

Design in the command line: Recipes for tasty outcomes

Manufatura Independente

Ana Isabel Carvalho & Ricardo Lafuente

Why?

- fast
- repeatable
- composable
- scriptable
- shareable

Great for graphic workflows

But hard to get into

Best practice: Document your recipes
(shell history is good enough if you're lazy)

Batch operations with ImageMagick

```
# resize all images in place  
mogrify -resize 25% *.png
```

```
# create png copies of all jpg files  
for f in *.jpg; do convert $f ${f/.jpg/.png}; done
```

```
# create a pdf with all the jpegs in this dir  
convert *.jpg out.pdf
```

exif

```
# remove all metadata from an image  
exiftool -all= -overwrite-original *.jpg
```

```
# add/replace author metadata  
exiftool *.jpg -overwrite-original -artist="Luther Blissett"
```


Scriptable gradient maps

[De la Piedra a la Nave](#)

Diseño colaborativo de
tipografía

[PREPARANDO](#)

[FOTOS DE LA SERRERÍA BELGA](#)

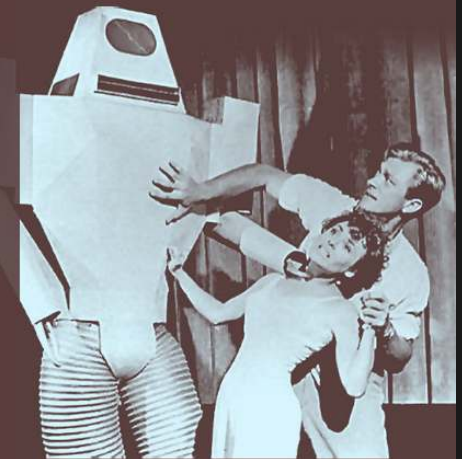
[REFERENCIAS](#)

[ACERCA DEL TALLER](#)

[FUENTES](#)

[SPECIMENS](#)

WORKSHOP BY [MANUFACTURA](#)
INDEPENDENTE AT [MEDIALAB-PRADO](#),
JULY 2012



animation with ffmpeg

Inkscape has a command line mode too

```
# convert SVG to PDF  
inkscape poster.svg --export-filename=poster.pdf
```

there's lots of options, run `inkscape --help`

pdftk

```
# merge many PDF files into one  
pdftk *.pdf cat output all.pdf
```

Booklet scripts with podfoimpose

```
# turn an A5 pdf into an A4 booklet ready for folding+stapli  
booklet-impose-a5 source.pdf
```

```
# get the total number of pages  
booklet-get-page-total source.pdf
```

```
# insert an A5 blank page at page 3  
booklet-insert-blank-page source.pdf a5 3
```

— JUST A WARNING, DON'T KILL LOCKDEP —

This CPU waits for a different one

*Return true if the calling CPU is allowed to print oops-related info.
This is a bit racy..*

*Called when the architecture enters its oops handler, before it prints
anything. If this is the first CPU to oops, and it's oopsing the first
time then let it proceed.*

*This is all enabled by the pause_on_oops kernel boot option. We do all
this to ensure that oopses don't scroll off the screen. It has the
side-effect of preventing later-oopsing CPUs from mucking up the display,
too.*

*It turns out that the CPU which is allowed to print ends up pausing for
the right duration, whereas all the other CPUs pause for twice as long:
once in oops_enter(), once in oops_exit().*

can't trust the integrity of the kernel anymore:

64-bit random ID for oopses:

Called when the architecture exits its oops handler, after printing everything.

This thread may hit another WARN() in the panic path.

CONFIG_SMP

100 usecs
1 second
0 usecs
1 second

filled in by handler

CONFIG_NUMA_BALANCING
CONFIG_SCHED_DEBUG

sys_waitid() overwrites everything in ru

<https://gitlab.com/manufacturaind/booklet-scripts>

booklet-impose-a5

A script to turn an A5 PDF into a set of 2-up A4 pages for double-sided printing. It's a way to quickly perform imposition so we can print A5 booklets in a double-sided A4 printer so that they come out ready to be folded and stapled.

