

Groff Cheatsheet

Groff (GNU troff) is a typesetting system that reads plain text mixed with formatting commands and produces formatted output. Output may be PostScript, PDF, html, or ASCII/UTF8 for display at the terminal. Formatting commands may be either low-level typesetting requests (“primitives”) or macros from a supplied set. Users may also write their own macros. All three may be combined.

Present on most Unix systems owing to its long association with Unix manuals (manpages), groff is capable of producing typographically sophisticated documents while consuming only minimal system resources.

Groff is released under the [GNU General Public License](#).

Compile

We need to compile our document into ps (PostScript) or pdf (Portable Document Format) formats which will be done by redirection:

1. ps format:

```
$ groff -ms file.ms > output.ps
$_
```

2. pdf format:

```
$ groff -ms -Tpdf file.ms > output.pdf
$_
```

- We also can compile with -mspdf flag to have some extra features like TOC:

```
$ groff -mspdf file.ms > output.pdf
$_
```

Comments

groff has three way to create a comment:

```
1  \" This is a comment
2  .\" Also a comment but, it has to be at the beginning of the line
3  .ig
4  This is a multiline comment
5  everything in this block will be ignored.
6  ..
```

Title

Every document (in general, not only in groff) at least needs a *title* and *author*:

```
1  .TL
2  The Title
```

```
3 .AU
4 The Author
```

We can add institution:

```
1 .AI
2 Home
```

We can also have abstraction:

```
1 .AB no
2 This is an abstraction
3 .AE
```

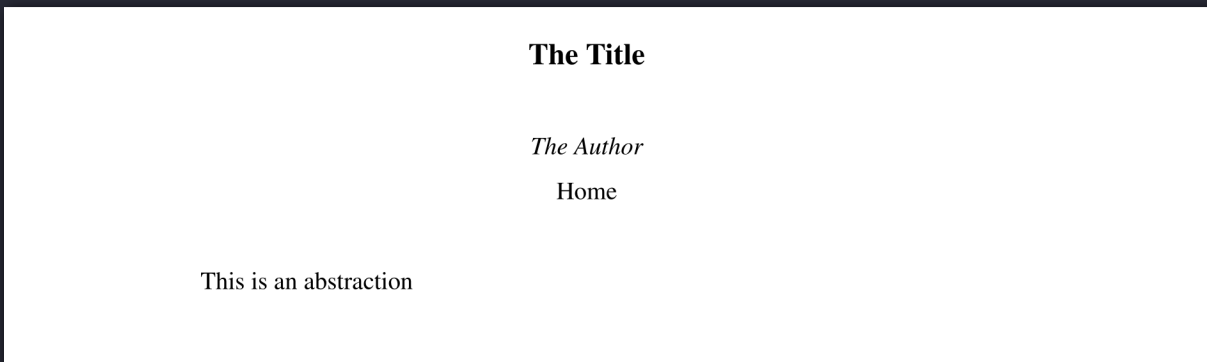


Figure 1: *Title, Author, Institution and Abstraction*

Page formatting

We can have page *footer* and *header*:

```
1 .ds RH Top right
2 .ds CH Top middle \" reserved for page number
3 .ds LH Top left
4 .ds RF Buttom right
5 .ds CF Buttom middle \" reserved for current date (.DA)
6 .ds LF Buttom left
```

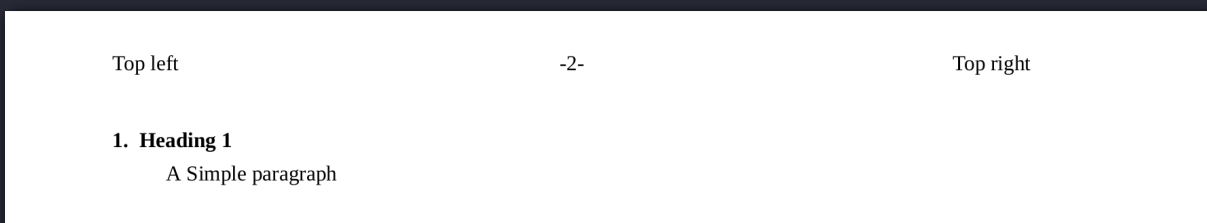


Figure 2: Page Headers

Change font size:

