
fontpreview

Release 1.1.0

Matteo Guadrini

Dec 18, 2020

CONTENTS:

- 1 Installation 3**
 - 1.1 Python 3
 - 1.2 Installation 3
- 2 Example 5**
 - 2.1 FontPreview example 5
 - 2.2 FontBanner example 7
 - 2.3 FontLogo example 10
 - 2.4 FontWall example 10
 - 2.5 FontPage example 12
 - 2.6 FontPageTemplate example 16
 - 2.7 FontBooklet example 18
 - 2.8 Declarative object creation 18
- 3 Command line 19**
 - 3.1 Simple usage 20
 - 3.2 Advanced usage 20
- 4 fontpreview package 23**
 - 4.1 fontpreview modules 23
- 5 Indices and tables 31**
- Python Module Index 33**
- Index 35**

fontpreview is a python library, which allows you to create simple and advanced previews of specific fonts.

In addition, the library includes some classes that allow the advanced creation of preview pages of the characters that make up a font.

INSTALLATION

Here are the installation instructions

1.1 Python

fontpreview is written in python3 (3.6 and higher). The only external library required is [Pillow](#) (fork of PIL):

1.2 Installation

```
$ pip install --user fontpreview
```

Note: If you want to use the command line tool, you need to install the system-wide library: `pip install fontpreview`

EXAMPLE

Here are some examples that allow basic and advanced use of the library.

2.1 FontPreview example

FontPreview is a class that creates objects that allow the representation of a font in an image. By default, it creates a white rectangle with a preview of the letters *a b c d e f* in black.

```
from fontpreview import FontPreview
fp = FontPreview('/tmp/noto.ttf')    # path of font file
fp.save('/tmp/fp.png')              # default directory is working directory
```

This is result:

a b c d e f

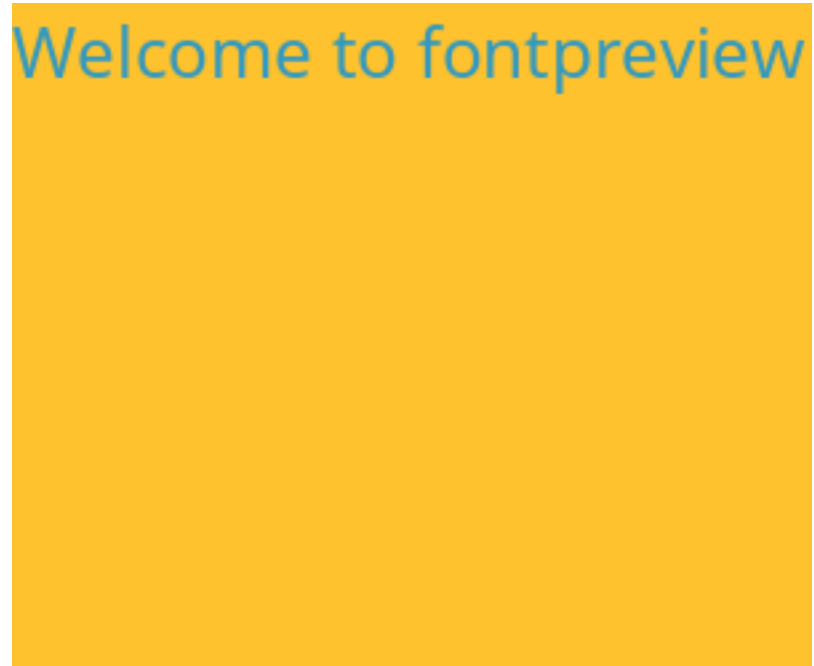
Now, let's modify some properties.

```
fp.font_text = 'Welcome to fontpreview'
fp.bg_color = (253, 194, 45)          # background color. RGB color: yellow
fp.dimension = (300, 250)             # specify dimension in pixel: 300 x 250
fp.fg_color = (51, 153, 193)         # foreground or font color. RGB color: blue
fp.set_font_size(50)                  # set font size to 50 pixel
fp.set_text_position('ltop')          # place the text at the top left.
```