The script uses `pdfofiimpose` with a `.plan` file with the imposition logic, which is directly based on an [original recipe](#) by [Pierre Marchand](#), the author of `pdfofiimpose`. We are also indebted to [stdin](#) for pointing the right way.

Read the `.plan` file source for an explanation of the logic to arrange and sort the pages so that they're laid out in the correct order.

Usage

```
booklet-impose-a5 source.pdf
```

and it will create `source-a4.pdf`.

booklet-get-page-total

Returns the total number of pages on a PDF file.

```
$ booklet-get-page-total source.pdf
24
```

dotfiles and GNU stow

```
# download and link to our config files  
git clone https://gitlab.com/manufacturaind/dotfiles.git  
cd dotfiles  
stow */
```

atool!

```
# decompress any archive  
aunpack archive.tar.bz2  
# and also compress  
apack archive.zip files/*
```

Scraping sources for inspiration

gallery-dl

```
# get photo albums  
gallery-dl https://flickr.com/photos/fdctsevilla/  
gallery-dl https://instagram.com/presentandcorrect/  
gallery-dl https://twitter.com/lubalincenter/
```

youtube-dl / yt-dlp

```
# download videos
yt-dlp https://www.youtube.com/watch?v=1M9VA0UBRQ0
# or only the audio
yt-dlp -x https://www.youtube.com/watch?v=1M9VA0UBRQ0
```

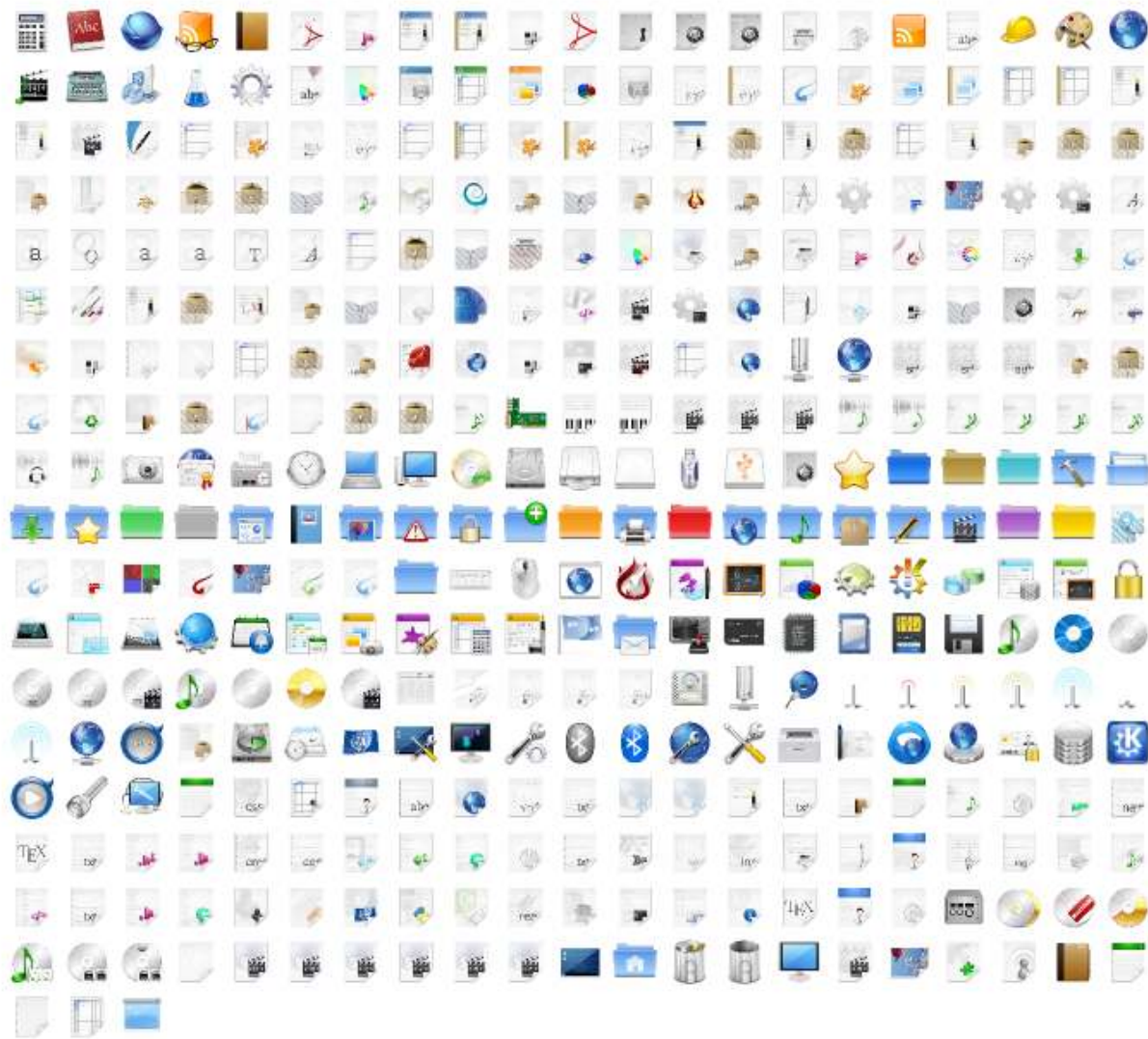
plain wget

```
# download image sequence  
for n in `seq 1 80`; do wget https://example.com/img/$n.jpg;
```