Figure 3: Page Footers

```
1 .nr PS 18
```

Text formatting

Headings

A groff document can have two types of headings:

- Ordered or Numbered
- Unordered

Numbered/Ordered headings

```
1 .NH 1
2 Heading level 1
3 .NH 2
4 Heading level 2
5 .NH 3
6 Heading level 3
7 .NH 4
8 Heading level 4
9 .NH 5
10 Heading level 5
11 .NH 6
12 Heading level 6
```

```
1. Heading level 1
1.1. Heading level 2
1.1.1. Heading level 3
1.1.1.1. Heading level 4
1.1.1.1.1. Heading level 5
1.1.1.1.1.1. Heading level 6
```

Figure 4: Numbered Headings

The `.NH` macro without any arguments (levels) is level 1 by default.

Unordered headings

```
1 .SH
2 A Simple heading
3 .SH 2
4 A Simple heading
```

A Simple heading

A Simple heading

Figure 5: Unordered Headings

Paragraphs

We have some options here:

No-line indented

```
1 .LP
2 This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.
3 This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.
```

Paragraphs

This is a simple paragraph.This is a simple paragraph.This is a simple paragraph. This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.

Figure 6: No-Line Indented Paragraph

First-line indented

```
1 .PP
2 This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.
3 This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.
```

Paragraphs

This is a simple paragraph.This is a simple paragraph.This is a simple paragraph. This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.

Figure 7: First-line Indented Paragraph

All-but-first-line indented

```
1 .XP
2 This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.
3 This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.
```

Paragraphs

This is a simple paragraph.This is a simple paragraph.This is a simple paragraph. This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.

Figure 8: All-But-First-Line Indented Paragraph

All-lines indented

```
1 .QP
2 This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.
3 This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.
```

Paragraphs

This is a simple paragraph.This is a simple paragraph.This is a simple paragraph. This is a simple paragraph.This is a simple paragraph.This is a simple paragraph.

Figure 9: All-Lines Indented

Macros

We can define a macro to stop repeating ourselves and save time, kinda like functions in every programming language and exactly like macros in C language.

```
1 .de MacroName
2 \ " Body
3 ..
```

And now we can use them just like other macros:

```
1 .MacroName
2 \ " body
```

Some useful macros

bullet list:

```
1 .de bl
2 .IP
3 \ (bu
4 ..
5 \ " usage
6 .bl
7 Test item one
8 .bl
9 Test item two
```

- Test item one
- Test item two

Figure 10: *bl* Bullet-list Macro

Boxed list

```
1 .de bb
2 .IP
3 \ [sq]
4 ..
5 \ " usage
6 .bb
7 TODO item one
8 .bb
9 TODO item two
```

- TODO item one
- TODO item two

Figure 11: *bl* Boxed-list Macro

Boxed code block:

```
1  .nr ln 1
2  .de cb
3  .QS
4  .B1
5  .ft CR
6  .sp 1n
7  .br
8  .sp 1n
9  .nf
10 .in +1m
11 .nm +0 1 1 -5
12 ..
13 .de /cb
14 .br
15 .sp 1n
16 .nm
17 .fi
18 .ft
19 .in
20 .B2
21 .QE
22 ..
23 \" usage
24 .cb
25 #include <stdio.h>
26 int main(int argc, char *argv[])
27 {
28     printf("Hello world!\\n");
29     return 0;
30 }
31 ./cb
```

```
1  #include <stdio.h>
2  int main(void)
3  {
4      printf("Hello world!\n");
5      return 0;
6  }
```

Figure 12: *cb* Boxed-codeblock Macro