

前言

同文输入法的内容很多，不可能仅靠一个说明文件就达到精通的地步。但是用一个实例来讲一个示范，我相信对入门一定是很有帮助的，此实例就是小鹤音形 app。不过用实例来讲解也带来一个缺点：你以后用同文创作的作品可能在很长一段时间内都会受到这个实例的严重影响、或者说思维上的束缚。

第一章先让大家简要了解一下同文输入法的强大功能，由此引入第二章之基于同文输入法制作而成的小鹤音形 app，然后重点给大家展示一下内部的主题文件。知道并了解了主题文件后，你就会知道要改哪些地方才能实现按键的调整、皮肤颜色的修改。比如一辆车，我送给你，你问：“车颜色怎么改？车门我想改成在底盘处打开，该怎么做？车高度太低了怎么调高？有没有这样的选项，我点一下“增加”，车门高度就会增加；我点一下“红色”，车身就会变成红色。很可惜，这辆车没有这样的按钮来给你调，也就是说，很遗憾目前你还不能通过按下某个叫“车高”的按钮来改变车的高度。你只能通过调整车的内部结构来调整车的高度。为了让你知道怎能调整这些参数，我就要介绍一下车的内部大概有哪些结构，再着重告诉你应该调整哪一个具体的结构才能实现改变车的身高。以上两章是的排板是根据以前我自己改皮肤、改按键时自己的学习顺序来排的，如对本文有任何意见或建议，欢迎联系。第三章是如何把同文输入法打包成自己的输入法，成为一个单独的 app。

你可以直奔教程部分，但我仍然建议你按着顺序看。这个介绍的目的是为了能让只是单纯想改几个按键的颜色、调整一下按键位置的同文新手会简单地改按键布局、改皮肤颜色。我尽量简单地写。

最后，希望看完之后的你能顺利地入同文的门。

目录

一、简介同文输入法（了解）	1
二、简介小鹤音形 app（了解）	4
2.1 内部文件 trime.yaml（了解）	5
2.2 更改 trime.yaml 以更改按键、皮肤颜色（教程）	5
2.2.1 改 Enter 键和 a 键的位置	
2.2.2 改变单个按键的颜色	
2.2.3 改变整体按键的颜色布局	
2.2.4 增加一个自己的副键盘	
2.3 提一下 schema.yaml 这个文件（了解）	5
三、让同文输入法变成自己的输入法 app.....	4
3.1Mt 管理器简介（了解）	5
3.2 用 Mt 管理器打包输入法（教程）	5
四、参考链接	

以上目录都可以直接点，另外，为避免混乱，在看到 2.1 之前都请不要点击那些有下划线的字（除了“返回目录”这个选项）

一、简介同文输入法

同文输入法其实是一个输入法平台。

在这个平台上你可以制作各种各样的输入法方案，如全拼、五笔、双拼、注音、真码、仓颉，还有各地的方言输入法、外语输入法，甚至外语、方言。总之灵活性非常大。在这个平台上，你可以制作属于自己的键盘布局，26 键、36 键、40 键、1 千键甚至 1 键布局都能实现。

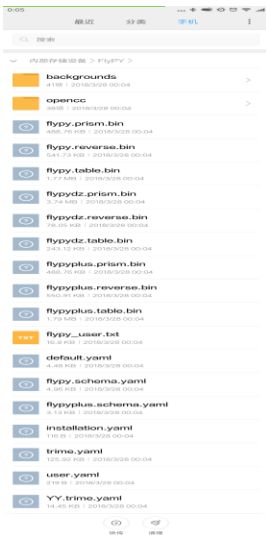
在这个平台上，你也可以定制属于自己的完整的输入法，并且功能可以做到几乎和大众的输入法一样，你要是愿意，甚至可以制作一个和百度输入法看起来几乎一样的界面、按键、皮肤配色，并且实现其大部分功能（为什么是几乎？因为有的功能不一样。比如云联想、页面的滚动，但同文不能云联想，只能做到本地联想.....），另外同文输入法有的功能是百度、搜狗所没有的、是同文独创的，例如连算。同文的缺点是入门门槛稍微有点高，不适合太多的人 DIY。不过同文的定位就是小众输入法、方言输入法，如此一来这其实也不算是缺点。