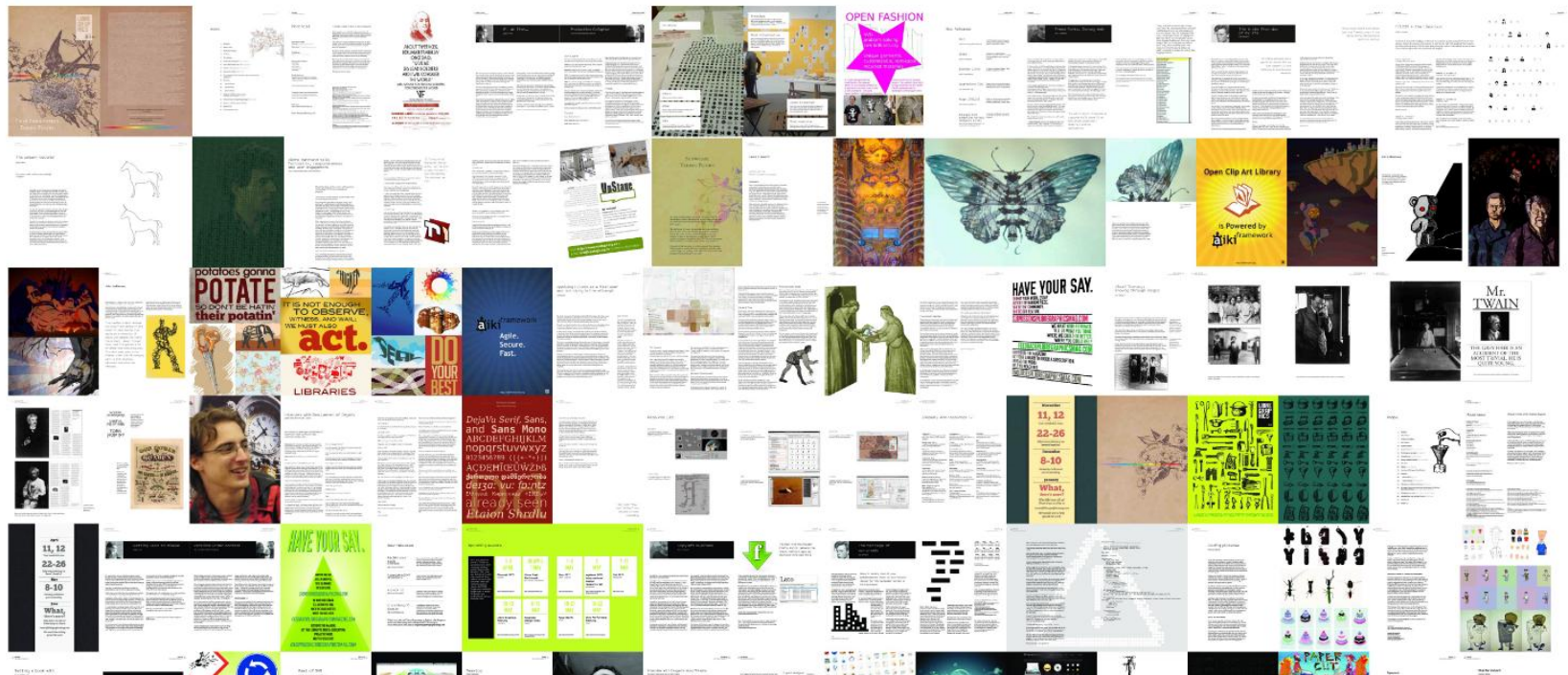
For mostly everything else
there's the [Internet Archive](#) <3