(continues on next page)

(continued from previous page)

```
# before saving the image, you need to draw it again
fp.draw()
fp.save('/tmp/fp.png')
```



A background image can also be set.

```
fp.bg_image = '/tmp/python.png'      # a background image
fp.draw()                             # draw it again
fp.save('/tmp/fp.png')
```



2.2 FontBanner example

FontBanner is a *FontPreview*-based class, which adds some features to work with one or more objects based on the *FontPreview* class. With this object since its creation, it is possible to define the orientation: *landscape* or *portrait*.

```
from fontpreview import FontBanner
fb = FontBanner('/tmp/noto.ttf', 'landscape', bg_color=(253, 194, 45))  # path of _
↪font file
fb.save('/tmp/fb.png')
```



Let's go and change some of the properties.

```
fb.set_mode('fontname')          # set font_text properties to font name
fb.set_orientation('portrait')    # set vertical orientation of image
fb.save('/tmp/fb.png')
```



Noto Sans

And now, let's add the *FontPreview* object created earlier.

```
fb.font_text = 'Python'  
fb.set_font_size(50)           # change font size: FontPreview method  
fb.bg_color = 'white'         # set color with name string  
fb.set_orientation((300, 800)) # change orientation and size again with tuple  
fb.draw()                     # draw it again  
fb.add_image(fp, (0, 150))    # add FontPreview object to FontBanner object  
fb.save('/tmp/fb.png')
```

Welcome to fontpreview



Python

2.3 FontLogo example

FontLogo is a *FontPreview*-based class, which represents a square where inside there are one or two letters. The *fontpreview* package logo was generated with this class.

```
from fontpreview import FontLogo
fl = FontLogo('/tmp/ noto.ttf', 'Fp')      # specify font and letters. Max 2
fl.save('/tmp/fl.png')
```



Being a *FontPreview* based object, it inherits all its characteristics.

```
fl.font_text = 'TS'
fl.bg_color = (45, 121, 199)      # background color. RGB color: blue
fl.fg_color = 'white'             # foreground color. RGB color: white
fl.set_text_position('rbelow')    # position is "right-below"
fl.save('/tmp/fl.png')
```



2.4 FontWall example

FontWall is a class that represents an image in which there are multiple objects based on the *FontPreview* class.

This object accepts a list of font paths (with which it automatically builds *FontBanner* objects) or a list of objects based on the *FontPreview* class.

The *FontWall* object has a mode, which can be *horizontal* or *vertical*, or just specify the usual tuple of with x and y (x, y) axis. It also accepts a maximum of tiles per row (if the orientation is horizontal) or column (if the orientation is vertical).

```
from fontpreview import FontBanner, FontWall
# Define the various parts of wall
fb = FontBanner('/tmp/ noto.ttf', 'landscape', mode='fontname')
fb2 = FontBanner('/tmp/ noto.ttf', 'landscape', mode='alpha')
fb3 = FontBanner('/tmp/ noto.ttf', 'landscape', mode='letter')
fb4 = FontBanner('/tmp/ noto.ttf', 'landscape', mode='paragraph')
fw = FontWall([fb, fb2, fb3, fb4])
fw.save('/tmp/fw.png')
```

Noto Sans

Aa Bb Cc Dd Ee Ff
1 2 3 4 5 6 7 8 9 0a b c d e f
g h i j k l
m n o p q r
s t u v w x y zLorem ipsum dolor sit amet,
consectetur adipiscing elit.

