

# Guten Tag



Handbook

إضافة علامات • إزالة العلامات

Přidat Značky • Odstranit Značky

Add Tags • Remove Tags

Añadir Etiquetas • Suprimir Etiquetas

Tags hinzufügen • Tags entfernen

Ajouter des Tags • Supprimer les Tags

Aggiungi Tag • Rimuovi Tag

タグを追加 • タグを削除

태그 추가 • 태그 제거

Adicionar Etiquetas • Remover Etiquetas

Добавить Метки • Удалить Метки

Etiket Ekle • Etiketleri Kaldır

添加标签 • 移除标签

新增標籤 • 移除標籤



Guten Tag

© 2021 Florian Pircher

Licensed under Attribution 4.0 International (CC BY 4.0)

OpenType is a registered trademark of Microsoft Corporation.

GitHub is a exclusive trademark registered in the United States by GitHub, Inc.

Mac and macOS are trademarks of Apple Inc., registered in the U.S. and other countries and regions.

Typeset on September 8, 2021.

# Contents

## **1 Installation 4**

- 1.1 Quick Installation 4
- 1.2 Install from the Plugin Manager 4

## **2 Guten Tag 5**

- 2.1 Adding & Removing Tags 5
- 2.2 Batch Editing Tags 6
  - 2.2.1 *Add Tags* 7
  - 2.2.2 *Remove Tags* 7
  - 2.2.3 *Rename Tags* 7
- 2.3 Keyboard Shortcuts 8

## **3 Using Tags 9**

- 3.1 Navigate Between Related Glyphs 9
- 3.2 Tags for OpenType Feature Code 10
- 3.3 Scripting With Tags 12

## **4 Preferences 14**

- 4.1 Glyph Preview Menu 14
- 4.2 Setting Preferences 15
  - 4.2.1 *Using mekkablue's Scripts* 15
  - 4.2.2 *From the Command Line* 15
  - 4.2.3 *From Glyphs' Macro Panel* 16

## **5 Remarks 17**

# 1

# Installation

If the installation still, feel free to reach out to me by opening a new topic on the Glyphs Forum.<sup>1</sup> You can also open an issue on GitHub.<sup>2</sup>

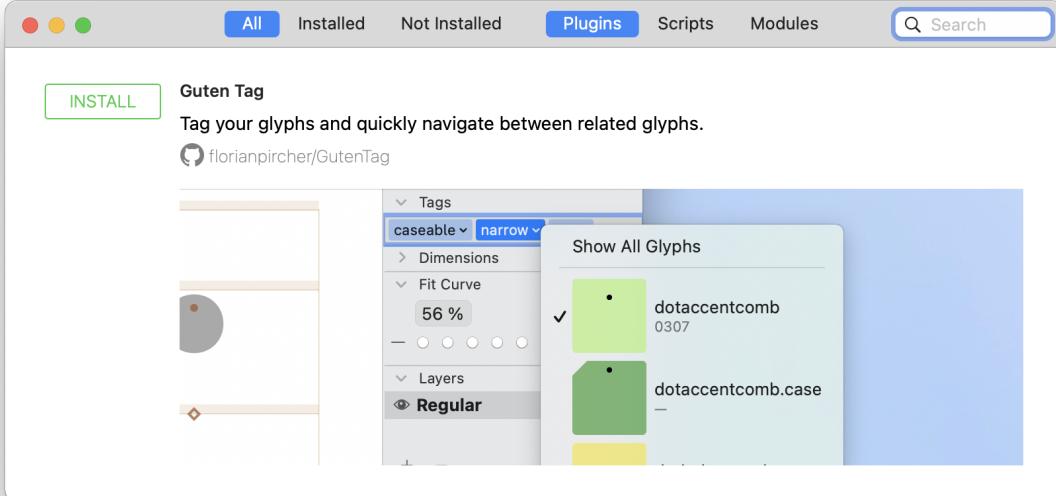
## 1.1 Quick Installation

If you are reading this handbook on the Mac on which you are using Glyphs, simply click on the button below to install Guten Tag:

Install Guten Tag in Glyphs

## 1.2 Install from the Plugin Manager

In Glyphs, open the Plugin Manager by selecting *Window → Plugin Manager → Plugins*. Search for “Guten Tag” and click *Install* next to the plugin preview. Relaunch Glyphs for the plugin to be loaded.