Libre Graphics magazine

```
montage *.png montage.png
```

\$ cat ~/libregraphicsmag/issues/*



\$ cat ~/libregraphicsmag/issues/*



LIBRE GRAPHICS MAGAZINE

First encounters, taking flight
Issue 1.1 à November 2010

- * If, at first... ginger coon à Editor's letter
- * Production colophon, Manufactura Indpendente
- * Make Art 2010 à Notebook
- * New releases
- * Freed fonts, strong web, Dave Crossland à Type Design Column
- * This is the first day of my life, Eric Schrijver à Column
- * F/LOSS in the classroom, Ludvine Loiseau à Dispatch
- * The unicorn tutorial, ginger coons à First time
- * Pierre Marchand talks FontMatrix, responsiveness and user engagement, Dave Crossland interviews Pierre Marchand à Interview
- * Camouflage and mimicry illuminated drawing, Laura C. Hewitt à Showcase
- * Pete Meadows à Showcase
- * John Leffasney à Showcase
- * Applying F/LOSS as a final user and not dying in the attempt, Lila Pagola à Feature
- * Visual literacy: knowing through images, Eric Schrijver à Visual Feature
- * Interview with Ben Laenen of DejaVu, ginger coons interviews Ben Laenen à Feature
- * Resource list
- * Glossary and resources
- * <https://gitlab.com/libregraphicsmag/vol1issue1>

Uses cases and affordances
Issue 1.2 à February 2011

- * Getting used to misuse, ginger coon à Editor's letter
- * Version under control, Manufactura Indpendente à Production colophon
- * New releases

Libre Graphics magazine Archive

Issue 2.4 – Capture

Cover, Issue 2.4

Source

↓ PDF high-res

↓ PDF low-res

Get a printed Issue ↗

This issue looks at Capture, the act of encompassing, emulating and encapsulating difficult things, subtle qualities. Through a set of articles we explore capture mechanisms, memory, archiving and preservation of volatile digital information, physicality and aesthetization of data.

MASTHEAD [Masthead](#)

EDITOR'S LETTER [Capturing capture](#)

– ginger coons

NOTEBOOK [Outils Libres Alternatifs](#)

– Sarah Garcin, Bachir Soussi Chiadmi, Raphaël Bastide

PRODUCTION COLOPHON [Dry Layout](#)

– Ana Isabel Carvalho, Ricardo Lafuente

NEW RELEASES [New Releases](#)

COLUMN [Evasive Maneuvers](#)

– Antonio Roberts

IN PRACTICE [+rwx data and pipes: Better design for otherwise non-writable infrastructure and unreadable data](#)

– Birgit Bachler, Walter Langelaar

COLUMN [Tonight we're making web sites like it's 1999](#)

– Eric Schrijver

DISPATCH [FONS: A recipe to make fonts out of bitmap images](#)

– Stéphanie Vilayphiou

SHOWCASE [Active Archives](#)

– Scandinavian Institute for Computational Vandalism

<https://gitlab.com/libregraphicsmag/archive>

Extract text from Scribus files

See the `scripts/scribus-extract.py` script.