Any changes made on the parts of the wall are made to the final result.

```
# Modify properties of first banner
fb.font_text = 'Harry Potter'
fb.bg_color = (43, 43, 43)
fb.fg_color = 'white'
fb.set_font_size(120)
# Modify properties of second banner
fb2.font_text = 'Harry Potter is a series of seven fantasy novels\nwritten by British_
↳author J. K. Rowling.'
fb2.bg_color = (150, 45, 46)
fb2.fg_color = 'white'
fb2.set_font_size(100)
fb2.set_text_position('ltop')
# Modify properties of third banner
fb3.font_text = 'The series was originally published in English by two major_
↳publishers,\nBloomsbury in the United Kingdom and Scholastic Press in the United_
↳States. '
fb3.bg_color = (63, 55, 36)
fb3.fg_color = 'white'
fb3.set_font_size(100) # the font is resized automatically because_
↳it exceeds the size of the banner
fb3.set_text_position('rcenter')
# Modify properties of last banner
fb4.font_text = 'A series of many genres, including fantasy, drama,\ncoming of age,_
↳and the British school story'
fb4.bg_color = (205, 193, 87)
fb4.fg_color = 'black'
fb4.set_font_size(100)
fb4.set_text_position('rbelow')
fw.draw(2) # draw it again, specify max_tile
fw.save('/tmp/fw.png')
```



2.5 FontPage example

FontPage is a class that represents a sample page per font. This object consists of three parts: header, body and footer. These three parts have a standard size defined by a *FontPageTemplate* (see below).

```
from fontpreview import FontPage, FontBanner
# Define the various parts of wall
header = FontBanner('/tmp/noto.ttf', 'landscape', mode='fontname')
body = FontBanner('/tmp/noto.ttf', 'landscape', mode='paragraph')
footer = FontBanner('/tmp/noto.ttf', 'landscape', mode='letter')
# Create FontPage object
fpage = FontPage()
fpage.set_header(header)
fpage.set_body(body)
fpage.set_footer(footer)
# Design all parts
fpage.draw()
fpage.save('/tmp/fpage.png')
```

Noto Sans

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.

a b c d e f
g h i j k l
m n o p q r
s t u v w x y z

Even with this object, any changes made to the individual parts of the page appear in the final result.

It is also possible to add a FontLogo object to the header, after the header has been defined.

```
from fontpreview import FontLogo
fl = FontLogo('/tmp/noto.ttf', 'Fp')      # create logo
fpage.set_logo(fl)                       # set logo on header
fpage.body.bg_color = (253, 194, 45)
fpage.body.set_font_size(150)
fpage.draw()
fpage.save('/tmp/fpage.png')
```

Fp

Noto Sans

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.

a b c d e f
g h i j k l
m n o p q r
s t u v w x y z

2.6 FontPageTemplate example

FontPageTemplate is a class that represents a template applicable to the *FontPage* object.

In this object, only the specifications of each part of the *FontPage* object (header, body, footer) are defined: font size, text position, unit.

The units (default 6) are equal parts divided across the height of the page.

```
from fontpreview import FontPageTemplate
template = FontPageTemplate(3508)           # max height of page
template.set_body(170, 1, 'lcenter')       # font_size, units, text_position
template.set_footer(100, 4, 'lcenter')     # font_size, units, text_position
# Create FontPage object
fpage = FontPage(template=template)
fpage.set_header(header)
fpage.set_body(body)
fpage.set_footer(footer)
# Design all parts
fpage.draw()
fpage.save('/tmp/fpage_template.png')
```

Fp

Noto Sans

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.

a b c d e f
g h i j k l
m n o p q r
s t u v w x y z

2.7 FontBooklet example

FontBooklet is a class that represents a book of *FontPage* object.

```
from fontpreview import FontPage, FontBanner, FontBooklet
# Define the various parts of page
header = FontBanner('/tmp/ noto.ttf', 'landscape' , mode='fontname')
body = FontBanner('/tmp/ noto.ttf', 'landscape' , mode='paragraph')
footer = FontBanner('/tmp/ noto.ttf', 'landscape' , mode='letter')
# Create FontPage object
fpage1 = FontPage(header=header, body=body, footer=footer)
fpage2 = FontPage(header=header, body=body, footer=footer)
# Design all parts
fpage1.draw()
fpage2.draw()
# Create book
book = FontBooklet(fpage1, fpage2)
book.save('/tmp/ noto_book/') # save page1.png, page2.png in /tmp/ noto_book/_
↪ folder
```