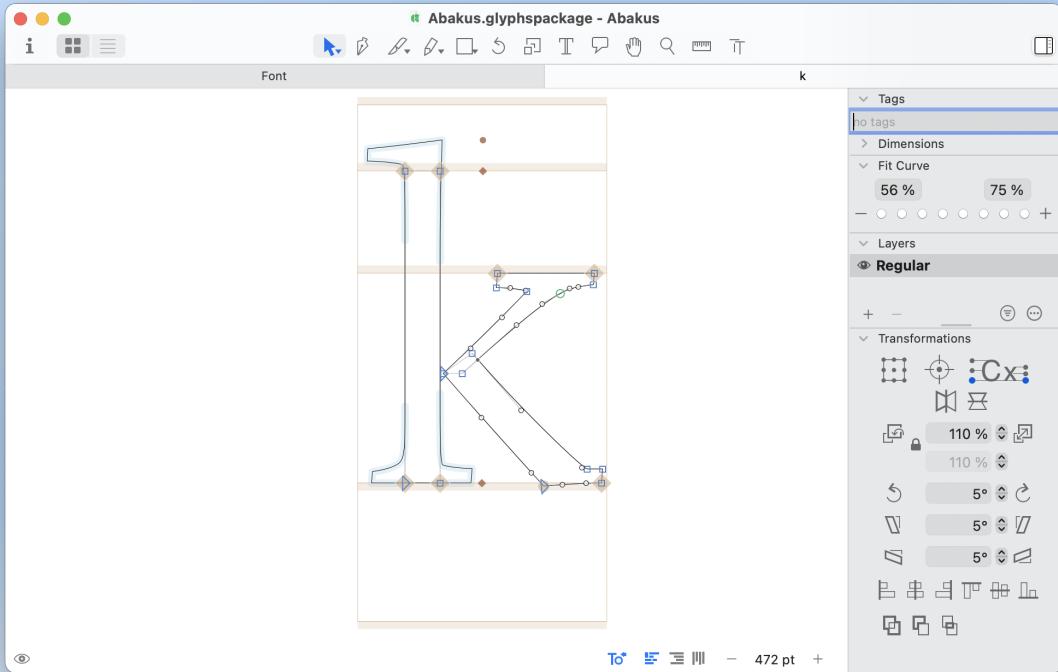
1 <https://forum.glyphsapp.com>, tag me with @florianpircher so I see your post

2 <https://github.com/florianpircher/GutenTag>

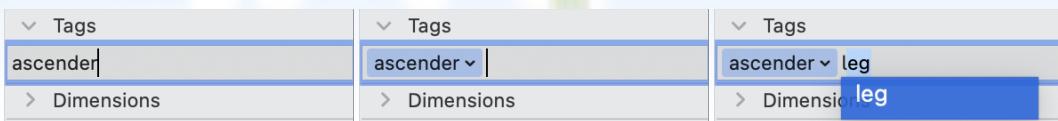
# 2

## Guten Tag

Guten Tag adds a *Tags* palette to the top right corner of your Glyphs window. There, you can edit the tags for the currently selected glyphs.

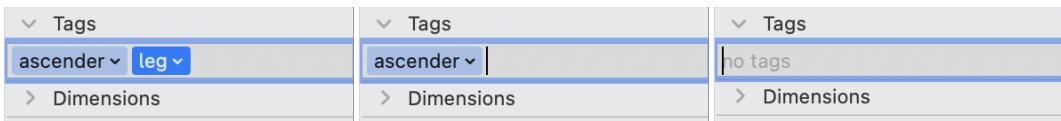


### 2.1 Adding & Removing Tags



Click on the tags field and start typing the name of a tag. Tags can include letters, numbers, spaces, and other punctuation marks. Only the comma (,) is special. Typing a comma or pressing Return will add the tag and display its name in a blue token. An autocomplete menu pops up when typing a tag that is already used within the font. Accept a suggestion by pressing Return.