Extracting image paths from Scribus files

```
cat 01-original-scribus-files/lgmag-1.1-p* | \
grep -oe '"[^ ]*(jpg|png\)"' | \
sort | uniq | \
sed 's//g; s/\.\\.//g' > images-1.1.txt
```

Re-downloading images from repos

After extracting the image lists from the previous recipe, we did this (example for 2.4):

```
mkdir -p images/2.4
cd images/2.4
while read f; do wget "https://gitlab.com/libregraphicsmag/vol2issue4/raw/master/$f"; done < ../../image-lists/i
```

Resizing images to a max width and height

We wanted max 1800px width and 1400px height, while keeping smaller images as they are.

```
cd images/2.4
for f in *; do echo $f; mogrify -resize 1800x1400> $f; done
```

or, to alter all images in subdirectories:

```
cd images
find . -name "*" | xargs mogrify -resize 970x4000\>
```

Review text files with a particular string

We needed to check every instance of a lowercase acronym (e.g. "svg") and edit it manually if it needed to be made uppercase again.

```
grep '\bsvg\b' issue* -lr | xargs vim -p
```

Turn all images into image links

We wanted to change all images into a thumbnail which would then link to the original image.

So this

```

```

should become

```
![](/images/2.1/picture.png)
```

So we use the following sed script:

```
cd content
find . -name "*.md" | xargs sed -i 's/!\[\]\(\(\images\[^\]\*\)\)/!\[\]\(\1)\(\1)/g; s/\(\images\//(\images-small\
```

The sed operation comes in two parts: first we create the "image inside link", and after that we fix the directory of the thumbnail so that it points to the `images-small` directory.

figlet / toilet

```
$ figlet "libreplanet"
```

```
  _ _ _ _ _      _ _ _ _ _      _ _ _ _ _      _ _ _ _ _  
| (-) | _ _ _ _ _ | _ _ _ _ _ | _ _ _ _ _ | _ _ _ _ _ | | | | | | | | | | | | | | | | | | | | | |
| | | | | \ / | \ / | \ / | \ / | \ / | \ / | \ / | \ / |  
| | | | | | ) | | | | | ) | | | | | ) | | | | | ) | | | | | ) |  
| _ _ _ _ _ / | \ | \ | \ | \ | \ | \ | \ | \ | \ | \ | \ |  
      | _ |
```

badger

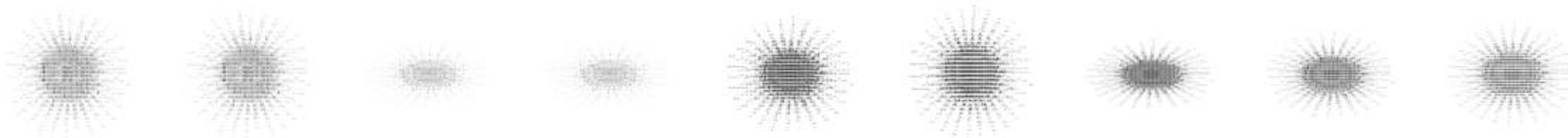
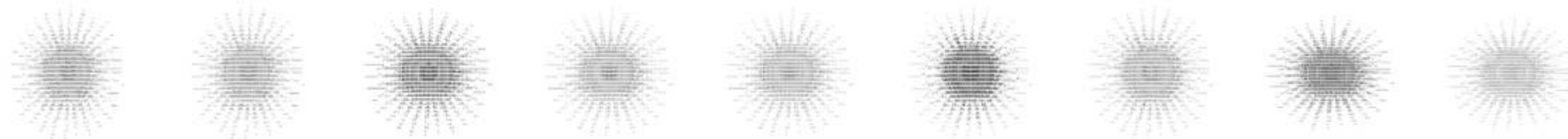
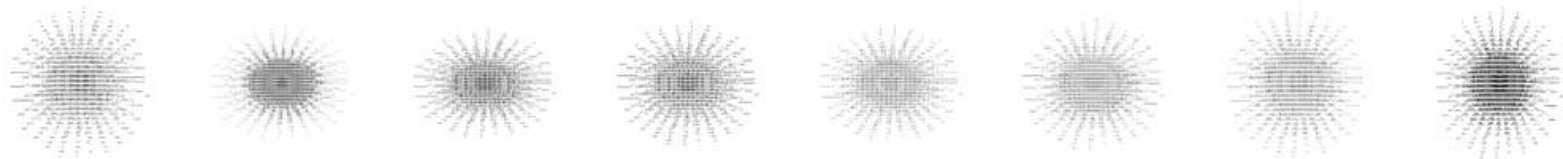
```
badger gnu.txt --font 5x8  
--leading=-1 --tracking=-1  
--textcolor="#dddddd" --bgcolor="#222222"
```

