のページの番号: `pn` 要求によって設定された値、または現在のページに 1 足した数のいずれか。

`\n[.ps]` スケールポイントの現在ポイントサイズ。

`\n[.psr]` スケールポイントの最後に要求されたポイントサイズ。

`\n[.pvs]` 現在のポスト垂直の行の間隔。

`\n[.rj]` `rj` 要求によって設定されるような右寄せされる行の数。

`\n[.s]` 10 進小数としての現在のポイントサイズ。

`\n[.slant]`

`\S` で設定されるような現在のフォントの傾き。

`\n[.sr]` 10 進小数としてポイント単位の最後に要求されたポイントサイズ (文字列の値)。

`\n[.ss]` `ss` 要求の最初の引数によって設定されたパラメータの値。

`\n[.sss]` `ss` 要求の 2 番目の引数によって設定されたパラメータの値。

`\n[.sty]` 現在のフォントスタイル (文字列の値)。

`\n[.t]` 次のトラップまでの距離。

`\n[.T]` オプション `-T` が使用されるなら、1 に設定します。

`\n[.tabs]` `ta` 要求のための引数として使用に適切な現在のタブ設定の文字列表現。

`\n[.trunc]`

最も最近生じる垂直位置のトラップによって切り詰められた垂直の空白の量、またはトラップが `ne` 要求によって生じるなら、`.ne` によって生成された垂直の動作の量を引いたもの。言い換えれば、トラップが生じるポイントで、トラップが生じなかった垂直位置と実際の垂直位置の差を表しています。レジスタ `.ne` とともに役に立ちます。

`\n[.u]` `fill` モードの 1 と `nofill` モードの 0 に等しくなります。

`\n[.U]` 安全な (`safer`) モードの 1 と安全でない (`unsafe`) モードの 0 に等しくなります。

`\n[.v]` 現在の垂直行の間隔。

`\n[.V]` 基本的な単位の利用可能な垂直の解像度。

`\n[.vpt]` 垂直位置トラップが有効であるなら、1、そうでなければ、0 です。

`\n[.w]` 前の文字の幅。

`\n[.warn]` 現在有効にされた警告の数値コードの合計。

`\n[.x]` メジャーバージョン番号。

`\n[.y]` マイナバージョン番号。

`\n[.Y]` `groff` のリビジョン番号。

`\n[.z]` 現在のディバージョンの名前。

### 書き込み可能なレジスタ

次のレジスタは、ユーザによって読み込み、書き込むことができます。それらは、デフォルト値をあらかじめ定義されますが、これらは、文書をカスタマイズするために修正することができます。