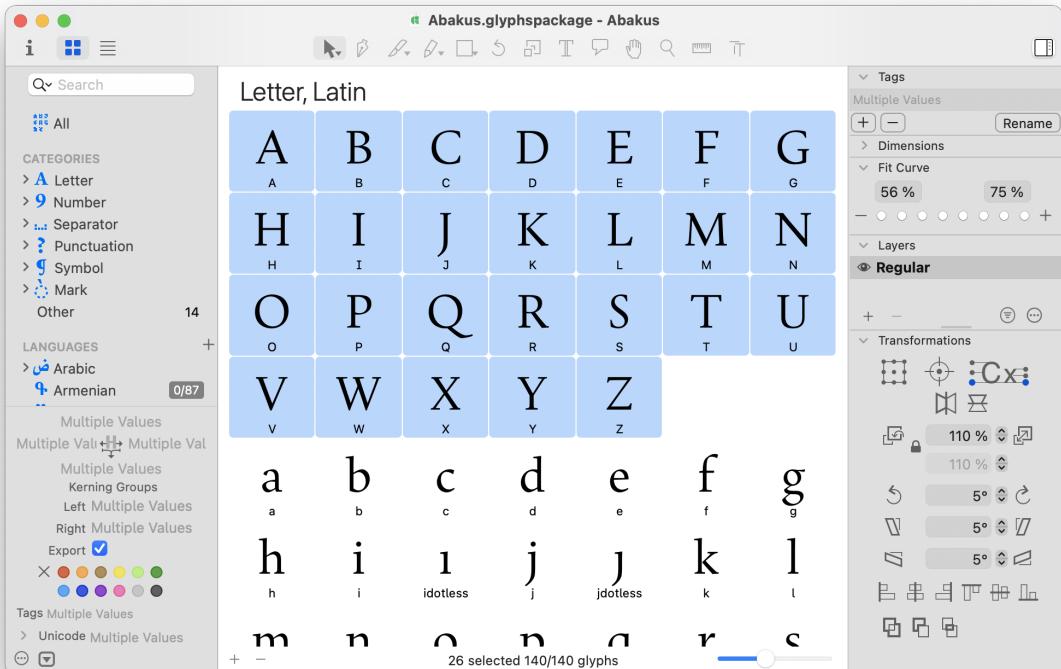
## 2 Guten Tag



You can edit tags like normal text: drag the mouse cursor to select tags or use keyboard commands like  $\text{Shift} + \leftarrow$  or  $\text{Shift} + \rightarrow$ . Selected tags appear with a white font on a dark blue background. Delete tags by pressing the Delete  $\text{Delete}$  key.

### 2.2 Batch Editing Tags

Editing tags from the tags field is possible when you have only one glyph selected or all selected glyphs have the same tags. If, however, the tags of the selected glyphs are different, then Guten Tag can no longer show a unified tags field. Instead, “Multiple Values” is displayed in the tags field and an additional row of controls appears at the bottom of the Guten Tag palette.

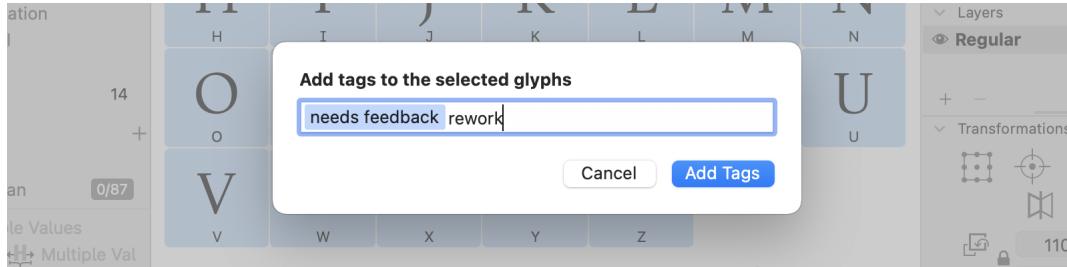


Typing into a multiple-values field will overwrite the tags for all selected glyphs. If you want to change tags individually without modifying any of the other tags, use the three buttons below the tags field.

## 2 Guten Tag

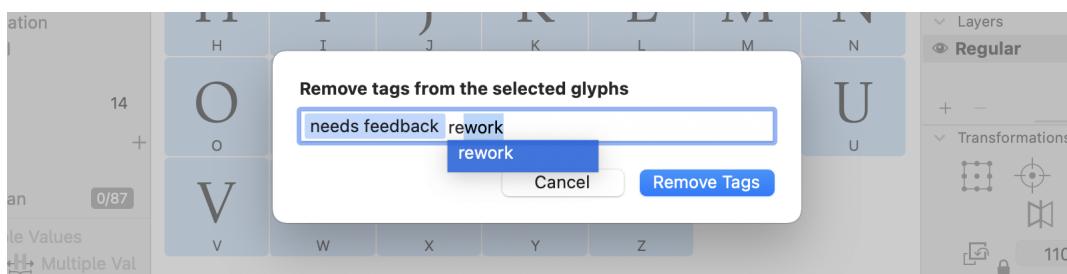
### 2.2.1 Add Tags

Pressing the Plus **[+]** button allows you to add tags to the currently selected glyphs. All existing tags of the selected glyphs will be maintained.