2.8 Declarative object creation

Each *FontPreview* and *FontPage* based object in this module has a declarative instance implementation.

```
from fontpreview import FontPreview, FontBanner, FontLogo, FontPage
# FontPreview object
fp = FontPreview('/tmp/ noto.ttf',
                 font_size=50,
                 font_text='some text',
                 color_system='RGB',
                 bg_color='blue',
                 fg_color='yellow',
                 dimension=(800, 400))
# FontBanner object
fb = FontBanner('/tmp/ noto.ttf',
                orientation='portrait',
                bg_color='blue',
                fg_color='yellow',
                mode='paragraph',
                font_size=70,
                color_system='RGB')
# FontLogo object
fl = FontLogo('/tmp/ noto.ttf',
              'F1',
              size=(170, 170),
              bg_color='yellow',
              fg_color='blue',
              font_size=50,
              color_system='RGB')
# FontPage object
page = FontPage(header=fb, logo=fl, body=fb, footer=fb)
page.draw()
```

COMMAND LINE

Here we explain how to use the *fontpreview* tool on the command line

Note: If you want to use the command line tool, you need to install the system-wide library: `pip install fontpreview`

This is help system:

```
$ fp --help
usage: fp [-h] [--verbose] [--version] [-t TEXT] [-b BG_COLOR] [-f FG_COLOR] [-i IMAGE]
        [-d DIMENSION DIMENSION] [-s SAVE_PATH] [-p TEXT_POSITION] [-z SIZE]
        font

FontPreview cli

positional arguments:
  font                  font file path

optional arguments:
  -h, --help            show this help message and exit
  --verbose, -v         enable verbosity, for debug
  --version, -V         show program's version number and exit
  -t TEXT, --text TEXT  text include to preview image (default: a b c d e f)
  -b BG_COLOR, --background BG_COLOR
                        background color (default: white)
  -f FG_COLOR, --foreground FG_COLOR
                        foreground color (default: black)
  -i IMAGE, --background-image IMAGE
                        background image path
  -d DIMENSION DIMENSION, --dimension DIMENSION DIMENSION
                        dimension x and y (default: 700x327)
  -s SAVE_PATH, --save SAVE_PATH
                        save file path (default: current directory)
  -p TEXT_POSITION, --text-position TEXT_POSITION
                        save file path (default: center)
  -z SIZE, --size SIZE  size of font (default: 64)
```

3.1 Simple usage

Save *fontpreview* image in a current directory from font file:

```
$ fp /tmp/noto.ttf
```

a b c d e f

3.2 Advanced usage

Use `-v` for debugging; `-d` setting dimension with **x** and **y** axis; `-b` setting background colors, `-f` setting foreground colors, `-p` setting text position, `-z` setting font size and `-s` specified file path to save.

For the color reference: [colors](#)

```
$ fp /tmp/noto.ttf -v -t 'Hello Noto' -d 1000 1000 -b 'green' -f 'blue' -p 'lcenter' -  
↪z 50 -s /tmp/fp.png  
DEBUG: set text position: "lcenter"  
DEBUG: font object => font_name:('Noto Sans', 'Regular'),font_size:50,text:Hello Noto,  
↪text_position:(0, 473),dimension:(1000, 1000)
```



Hello Noto

FONTPREVIEW PACKAGE

This package contains three modules that allow, through specific classes, to create simple and advanced font previews.

4.1 fontpreview modules

4.1.1 fontpreview

fontpreview module contains **FontPreview** class

```
class fontpreview.fontpreview.FontPreview(font, font_size=64, font_text='a b c d e f', color_system='RGB', bg_color='white', fg_color='black', dimension=(700, 327))
```

Bases: object

Class that represents the preview of a font

```
__init__(font, font_size=64, font_text='a b c d e f', color_system='RGB', bg_color='white', fg_color='black', dimension=(700, 327))
```

Object that represents the preview of a font

Parameters

- **font** – font file
- **font_size** – font size. Default is 64.
- **font_text** – font text representation. Default is ‘a b c d e f’.
- **color_system** – color system string. Default is ‘RGB’.
- **bg_color** – background color of preview. Default is ‘white’.
- **fg_color** – foreground or font color of preview. Default is ‘black’.
- **dimension** – dimension of preview. Default is 700x327.

```
__str__()
```