想详细了解同文，可加群：480159874，关键词：osfans。[返回目录](#)

二、简介——小鹤音形 app

在此通过[展示小鹤音形 app 内部文件](#)的方式介绍同文输入法。

首先从[这里](#)或小鹤双拼的[官方网盘](#)找到并下载此小鹤音形 app，最后安装完毕之后，选择小鹤音形作为你的输入法，手机里会多出一个叫 [flyPY 的文件夹](#)，里面总共有这么多的文件：（一秒略过，此图只是给你展示其长相，另外如果你找不到这个文件夹，可能是安装不正确）



没错，就只有这么多文件。下面列出来：

文件夹： backgrounds #背景文件夹，内有图片，删除则输入法崩溃,不少小鹤用户无法使用都因此。

文件夹： openc #删除则不能实现繁体⇌简体转换

Bin 文件： flypy.prism.bin # app 执行的二进制文件，也不可删，其实下列所有都不可删

flypy.reverse.bin

flypy.table.bin

flypydz.prism.bin

flypydz.reverse.bin

flypydz.table.bin

flypyplus.prism.bin

flypyplus.reverse.bin

flypyplus.table.bin

用户词库:flypy_user.txt #用户在里面加词，原名为 custom_phrase.txt

Yaml 文件:default.yaml

flypy.schema.yaml

flypyplus.schema.yaml

installation.yaml
trime.yaml
user.yaml
YY.trime.yaml

后缀为 yaml 的文件可改，后缀为 bin 的文件无法编辑。

其实还应该有三个重要文件：

- ① flypy.dict.yaml #小鹤音形码表文件
- ② flypy_plus.dict.yaml #小鹤音形+码表文件
- ③ flypydz.dict.yaml #小鹤音形单字文件——此处为用来长按 z 键反查的反查码表

显然，dict 就是 dictionary 词典的意思，是码表文件。对应的二进制文件是 flypy.xx.bin 和 flypy_plus.xx.bin
bin 是给机器看的，dict 是给人看的，有了 bin,这个 dict 也就不重要了，所以可删掉它。

[返回目录](#)

2.1 内部文件“trime.yaml”