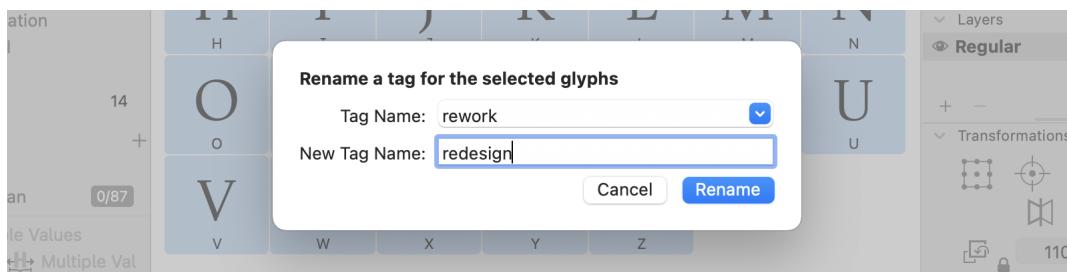
### 2.2.2 Remove Tags

Removing tags works similarly. Press the Minus **[−]** button and enter the tags that you want to remove from all selected glyphs.



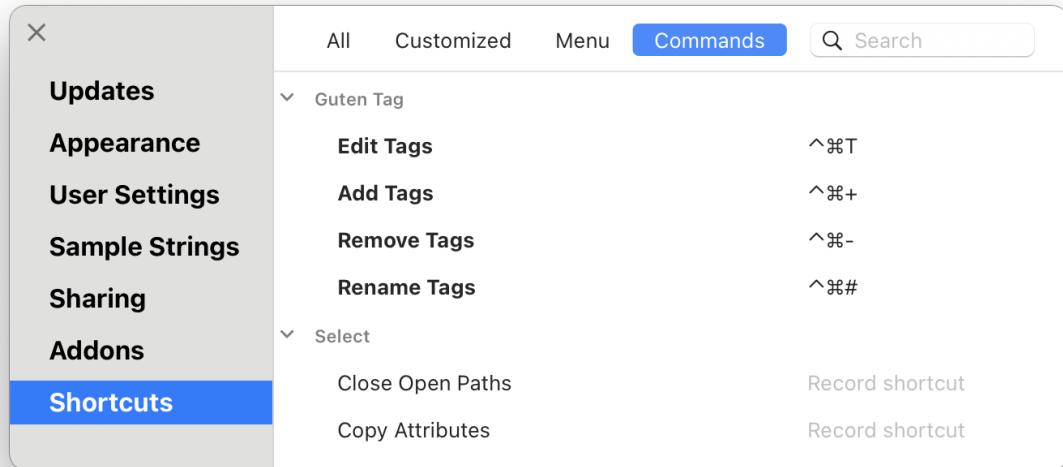
### 2.2.3 Rename Tags

You can also batch rename tags by pressing *Rename*. Select the tag that you want to rename and enter its new name. Merge two tags by renaming one tag to the name of the other.



### 2.3 Keyboard Shortcuts

Guten Tag offers keyboard shortcuts for selecting the tags field in the palette and for performing batch editing on all selected glyphs. Configure these shortcuts in *Glyphs → Preferences... → Shortcuts → Commands*.



*Edit Tags* selects the tags field in the palette. Press Escape or Return to unselect the field. *Add Tags*, *Remove Tags*, and *Rename Tags* perform the batch editing operations on all selected glyphs as described in the previous section (page 6).

Click *Record shortcut* next to a command name and press a combination of modifier keys – Command (⌘), Control (^), Shift (⇧), and Option (⌥) – and a normal (non-modifier) key. Press this keyboard shortcut again when working in edit view to perform the operation.