String representation of font preview

Returns string

```
__weakref__
```

list of weak references to the object (if defined)

```
draw(align='left')
```

Draw image with text based on properties of this object

Parameters **align** – alignment of text. Available ‘left’, ‘center’ and ‘right’

Returns None

save (*path*='/home/docs/checkouts/readthedocs.org/user_builds/fontpreview/checkouts/latest/docs/source/fontpreview.png')
Save the preview font

Parameters *path* – path where you want to save the preview font

Returns None

set_font_size (*size*)
Set size of font

Parameters *size* – size of font

Returns None

set_text_position (*position*)
Set position of text

Parameters *position* – Position can be a tuple with x and y axis, or a string. The strings available are 'center', 'top', 'below', 'rcenter', 'rtop', 'rbelow', 'lcenter', 'ltop' and 'lbelow'.

Returns None

show ()
Displays this image.

Returns None

4.1.2 fontbanner

fontbanner module contains **FontBanner**, **FontLogo** and **FontWall** class

```
class fontpreview.fontbanner.FontBanner (font, orientation='landscape', bg_color='white',  
                                           fg_color='black', mode='letter', font_size=64,  
                                           color_system='RGB')
```

Bases: *fontpreview.fontpreview.FontPreview*

Class that represents the banner of a font

```
__init__ (font, orientation='landscape', bg_color='white', fg_color='black', mode='letter',  
          font_size=64, color_system='RGB')
```

Object that represents the banner of a font

Parameters

- **font** – font file
- **orientation** – the orientation of the banner; 'landscape', 'portrait' or tuple(x,y).
- **bg_color** – background color of preview. Default is 'white'.
- **fg_color** – foreground or font color of preview. Default is 'black'.
- **mode** – the text inside the banner; 'letter', 'fontname', 'paragraph', 'alpha' and 'combination'.
- **font_size** – font size. Default is 64.
- **color_system** – color system string. Default is 'RGB'.

```
__str__ ()
```

String representation of font banner

Returns string

add_image (*image, position*)

Adds an additional image to the banner

Parameters

- **image** – path of image
- **position** – position of image

Returns None

set_mode (*mode, align='center'*)

Set the text mode

Parameters

- **mode** – mode that sets the text in the banner
- **align** – alignment of text. Available 'left', 'center' and 'right'

Returns None

set_orientation (*orientation, font_position='center'*)

Set orientation of banner

Parameters

- **orientation** – the orientation of the banner; 'landscape' or 'portrait'
- **font_position** – font position respect dimension of banner

Returns None

```
class fontpreview.fontbanner.FontLogo (font,          letters,          size=(100,          100),
                                         bg_color='white', fg_color='black', font_size=64,
                                         color_system='RGB')
```

Bases: [fontpreview.fontpreview.FontPreview](#)

Class that represents the logo of a font

```
__init__ (font, letters, size=(100, 100), bg_color='white', fg_color='black', font_size=64,
          color_system='RGB')
```

Object that represents the logo of a font

Parameters

- **font** – font file
- **letters** – One or two letters (or anything)
- **size** – size of logo square. Default is (100, 100)
- **bg_color** – background color of preview. Default is 'white'.
- **fg_color** – foreground or font color of preview. Default is 'black'.
- **font_size** – font size. Default is 64.
- **color_system** – color system string. Default is 'RGB'.

new_size (*size*)

Define new size of FontLogo object

Parameters **size** – size of fontlogo object

Returns None

```
class fontpreview.fontbanner.FontWall (fonts, max_tile=2, mode='horizontal')
```

Bases: object

Class that represents the wall of fonts

```
__init__ (fonts, max_tile=2, mode='horizontal')
```

Object that represents the wall of fonts

Parameters

- **fonts** – font list; string or FontPreview object
- **max_tile** – maximum tile per row/column
- **mode** – image alignment, ‘horizontal’ or ‘vertical’

```
__str__ ()
```

String representation of font wall

Returns string

```
__weakref__
```

list of weak references to the object (if defined)

```
draw (max_tile)
```

Draw wall with fonts on properties of this object

Parameters **max_tile** – maximum tile per row

Returns None

```
save (path='/home/docs/checkouts/readthedocs.org/user_builds/fontpreview/checkouts/latest/docs/source/fontwall.png')
```

Save the font wall

Parameters **path** – path where you want to save the font wall

Returns None

```
show ()
```

Displays this image.