这是改按键、皮肤颜色的重要文件。我们叫它主题文件。

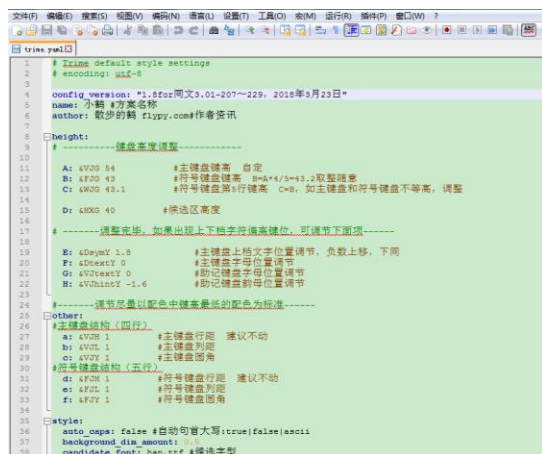
先让读者看看用相应软件打开此文件是个什么样子。

手机端用 Quickedit 这个 app 打开 trime.yaml 文件

里面是这样的：（也一秒略过，比较小）



电脑端用 notepad++打开的样子：（一秒略过）



详细代码如下（虽然被我缩略了，其实仍不详细，毕竟原代码将近四千行……）

下面带有井号键的绿色文字，是小鹤注释；

红色斜体的并且带有井号的是我写的注释，帮助大家进一步理解。

可以看到很多被绿色井号键、分隔线分隔的模块。

```
# Trime default style settings
# encoding: utf-8

config_version: "1.8for 同文 3.01-207~229, 2018 年 3 月 23 日" #版本信息
name: 小鹤 #方案名称
author: 散步的鹤 flypy.com #作者资讯

height:
# -----键盘高度调整-----# 下列内容，你只用改数字，然后部署一下同文输入法就能实现效果。部署方法有很多，点击 app 选择部署、上滑输入法主键盘界面的 Enter 键再选择一次对应主题。

A: &VJG 54 #主键盘键高，前面的字母 ABCDEFG……可先不用管，不是重点。
B: &FJG 43 #符号键盘键高 B=A*4/5=43.2 取整随意
C: &WJG 43.1 #符号键盘第 5 行键高 C≈B，如主键盘和符号键盘不等高，调整

D: &HXG 40 #候选区高度，即候选栏的高度，显示候选字的区域的高度，最上排一行的高度
D: &TG xx #新设置的键盘的某按键高度一
D: &WJGB xx #新设置的键盘的某按键高度二
# -----调整完毕，如果出现上下档字符偏离键位，可调节下面项-----

E: &DsymY 1.8 #主键盘上档文字位置调节，负数上移，下同
F: &DtextY 0 #主键盘字母位置调节
G: &VJtextY 0 #助记键盘字母位置调节
H: &VJhintY -1.6 #助记键盘韵母位置调节

#-----调节尽量以配色中键高最低的配色为标准-----
other:
#主键盘结构（四行）
a: &VJH 1 #主键盘行距 建议不动
b: &VJL 1 #主键盘列距
c: &VJY 1 #主键盘圆角
#符号键盘结构（五行）
d: &FJH 1 #符号键盘行距 建议不动
e: &FJL 1 #符号键盘列距
f: &FJY 1 #符号键盘圆角

如果读不懂这些词语，什么符号键盘行距、列距，你可以逐个改动来了解变化。改动之后，保存→上滑一下 Enter 键（或点击 app 里面的“部署”选项）即可见到变化
```

[返回目录](#)

读完了，读者可能会一脸茫然，除了看得懂几个英文字母 config_version、height、name、other 之外，这个 abdd 是什么，冒号后面又是什么，为什么会有&……（但目前这些都不是重点，重点是如果你改且仅想改小鹤音形键盘的键盘高度、按键距离、按键上的文字的距离，只看到这里就够了，不用再往下看。）

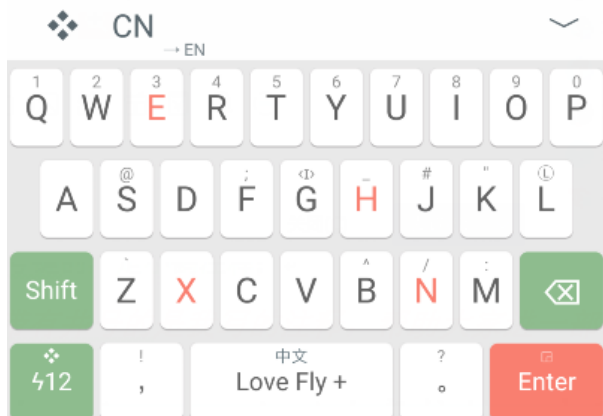
那么请打开下载好的小鹤音形输入法（如果你没学过小鹤音形，请做好暂时打不了字的准备）
我们结合 **app** 就好理解这些代码到底对应什么内容了。（具体怎么对应的？我也不知道，这涉及到软件知识了，我不懂这个）
注意一边看你输入法界面，一边对比下图。

└	小小目录	┐
	主键盘	
	副键盘	
	符号键盘	
	应用键盘	
	菜单键盘	
└	补丁	┐

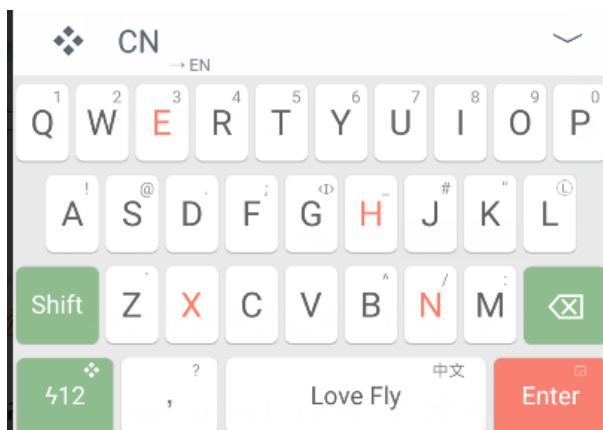
打开输入法，上滑一下 **Enter** 键，你看到两个主题——霓裳、小鹤主题（[萌新主题](#)是什么？不重要，这只是我自己更改的一个主题，文章最后会有展示）