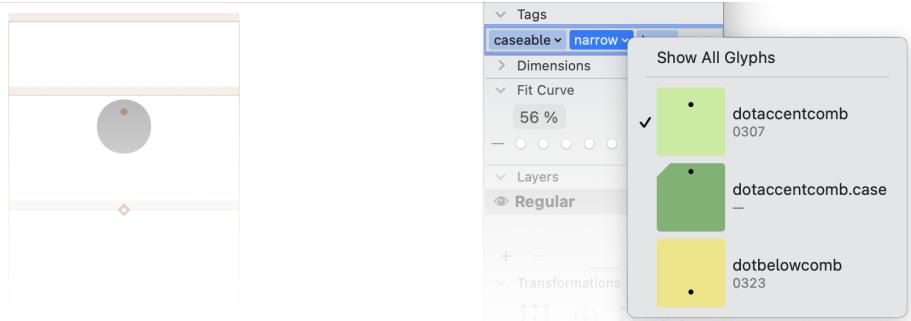
# 3

# Using Tags

In the previous chapter, you have learned how to add and remove tags to and from glyphs. This chapter introduces workflows for which tags can be helpful.

## 3.1 Navigate Between Related Glyphs

Each tag has a small disclosure chevron next to its name. Click on the chevron to open a menu containing all glyphs with the tag. You can also Control-click anywhere on the tag to access the menu.



The menu presents all glyphs belonging to the tag with their names, images, Unicode values, and colors. Select a glyph from the menu to open it in the edit view. If you are already in the edit view, the currently edited glyph gets replaced by the selected glyph.

A checkmark next to a glyph indicates the current glyph. When multiple glyphs are selected, each selected glyph is marked by a horizontal line instead.

Selecting *Show All Glyphs* at the top of the menu opens all glyphs with the selected tag in a new edit view tab.

The size of the glyph images and the number of glyphs to display in the menu can be adjusted. See section 4.1 on page 14 for details.

Glyph navigation using tags can be useful for a variety of cases. You can tag glyphs by their shape (for example *round* or *square*), by their proportions (*narrow*, *tabular*, ...), by their placement (*top*, *bottom*, *overlay*, ...), relation to other glyphs (*caseable/case*, *ligating*, ...), or design features (*tail*, *leg*, *crossbar*, ...).

### 3 Using Tags

This way, when you make changes to one glyph, you can quickly jump to the other glyphs with the same tags and apply the change there, too.

#### 3.2 Tags for OpenType Feature Code

You can use tags in your feature code to create glyph classes. The two expressions `$["some-tag" in tags]` and `$[tags contains "some-tag"]` are equivalent: They expand into a space-separated list of glyph names for all glyphs with the tag *some-tag*. See the Glyphs tutorial on tokens<sup>1</sup> for a general introduction to the `$[...]` notation. In case of tags, the token

```
$["narrow" in tags]
```

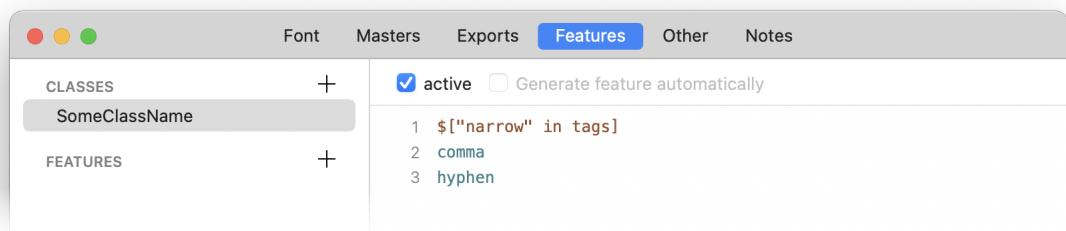
expands to

```
dotaccentcomb dotaccentcomb.case dotbelowcomb
```

if the glyphs *dotaccentcomb*, *dotaccentcomb.case*, & *dotbelowcomb* have the tag *narrow*.

Tokens allow you to describe what kind of glyphs you want in your glyphs classes. The `$["..." in tags]` expression shown above collects all glyphs for a given tag. When you add or remove tags using Guten Tag, you don't need to update your feature code. All glyphs classes defined using tokens are updated automatically on export by Glyphs.

A glyphs class can be defined from the sidebar of the feature code editor or inline with other feature code. If you create glyph classes in the sidebar, place the `$[...]` token directly into the code editor:



Otherwise, wrap the token in square brackets and assign it to a @Class.

```
@Narrow = [$["narrow" in tags]];
```

<sup>1</sup> <https://glyphsapp.com/learn/tokens>

### 3 Using Tags

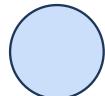
You can combine tags with other predicates:

```
@NarrowMarks = [$["narrow" in tags and category == "Mark"]];  
@NarrowCTA = [$["narrow" in tags and  
name in class(CombiningTopAccents)]];
```

Glyphs tokens can be used to perform set arithmetic on tags. All basic set operations are presented in the following examples for the two tags A and B.