Returns None

```
fontpreview.fontbanner.resize (image, bg_image)
```

Resize image

Parameters

- **image** – image to resize
- **bg_image** – background image

Returns Image object

4.1.3 fontpage

fontpage module contains **FontPage** and **FontPageTemplate** class

```
class fontpreview.fontpage.FontBooklet (*pages)
```

Bases: object

Class that represents the booklet of a font page

```
__init__ (*pages)
```

Object that represents the booklet of a font page

Parameters **pages** – FontPage’s object

__iter__ ()

Iterating on each FontPage

Returns next value

__weakref__

list of weak references to the object (if defined)

save (*folder*, *extension*='png')

Save on each FontPage image

Parameters

- **folder** – path folder where you want to save each font page
- **extension** – extension of image file. Default is 'png'

Returns None

class fontpreview.fontpage.**FontPage** (*template*=None, *dimension*=(2480, 3508), *header*=None, *logo*=None, *body*=None, *footer*=None)

Bases: object

Class that represents the page of a font banners

__init__ (*template*=None, *dimension*=(2480, 3508), *header*=None, *logo*=None, *body*=None, *footer*=None)

Object that represents the page of a font banners

Parameters

- **template** – template used to build the page
- **dimension** – dimension of page. Default A4 in pixels.
- **header** – header of fontpage object
- **logo** – logo of fontpage object on header part
- **body** – body of fontpage object
- **footer** – footer of fontpage object

__str__ ()

String representation of font page

Returns string

__weakref__

list of weak references to the object (if defined)

draw (*separator*=True, *sep_color*='black', *sep_width*=5)

Draw font page with header, logo, body and footer

Parameters

- **separator** – line that separates the parts
- **sep_color** – separator color
- **sep_width** – separator width

Returns None

save (*path*='/home/docs/checkouts/readthedocs.org/user_builds/fontpreview/checkouts/latest/docs/source/fontpage.png')

Save the font page

Parameters **path** – path where you want to save the font page

Returns None

set_body (*body*)

Set body of Font page

Parameters **body** – FontPreview object

Returns None

set_footer (*footer*)

Set footer of Font page

Parameters **footer** – FontPreview object

Returns None

set_header (*header*)

Set header of Font page

Parameters **header** – FontPreview object

Returns None

set_logo (*logo*)

Set logo of Font page

Parameters **logo** – FontLogo object

Returns None

show ()

Displays this image.

Returns None

class fontpreview.fontpage.**FontPageTemplate** (*page_height=3508, units_number=6*)

Bases: object

Class representing the template of a FontPage object

__init__ (*page_height=3508, units_number=6*)

Object representing the template of a FontPage object

Parameters

- **page_height** – height of FontPage object. Default is 3508.
- **units_number** – division number to create the units

__str__ ()

String representation of font page

Returns string

__weakref__

list of weak references to the object (if defined)

set_body (*font_size, units, text_position*)

Setting the body properties

Parameters

- **font_size** – the body font size
- **units** – the body units number

- **text_position** – the body text position

Returns None

set_footer (*font_size, units, text_position*)

Setting the footer properties

Parameters

- **font_size** – the footer font size
- **units** – the footer units number
- **text_position** – the footer text position

Returns None

set_header (*font_size, units, text_position*)

Setting the header properties

Parameters

- **font_size** – the header font size
- **units** – the header units number
- **text_position** – the header text position

Returns None

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

f

`fontpreview.fontbanner`, [24](#)
`fontpreview.fontpage`, [26](#)
`fontpreview.fontpreview`, [23](#)