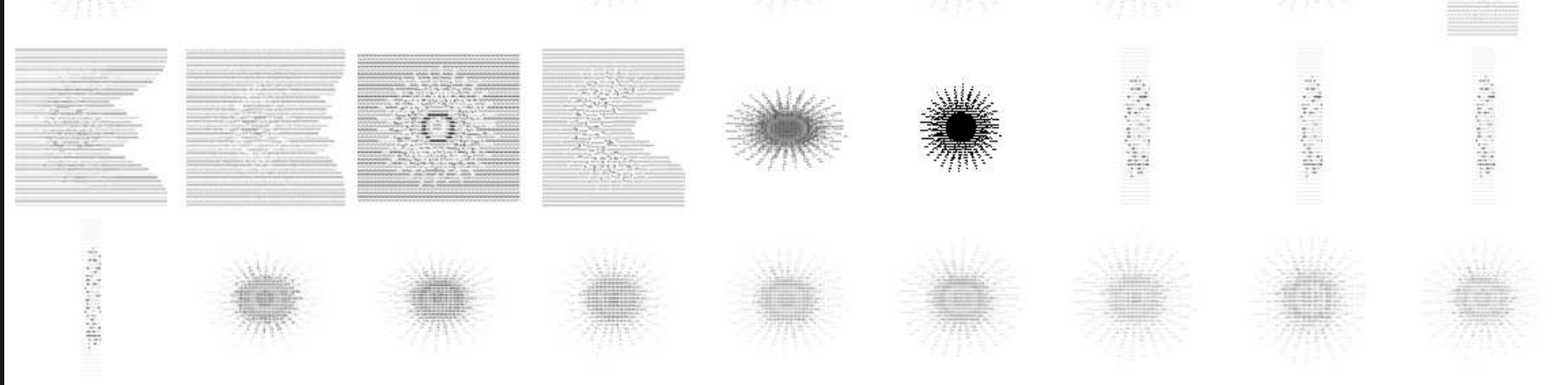
<https://gitlab.com/manufacturaind/badger>

gnucash.org

gnu.org

GNU.ORG





Free
software
Free
society

FREE
SOFTWARE
FREE
SOCIETY

Free
software
Free
society

FREE
SOFTWARE
FREE
SOCIETY

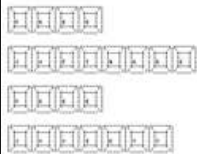
FREE
SOFTWARE
FREE
SOCIETY

FREE
SOFTWARE
FREE
SOCIETY

FREE
SOFTWARE
FREE
SOCIETY



FREE
SOFTWARE
FREE
SOCIETY



FREE
SOFTWARE
FREE
SOCIETY

FREE
SOFTWARE
FREE
SOCIETY

FREE SOFTWARE
FREE SOCIETY

FREE SOFTWARE
FREE SOCIETY

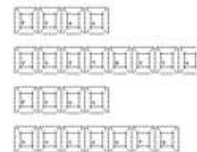
FREE SOFTWARE
FREE SOCIETY

FREE SOFTWARE
FREE SOCIETY

Free software
Free society



Free
software
Free
society



Free
software
Free
society

Free
software
Free
society

Free software
Free society

Free software
Free society

Free software
Free society

Free software
Free society

I POWER
GNU

I POWER
GNU

I POWER
GNU

I power GNU



I power
GNU

I POWER
GNU

I power
GNU

I POWER
GNU

I power GNU

I power GNU

I power GNU

I power GNU

I <3
GNU

I <3 GNU

I <3 GNU

I <3 GNU

I <3 GNU

I <3 GNU

I <3 GNU

I <3 GNU

I <3 GNU

I <3 GNU

I <3 GNU

I <3 GNU

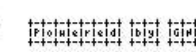
I <3 GNU

POWERED
BY GNU

POWERED
BY GNU

Powered
by GNU

Powered by
GNU



Powered by
GNU

Powered by
GNU

Powered by
GNU

Powered by
GNU

Powered by GNU

Powered by GNU





typebits

Group ID: 1447661 [Leave group](#)