霓裳主题的[主键盘](#)如下：



小鹤主题的主键盘如下：



两者的区别仅在于一个可方便输入句号，一个不行。

空格键的 *Love Fly* 是输入法名称（嗯？不是小鹤音形嘛，怎么叫 *Love Fly* 了——这确实是小鹤音形，只是作者让空格处不显示“小鹤音形”这四个字。他也可以让空格处完全空白，总之这完全取决于作者。事实上，你所看到的键盘上的所有字、按键的位置、颜色都可以改变，你甚至可以换成你喜欢的图片为按键背景颜色）（还有 *xehn* 这次个按键上的字母怎么是红色？——因为 *xn*(*xiao* 小) *he*(*he* 鹤)，这是双拼方案嘛，作者自定义的）

副键盘是除了主键盘之外的键盘，主键盘是你一打开输入界面首先弹出来的键盘。另外，上面两张图其实都是主键盘，每一个主题都有一个主键盘。所以我才说“霓裳主题的主键盘如下、小鹤主题的主键盘如下”。

符号键盘如下——从主键盘界面点击右下角的“412 键”即可进入



有数字、英文、中文、颜文字、emoji、特殊、数学…这些小键盘就是符号键盘，其实也是副键盘的一种

应用键盘如下——从主键盘界面往下滑一下 L 键即可进入



点击之，可以跳转到相应的软件、链接、主页——其实应用键盘也是一个副键盘
(嗯？英文小键盘旁边的数字小键盘变成了“应用”键盘？这个我也不知道，不过不重要，我们继续往下看吧)

菜单键盘如下——从主键盘界面点击其候选栏处的  这个图标即可进入



此 app 内的菜单键盘是以补丁形式存在的副键盘。(补丁是什么？不告诉你，因为这不是重点，我们继续)
由此也可看出主键盘和副键盘的一个特点——即一切都可以从主键盘出发。
[返回目录](#)

下面再给大家看看 trime.yaml 的整体结构

[1234 再来一次:](#)

绿色的并且带有井号键的，是小鹤注释；

红色斜体的并且带有井号的是我写的注释，帮助大家进一部理解。

```
# Trime default style settings
# encoding: utf-8

config_version: "1.5for 同文 3.01-207 2017-09-18"
name: 萌新主题 #方案名称

author: 散步的鹤 flypy.com #若有重大改动，这里可写上你的名字。

height:
# -----键盘高度调整开始-----#
.....更多详细内容略.....
```



```
#调整结束-----#

# -----调整完毕， 如果出现上下档字符偏离键位，可调节下面项-----
.....更多详细内容略.....
#调整结束---#

#-----调节尽量以配色中键高最低的配色为标准-----#
other:
#主键盘结构（四行）
.....更多详细内容略.....
#符号键盘结构（五行）
.....更多详细内容略.....
#调整结束-----#
#---调整键盘风格开始-----#
style:
    auto_caps: false #自动句首大写:true|false|ascii
    background_dim_amount: 0.5
    candidate_font: han.ttf #候选字型

keyboards: [.default, number, .....symbolsD2] #键盘配置，给你定义的新键盘上户口。

增加键盘的时候必须要在这一行写上你增加的键盘的名字

.....更多详细内容略.....
#调整键盘风格结束-----#

#-----后备背景颜色开始-----#当你没有定义（不管是忘了还是故意没定义）键盘颜色时系统默认
会从这里选取颜色。
fallback_colors:
    candidate_text_color: text_color
    comment_text_color: candidate_text_color
    border_color: back_color
.....更多详细内容略.....
#调整后备背景颜色结束-----#

# -----配色方案开始-----# #皮肤背景、颜色在这里写。

#配色方案
preset_color_schemes:
    default:
        back_color: 0xffffefefe #候选区背景状态栏
        border_color: 0xffe9e9e9 #悬浮边框
        candidate_separator_color: 0x00000000 #候选分割背景
        zytb: 0x00ffffff #a1左右隐形按键的背景色,ffff 意为透明色
        AAA: XHkey.png #主键背景色
        AAH: XHgnb.png #主键按下时的背景色
        .....更多详细内容略.....

# -----配色结束-----# #结束
```

```
# -----所有键盘布局开始-----##在这里
```