Symbols

`__init__()` (*fontpreview.fontbanner.FontBanner* method), 24
`__init__()` (*fontpreview.fontbanner.FontLogo* method), 25
`__init__()` (*fontpreview.fontbanner.FontWall* method), 26
`__init__()` (*fontpreview.fontpage.FontBooklet* method), 26
`__init__()` (*fontpreview.fontpage.FontPage* method), 27
`__init__()` (*fontpreview.fontpage.FontPageTemplate* method), 28
`__init__()` (*fontpreview.fontpreview.FontPreview* method), 23
`__iter__()` (*fontpreview.fontpage.FontBooklet* method), 27
`__str__()` (*fontpreview.fontbanner.FontBanner* method), 24
`__str__()` (*fontpreview.fontbanner.FontWall* method), 26
`__str__()` (*fontpreview.fontpage.FontPage* method), 27
`__str__()` (*fontpreview.fontpage.FontPageTemplate* method), 28
`__str__()` (*fontpreview.fontpreview.FontPreview* method), 23
`__weakref__` (*fontpreview.fontbanner.FontWall* attribute), 26
`__weakref__` (*fontpreview.fontpage.FontBooklet* attribute), 27
`__weakref__` (*fontpreview.fontpage.FontPage* attribute), 27
`__weakref__` (*fontpreview.fontpage.FontPageTemplate* attribute), 28
`__weakref__` (*fontpreview.fontpreview.FontPreview* attribute), 23

A

`add_image()` (*fontpreview.fontbanner.FontBanner* method), 24

D

`draw()` (*fontpreview.fontbanner.FontWall* method), 26
`draw()` (*fontpreview.fontpage.FontPage* method), 27
`draw()` (*fontpreview.fontpreview.FontPreview* method), 23

F

`FontBanner` (class in *fontpreview.fontbanner*), 24
`FontBooklet` (class in *fontpreview.fontpage*), 26
`FontLogo` (class in *fontpreview.fontbanner*), 25
`FontPage` (class in *fontpreview.fontpage*), 27
`FontPageTemplate` (class in *fontpreview.fontpage*), 28
`FontPreview` (class in *fontpreview.fontpreview*), 23
`fontpreview.fontbanner` module, 24
`fontpreview.fontpage` module, 26
`fontpreview.fontpreview` module, 23
`FontWall` (class in *fontpreview.fontbanner*), 25

M

`module`
`fontpreview.fontbanner`, 24
`fontpreview.fontpage`, 26
`fontpreview.fontpreview`, 23

N

`new_size()` (*fontpreview.fontbanner.FontLogo* method), 25

R

`resize()` (in module *fontpreview.fontbanner*), 26

S

`save()` (*fontpreview.fontbanner.FontWall* method), 26
`save()` (*fontpreview.fontpage.FontBooklet* method), 27
`save()` (*fontpreview.fontpage.FontPage* method), 27
`save()` (*fontpreview.fontpreview.FontPreview* method), 24

`set_body()` (*fontpreview.fontpage.FontPage method*),
28

`set_body()` (*fontpreview.fontpage.FontPageTemplate
method*), 28

`set_font_size()` (*fontpre-
view.fontpreview.FontPreview method*), 24

`set_footer()` (*fontpreview.fontpage.FontPage
method*), 28

`set_footer()` (*fontpre-
view.fontpage.FontPageTemplate method*),
29

`set_header()` (*fontpreview.fontpage.FontPage
method*), 28

`set_header()` (*fontpre-
view.fontpage.FontPageTemplate method*),
29

`set_logo()` (*fontpreview.fontpage.FontPage method*),
28

`set_mode()` (*fontpreview.fontbanner.FontBanner
method*), 25

`set_orientation()` (*fontpre-
view.fontbanner.FontBanner method*), 25

`set_text_position()` (*fontpre-
view.fontpreview.FontPreview method*), 24

`show()` (*fontpreview.fontbanner.FontWall method*), 26

`show()` (*fontpreview.fontpage.FontPage method*), 28

`show()` (*fontpreview.fontpreview.FontPreview method*),
24