New subgroup

New project

Fonts and scripts for the Type:Bits workshops by Manufactura Independente

Subgroups and projects

Shared projects

Archived projects

Search by name

Updated date

	typebits.gitlab.io	Documentation for the Type:Bits workflow in MkDocs	★ 1	12 hours ago
	font-skeleton	Skeleton for bitmap fonts, made to be forked and edited	★ 0	1 year ago
	font-flamingo	A wavy typeface with a whimsical attitude.	★ 0	2 years ago
	font-mozblock	A typeface designed inside Minecraft at the "Design open web fonts with Minecraft" wor...	★ 0	2 years ago
	font-setperset	A 7x7 bitmap font created at the fonts.txt workshop in Barcelona (2014)	★ 0	2 years ago
	font-spalte		★ 0	2 years ago
	font-roseta	A spiraling typeface with an attitude. You can hear it singing if you get close enough.	★ 0	3 years ago
	font-ladrillo	A stern font that nevertheless still knows how to have fun.	★ 0	3 years ago
	font-muralla	A sturdy typeface which won't take no for an answer.	★ 0	3 years ago

Creating a new font

Want to dive in the wonderful world of bitmap font drawing?

There are two available methods to quickly get started with a ready-to-edit bare font template:

1. **Fork the font-skeleton repository.** Faster to get started, but needs some tweaking to the font's configuration.
2. **Use the cookiecutter-bitmapfont template.** Requires you to install `cookiecutter`, but makes it easier to define the configuration values through a series of simple questions.

Additionally, you can try forking an existing font to have an existing base to work from -- just follow method 1 and instead of `font-skeleton`, choose another repository to fork from.

Method 1: Fork the font skeleton repo

You can get started with a bare version of a bitmap font by forking the `font-skeleton` repository on the GitLab.com web interface.

If you want an easier start, try forking one of the existing fonts in the [Type:Bits repository list](#).

Table of contents

- [Method 1: Fork the font skeleton repo](#)
- [Method 2: Use the Cookiecutter template](#)
- [Now what?](#)

Type:Bits Building font files

About

Fonts

Principles

Past workshops

Tools & Recipes ^

Creating a new font

Building font files

Font specimens

The BMF font format

Colophon

```
sudo apt-get install python-virtualenv
```

Now, just run `make install` to pull and install the remaining dependencies.

Finally, clone the `bitmapfontbuilder` repository to the same directory where you've pulled this font, so that both are inside the same folder.

```
cd ..
git clone https://gitlab.com/typebits/bitmapfontbuilder
```

Building locally

Whenever you want to re-generate the font files from the source, just type

```
make build
```

This will generate the files in FontForge (`.sfd`), TrueType (`.ttf`) and OpenType (`.otf`) formats.

Building and deploying to a repository

A better workflow is to create a Git repository from this font directory and running

```
make deploy
```

This command is identical to `make build` but pulls changes from the repository before building, generates the font files, commits the newly built files and pushes the changes back into the remote repository. This is particularly handy if you're working with more people and want to incorporate their changes quickly.

Table of contents

[Method 1: Using GitLab to automatically build font files](#)

[Method 2: Build font files in your computer](#)

[Installing dependencies](#)

[Building locally](#)

[Building and deploying to a repository](#)

transpacing

a script to transplant spacing from one font to another.

```
font with no spacing  
+ similar font with good spacing  
-----  
= font with decent spacing
```

abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789.:;(*!?)

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over

The quick brown fox jumps

The quick brown

Name: M+ 1m
Style: regular
Type: TrueType font
Size: 1.1 MB
Version: Version 1.042
Copyright: © 2011 M+ FONTS PROJECT

abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789.:;(*!?)