设计你的键盘布局、对按键位置、功能、背景进行定义。同文的键盘按键只是简单的堆砌，从上到下，定义一个键，往下再定义一个键，它就是一个一个地堆砌，像砌砖一样，砖块的大小由你掌控、顺序也由你掌控，但你不能掌控键使之同时占两行甚至更多行。如果硬要一个键占两行，你只能设计两个相同按键，然后配上同样的背景并让按键间隔为零使之看起来像只有一个按键，听起来有点复杂，我们还是往下看吧。

```
#默认键盘布局开始-----
```

```
preset_keyboards: #预设键盘
```

```
    default: #最重要的主键盘在这里
```

```
keys:
```

```
    # 第一行
```

```
    - {click: q, long_click: 1, swipe_up: 1, key_back_color: AAA, hilited_key_back_color: AAH, swipe_down: K_symbols03, hint: ↓应用}
```

```
    - {click: w, long_click: 2, swipe_up: 2, key_back_color: BBB, hilited_key_back_color: BBH, swipe_down: K_symbolsYDS, hint: 必读}
```

```
    .....更多详细内容略.....
```

```
#助记副键盘布局开始-----
```

```
    defaultVJ:
```

```
        __include: /preset_keyboards/default #看到这个横杠没有？这个就是补丁头。
```

```
        __patch: #这个当然是补丁的开始。
```

```
        .....略...
```

```
# -----密码 letter 键盘布局开始-#和上面主键盘的编写模式一样，从略。
```

```
    .....更多详细内容略.....
```

```
#end 密码键盘
```

```
# -----数字键盘开始-----
```

```
    number:
```

```
    .....更多详细内容略.....
```

```
# -----菜单键盘-----
```

```
    menu:
```

```
# -----菜单键盘补丁-----#这里的补丁是实现翻页功能，是为了在上一个 menu 键盘的，
```

也就是菜单键盘的基础上通过滑动、点击翻页等操作来实现跳转到这个 menux 键盘，而且表面上看起来就是在翻页。总之一页就是一个键盘，你每定义一页就是定义一个键盘。要实现页与页之间的翻页操作，你就要设置一个键盘跳转命令。

```
    menux:
```

```
        __include: /preset_keyboards/menu
```

```
        __patch:
```

```
# -----符号小键盘布局-----  
# 数字小键盘  
symbols01:  
  
# 功能小键盘  
symbols02:  
  
# 应用小键盘  
symbols03:  
  
# 英文符号键盘 # 这个英文键盘也有很多页，就是很多键盘（英文键盘.....有很多键盘？嗯，是有语病，  
但是.....但是.....但是不重要）、从 symbols11 翻到到 symbols14 共四个键盘，即四页  
  
symbols11:  
symbols12:  
symbols13:  
symbols14:  
  
# 中文符号键盘 # 这个中文键盘有两页，即两个键盘  
  
symbols21:  
symbols22:  
  
# 颜文字符号键盘 # 颜文字符号键盘有六页，即六个键盘  
  
symbols31:  
symbols32:  
symbols33:  
symbols34:  
symbols35:  
symbols36:  
  
# Emoji 小键盘 # 很多页  
  
symbols41:  
.....更多详细内容略.....  
  
# 特殊符号键盘 # 很多页  
  
symbols51:  
.....更多详细内容略.....  
  
# 其他小键盘  
.....更多详细内容略.....  
# 你还可以在下面写上任意的小键盘  
.....然后详细内容先略过，后面讲.....  
# -----所有键盘布局结束-----#
```

-----安卓按键定义开始-----##此处功能

我忘记了，详情请看同文输入法的[wiki 详解](#)，另外，这也不是重点。

android_keys:

name: [VoidSymbol, SOFT_LEFT, SOFT_RIGHT, HOME, BACK, CALL, ENDCALL,更多详细内容略.....

preset_keys:

安卓

BRIGHTNESS_DOWN: {label: 亮度-,}

BRIGHTNESS_UP: {label: 亮度+,}

CALCULATOR: {}

.....更多详细内容略.....