*Identity* All glyphs that have the tag A.

```
$["A" in tags]
```



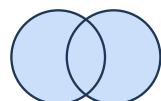
*Complement* All glyphs that do not have the tag A.

```
$[not "A" in tags]
```



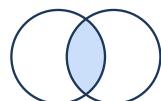
*Union* All glyphs that have the tag A or the tag B (or both).

```
$["A" in tags or "B" in tags]
```



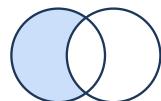
*Intersection* All glyphs that have both the tag A and the tag B.

```
$["A" in tags and "B" in tags]
```



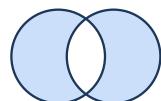
*Difference* All glyphs that have the tag A but not the tag B.

```
$["A" in tags and not "B" in tags]
```



*Symmetric difference* All glyphs that either have the tag A or the tag B, but not both. The symmetric difference can be expressed in two ways.

```
# union of differences:  
$[( "A" in tags and not "B" in tags) or  
  (not "A" in tags and "B" in tags)]  
# one and only one:  
$[ ("A" in tags or "B" in tags) and  
  not ("A" in tags and "B" in tags)]
```



### 3 Using Tags

All examples above can be combined with all other token predicates: Whether a glyph has components (`hasComponents == true`), how many tags it has (`countOfTags > 5`), if the name contains a suffix (name like `"*.alt"`), how high an anchor of the glyph is (`layer0.anchors.top.y > 580`), and many more. Again, read the Glyphs tutorial on tokens to see all possibilities.

#### 3.3 Scripting With Tags

Tags are also useful for scripting. The Glyphs API exposes the `tags` property<sup>2</sup> on glyphs. This property is wrapped by the Python Scripting API starting from Glyphs 3.0.3. In Python, you can iterate over the tags of a glyph like so:

```
for tag in someGlyph.tags:  
    print(tag)
```



Set tags using the `setTags`: (in Python `setTags_`) method:

```
someGlyph.setTags_(["caseable", "narrow", "top"])
```

There is also a `countOfTags` property as well as a variety of accessor methods:

`addTag`: Adds a tag to a glyph.

`removeObjectFromTags`: Removes a tag from a glyph.

<sup>2</sup> <https://docu.glyphsapp.com/Core/Classes/GSGlyph.html#/api/name/tags>

### 3 Using Tags

`removeObjectFromTagsAtIndex:` Removes a tag by its index.

`indexOfObjectInTags:` Returns the index of the given tag.

`objectInTagsAtIndex:` Returns the tag for the given index.

Use the `allTags` method on a font to get the tags for all glyphs of that font.

```
for tag in someFont.allTags():
    print(tag)
```

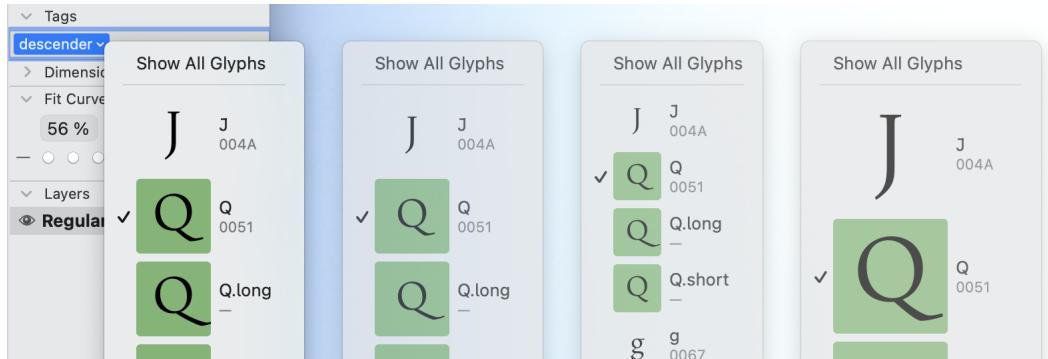
Python method names use an underscore (\_) instead of a colon (:). You can find the full documentation at <https://docu.glyphsapp.com/Core/index.html>

# 4

# Preferences

Preferences allow you to customize the behavior and appearance of Guten Tag. All preference names are prefixed by “`com.FlorianPircher.GutenTag.`”. See section 4.2 for instructions on how to set preference values.