`\n[%]` 現在のページ番号。

`\n[c.]` 現在の入力行番号。

`\n[ct]` 文字タイプ (幅関数 `\w` によって設定します)。

`\n[dl]` 最後に完了したディバージョンの最大の幅。

`\n[dn]` 最後に完了したディバージョンの高さ。

`\n[dw]` 現在の週の日 (1-7)。

- `\n[dy]` 現在の月の日 (1-31)。
- `\n[hours]` 真夜中過ぎの時刻。起動時に初期化されます。
- `\n[hp]` 入力行の現在の水平位置。
- `\n[llx]` 与えられた PostScript の (PostScript 単位の) イメージの左下の x 座標 (`.psbb` によって設定されます)。
- `\n[lly]` 与えられた PostScript の (PostScript 単位の) イメージの左下の y 座標 (`.psbb` によって設定されます)。
- `\n[ln]` 出力行番号。
- `\n[minutes]`  
時間の後の分数。起動時に初期化されます。
- `\n[mo]` 現在の月 (1-12)。
- `\n[nl]` 最後に印刷 (表示) された基準行の垂直位置。
- `\n[rsb]` レジスタ `sb` に似ていますが、文字の高さと深さを考慮します。
- `\n[rst]` レジスタ `st` に似ていますが、文字の高さと深さを考慮します。
- `\n[sb]` 基準行の下文字列の深さ (幅関数 `\w` によって生成されます)。
- `\n[seconds]`  
分の後の秒数。起動時に初期化されます。
- `\n[skw]` `\w` 引数の最後の文字のセンタからの右のスキップ幅。
- `\n[slimit]`  
0 より大きいなら、入力スタックのオブジェクトの最大数。≤0 であるなら、制限は、ありません、すなわち、仮想メモリが使い果たされるまで再帰が続くことができます。
- `\n[ssc]` 添字の前の最後の文字に追加されるべき (負で可能性がある) 水平の空白の量 (幅関数 `\w` によって生成されます)。
- `\n[st]` 基準行の上文字列の高さ (幅関数 `\w` によって生成されます)。
- `\n[systat]`  
最後の `sy` 要求によって実行された `system()` 関数の返り値。
- `\n[urx]` 与えられた PostScript の (PostScript 単位の) イメージの右上の x 座標 (`.psbb` によって設定されます)。
- `\n[ury]` 与えられた PostScript の (PostScript 単位の) イメージの右上の y 座標 (`.psbb` によって設定されます)。
- `\n[year]` 現在の年 (2000 年問題対応)。
- `\n[yr]` 現在の年マイナス 1900。2000 年問題対応のために、代わりにレジスタ `year` を使用します。

## 互換性

[CSTR #54] によって定義されるような伝統的な troff と比べた groff 言語の違いは、`groff_diff(7)` に文書化されています。

groff システムは、互換モードを提供しています、どのようにこれを呼び出すかに関して **groff(1)** を参照してください。

## バグ

[groff bug mailing list](mailto:bug-groff@gnu.org) [〈bug-groff@gnu.org〉](mailto:bug-groff@gnu.org) にバグを報告してください。バグを再現することができる完全で、自己完結型の使用例、利用者が使用している groff のバージョンを含めてください。

## 作者

Copyright (C) 2000, 2001, 2002, 2003, 2004, 2005 Free Software Foundation, Inc.

この文書は、FDL (GNU Free Documentation License) バージョン 1.1 以降の条件の下で配布されています。利用者は、システムの FDL のコピーを受け取るべきで、それは、また、[GNU copyleft site](http://www.gnu.org/copyleft/fdl.html) [〈http://www.gnu.org/copyleft/fdl.html〉](http://www.gnu.org/copyleft/fdl.html) でオンラインで利用可能です。

この文書は、*groff*, GNU roff 配布の一部です。それは、[Bernd Warken](mailto:bwarken@mayn.de) [〈bwarken@mayn.de〉](mailto:bwarken@mayn.de) によって書かれました。それは、[Werner Lemberg](mailto:w1@gnu.org) [〈w1@gnu.org〉](mailto:w1@gnu.org) によって保守されています。

## 関連項目

groff 言語のための主な情報のソースは、**groff info(1)** ファイルです。血みどろ (gory) の詳細に加えて、多くの例を含んでいます。

### **groff(1)**

groff プログラムの使用法と groff システムの文書と入手できるものへのポインタ。

### **groff\_diff(7)**

伝統的な roff と比べての groff 言語との違い。これは、groff に特有なあらかじめ定義された言語要素のための権威ある文書です。

### **groff\_char(7)**

あらかじめ定義された groff 文字 (グリフ文字)。

### **groff\_font(5)**

フォントと DESC ファイルの仕様。

**roff(7)** roff の歴史、すべての roff システムによって共有される共通の部分とさらなる文書へのポインタ。

### *[CSTR #54]*

[Nroff/Troff User's Manual by Osanna & Kernighan](http://cm.bell-labs.com/cm/cs/54.ps) [〈http://cm.bell-labs.com/cm/cs/54.ps〉](http://cm.bell-labs.com/cm/cs/54.ps) — 伝統的な troff のためのバイブル。