#安卓按键定义结束-----#

rime 编辑开始-----##

按键定义，定义按键的名字和功能（为什么要定义？不急，后面一节讲[增加一个你的副键盘](#)的时候就清楚了）

Shift_L: {label: Shift,}# 中文状态一次锁定，英文状态长按锁定

Shift_L1: {label: Shift,} # 中文状态长按锁定

Shift_L2: {label: ⬆,}# 中文状态长按锁定

.....更多详细内容略.....#

#-rime 编辑结束-----#

rime 组合键定义#这里你可以定义一些组合键，看下面的 BackToPreviousSyllable 这个新定义的按键——如果你设置了一个这样的按键在主键盘，部署使输入法成功写入之后，点击它（label 的意思是标签，就是你会看这个键上面写着‘删音节’三个字，你也可以写成别的），它就会 send（发送）一个组合命令给系统——这个组合命令就是 Control 命令和 BackSpace 命令。

F4: {label: 方案菜单, send: F4}

BackToPreviousSyllable: {label: 删音节,}

CommitRawInput: {label: 编码,}

.....更多详细内容略.....

rime 状态#在这里你可以设置输入法的状态转换按键

Mode_switch: {.....states: [中文, EN]} #看到“中文”和“EN”了没？这就是键盘最上那行的状态转换图标，点一下键盘变成英文键盘，再点一下又变回中文键盘的那个图标。

.....更多详细内容略.....

trime 设置#在这里定义你自己定义的按键的功能

K_default: {label: 默认,}

K_symbols01: {label: "▲",}#symbols01,这就是前面写的符号键盘之一（即其中一页）

K_symbolsF7x: {label: "►", send: Eisu_toggle, select: symbolsF7}

K_symbolsYDS: {label: "必读", send: Eisu_toggle, select: symbolsYD}

.....更多精彩内容略.....

trime 命令#这应该是同文的特点之一，仔细看小鹤音形 app 里的应用键盘，你应该看到“淘宝”这个按键，你也可以定义自己的快速跳转应用，这里我设了个支付宝，但是上面的图中没有体现。

#这里还有一个“日期”的跳转，但是它的选项(option)里并没有可供跳转的包名或者链接，只是十分格式化、古板的 yyyy,m,d——这就只能让它输出日期，yyyy 会自动变成年、m 会自动变成月，d 会自动变成日

#“淘宝”这个命令里就有一个包名，包名是什么？包名是安卓系统（其他系统我没了解过）.....我也不知道是什么，但是我知道你写上包名，并且点击一下这个按键，输入法会发送这个包名给你的手机系统，手机系统接收到这个包名之后就会打开对应的应用。你可以下载一个包名查看器 app，从包名查看器里可查到你想知道的应用的包名，什么？点击这些应用不能成功跳转？原因有二：①包名不对，②你从开机到现在都没有打开过这个应用。另外，这个功能目前其实实现得也不怎么良好，我们还是忽略吧，往下看。本文不讲怎么跳转，读者可自己摸索。

```
Date: {label: "日期",.....option: "yyyy年M月d日"}
Taobao: {label: 淘宝, ....., option: "com.taobao.taobao"}
VFB: {label: 支付宝, .....}
LLQ: {label: ①, send: function, command: web_search, option: "%4$s"} # 直达光标前 http
开头的网址或搜索光标前内容
uwkyd: {label: "上网阅读", send: function, command: run, option: "http://uee.me/FUBx"} #
上网看引导
uwkyd2: {label: "官方示范", send: function, command: run, option: "http://uee.me/Jm6X"}
#上网看引导#
uwkyd3: {label: "关于萌新主题", send: function, command: run, option: "http://uee.me/JmLV"}
#上网看引导#
```

这就是小鹤音形的 trime.yaml 文件的整体结构，或曰小鹤主题文件的整体结构。

[返回目录](#)

2.2 更改 trime.yaml 以改变按键、皮肤颜色

前面说的基本都是在解释，下面就是真正的教程了，下面的过程中有什么不明白的，就再往回再看一下“[下面再给大家看看 trime.yaml 的整体结构](#)”这节内容吧。所谓教程……我想应该是那种按部就班地操作就能达到目的的不需要理解的固定步骤。建议在“动刀”学习此教程之前先用上面建议过的两种软件（手机为 Quikedit，电脑为 notepads++）打开这个 trime.yaml 文件（这个文件在哪里？上文讲过了，在手机的 [flypy 文件夹](#)里）