## 4.1 Glyph Preview Menu



The glyph preview menu shows small images for each glyph belonging to the selected tag. The attributes of the menu can be customized as follows.

`GlyphPreviewSize` Defines the width and height of the preview images. The default value is 56 points, but you could make it much larger like 144 points, or much smaller like 18 points.

`GlyphPreviewInset` Controls the size of the glyph within the image. The value is applied at the top and bottom such that the font-size of the glyph is  $\text{GlyphPreviewSize} - 2 \times \text{GlyphPreviewInset}$ . Increase the value if your glyphs have particularly extensive ascenders and descenders; decrease the value if you want to see your glyphs in more detail. The default value is 6 points.

`MaximumGlyphPreviewCount` Limits the number of glyphs in the glyph preview menu. This is useful because creating glyph previews can be im-

## 4 Preferences

possibly slow if there are many glyphs for a tag. The limit can be lifted by setting it to -1. By default, a limit of 1000 glyphs is used.

### 4.2 Setting Preferences

There are three ways of setting preferences. If you have the mekkablue scripts installed, use the approach shown in section 4.2.1. If you are accustomed to the command line, use the approach shown in section 4.2.2. Otherwise, use Glyphs' Macro panel (*Window → Macro Panel*) as shown in section 4.2.3.

#### 4.2.1 Using mekkablue's Scripts

Select *Script → mekkablue → App → Set Hidden App Preferences*. Enter the preference key (prefixed by `com.FlorianPircher.GutenTag.`) and the value you want to set. Confirm by pressing *Apply*.



Press *Reset* instead to reset the preference to its default value.

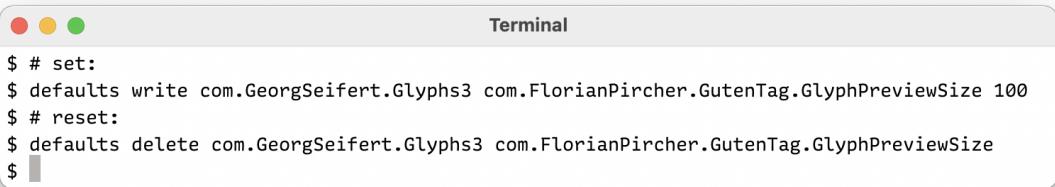
#### 4.2.2 From the Command Line

Run the following command with `$KEY` and `$VALUE` substituted.

```
defaults write com.GeorgSeifert.Glyphs3 \
    com.FlorianPircher.GutenTag.$KEY $VALUE
```

For example, to set `GlyphPreviewSize` to 100:

## 4 Preferences



```
$ # set:  
$ defaults write com.GeorgSeifert.Glyphs3 com.FlorianPircher.GutenTag.GlyphPreviewSize 100  
$ # reset:  
$ defaults delete com.GeorgSeifert.Glyphs3 com.FlorianPircher.GutenTag.GlyphPreviewSize  
$
```

Run `defaults delete ...` without `$VALUE` to reset the preference to its original value.

### 4.2.3 From Glyphs' Macro Panel

Run the following line with `$KEY` and `$VALUE` substituted.

`Glyphs.defaults["com.FlorianPircher.GutenTag.$KEY"] = $VALUE`

For example, to set `GlyphPreviewSize` to 100:



# Remarks

5

If Guten Tag does not work as it should or is missing a helpful feature, do not hesitate to contact me. This includes incorrect translations, typos in this handbook, or any other related issue.

Email [florian@addpixel.net](mailto:florian@addpixel.net)

GitHub <https://github.com/florianpircher/GutenTag>

This handbook was typeset in Würzburg, Germany with `LuaLaTeX` in *Kaius* by Lisa Fischbach and *Codelia* by Toshi Omagari, both of whom generously made adjustments to their fine work to accomodate the needs of this handbook.

Additional fonts in use are *Lyon Arabic* by Khajag Apelian, Wael Morcos, and Kai Bernau; *Kaiti* by Zhang Jiasheng, Zhou Huanbin, and Chen Lütan; and *Choieongho* by Joachim Müller-Lancé, Ku Moa, and Choi Jeongho.

Special thanks to Georg Seifert for reviewing early versions of the plugin code, providing insightful comments on the internals of the plugin, and extending the Glyphs API to make working with tags more convenient.



GUTE NACHT