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over

The quick brown fox jumps

The quick brown

Name:
Style:
Type:
Size:
Version:
Copyright:
Project:

```
stotal 20  
drwxrwxr-x.  4 rlafuente rlafuente 4096 Feb 17 14:25 .  
drwxrwxr-x. 34 rlafuente rlafuente 4096 Feb 17 13:41 ..  
drwxrwxr-x.  2 rlafuente rlafuente 4096 Feb 17 13:43 ffilters  
#!/usr/bin/env python  
import sys  
import os, shutil  
import fontforge  
  
if len(sys.argv) != 4:  
    print "Usage: <source font> <target font> <output filename>"  
    sys.exit()  
  
SOURCE = os.path.abspath('./sources/DejaVuSansCondensed-Bold.ttf')  
*[master][~/code/tinytypetools/transpacing]* ./transpacing Reglo-Bold.sfd mplus-1m-regular  
ttf franken.ttf  
Reglo-Bold.sfd  
mplus-1m-regular.ttf  
franken.ttf  
*[master][~/code/tinytypetools/transpacing]* ls  
franken.ttf mplus-1m-regular.ttf Reglo-Bold.sfd transpacing  
*[master][~/code/tinytypetools/transpacing]* go franken.ttf  
*[master][~/code/tinytypetools/transpacing]* go mplus-1m-regular.ttf  
*[master][~/code/tinytypetools/transpacing]* import  
import: missing an image filename 'import' @ error/import.c/ImportImageCommand/1295.  
*[master][~/code/tinytypetools/transpacing]* import ~/transpacing.png  
*[master][~/code/tinytypetools/transpacing]* import ~/transpacing.png  
*[master][~/code/tinytypetools/transpacing]* import ~/transpacing.png
```

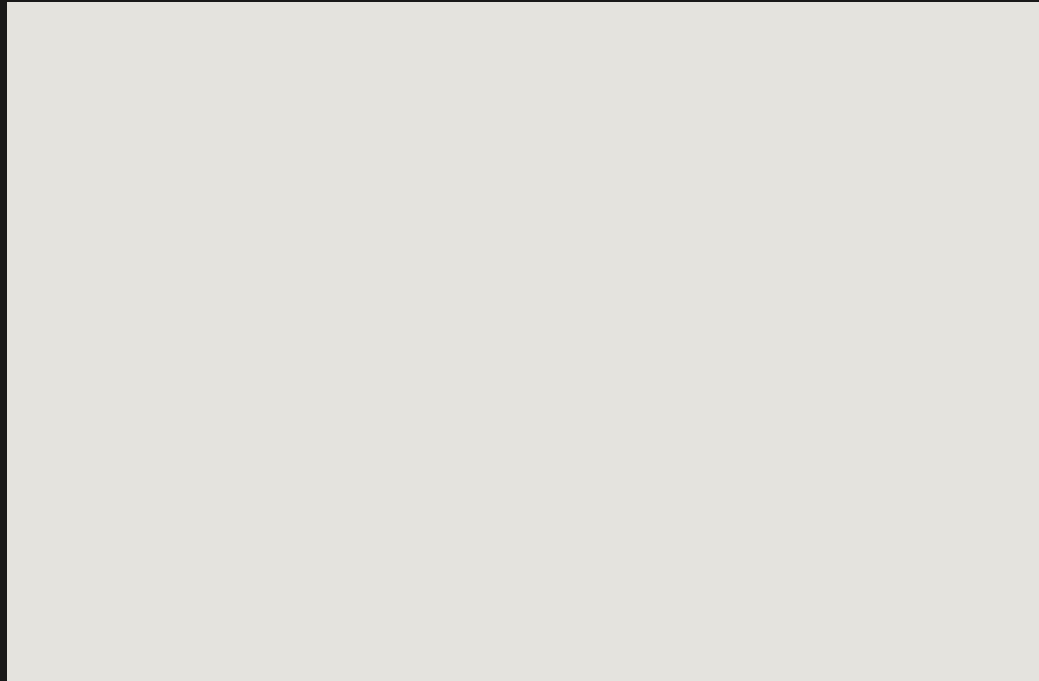
Install Font

fontconvert

```
fontconvert Douar.sfd --otf --ttf --woff
```

ffilters

```
import fontforge
font = fontforge.open('douar.sfd')
for glyph in font:
    font[glyph].stroke('circular', 30, 'square', 'bevel', ('cle
font.generate('douar-outline.ttf')
```



Thank you!

manufacturaindependente.org