以主键盘为例

首先

我拿出一个按键来详细讲讲里面每个命令的意思

下面是 L 键的定义

```
- {click: l, long_click: LLQ, swipe_up: LLQ, swipe_down: K_symbols03, key_text_color:
zytb, key_back_color: zytb, key_symbol_color: zytb, hilited_key_text_color: zytb,
hilited_key_back_color: zytb, hilited_key_symbol_color: zytb, width: 5}
```

Click: l——点击字母 L,

long_click: LLQ——长按字母 L, 跳转到 rime 命令“LLQ”,

swipe_down: K_symbols03——下滑 L 键, 跳转到 K_symbols03 这个功能小键盘,

key_text_color: zytb——按键_标签_颜色, 按键上的标签的颜色为 zytb 对应的代码

key_back_color: zytb ——按键_背景_颜色, 按键的背景的颜色为 zytb 对应的代码

key_symbol_color: zytb——按键_符号_颜色, 按键的上标符号的颜色为 zytb 对应的代码

hilited_key_text_color: zytb——按下_按键_颜色, 按键按下时的标签颜色为 zytb 对应的代码

hilited_key_back_color: zytb ——按下_按键_标签_颜色, 不用解释了吧, 很多都是英文单词, 都看得懂😂

hilited_key_symbol_color: zytb ——按下_按键_标签_颜色。

width: 5——按键宽度为 5。放在按键之外, 键盘之内定义, 则为定义整个键盘的所有按键的宽度。放在按键之内, 则为定义单个按键的宽度, 并且能无视整体键盘的按键宽度定义, 即优先于键盘之内、按键之外的那个 width。看不懂? [看下面的代码](#)。一个行的宽度值为 100, 比如你想在某行里放两个键, 且仅放这两个键, 那你可以这样:

```
- {click: 嗯, width: 50}
- {click: 哦, width: 50}
```

50+50 刚好等于 100。如果我设成一个是 50, 一个是 58, 即总和超过 100 呢? 那么它会放在下一行——想象你在砌砖, 砖头不能弯曲, 而空间又只有那么大, 不够放怎么办? 当然是放下一行啦。

```
- {click: Return, .....width: 20} #这个Enter 键里面也有一个width, 这个宽度定义width 就无视了
下面的这个width, 使该Enter 键的宽度为20.
```

```
name: "26 键 2"
```

```
width: 10 #这是按键外的width, 用来定义整个键盘的每一个按键宽度
```

然后是操作:

① 改 Enter 键和 a 键的位置:

翻到第 539 行左右, 看到如下内容

```
keys:
  # 第一行
  - {click: q, .....}
  .....其他按键.....
  #第二行
  - {click: a, .....}
  - {click: a, .....} #按键 a 在这里
  - {click: s, .....}
  .....其他按键.....
  - {click: Return, .....width: 20} #按键Enter 在这里
name: "26 键 2"
width: 10
```

把它们顺序改变一下就行→

```
keys:
  # 第一行
  - {click: q, .....}
  .....其他按键.....
  #第二行
  - {click: a, .....}
  - {click: Return, .....} #按键Enter 在 a 原来的地方了
```

```
- {click: s, .....}  
.....其他按键.....  
- {click: a, .....width: 20} #按键 a 在原来 Enter 键的地方了  
name: "26 键 2"  
width: 10
```

注意，除了键内宽度命令我没换，我把剩余的所有内容都换过来了，就是上面代码中省略的东西全换过来了，并不是简单地只把字母 a 改成 Enter，把 Enter 改成 a。

（你看到有两个 a？是的，就是有两个 a，但是上面的那个 a 键的背景是透明的，这个透明的 a 键就是你能看到的那个 a 键它左边的那个空白区域，透明了，所以你能只看到一个 a，但是点击那个空白区域仍能点出 a 这个字母。L 键同理。

更改完成后，在 app 里点击“部署”、或上滑一下 Enter 键，重新选择一下“小鹤”主题即可完成设置。

② 改变单个按键的颜色：

```
- {click: q, long_click: 1, swipe_up: 1, key_back_color: AAA, hilited_key_back_color: AAH}
```

把里面的

key_back_color: [AAA](#)，和 hilited_key_back_color: [AAH](#)

AAA 改成你想要的颜色的十六进制代码，例如我想要橙色，百度一下该颜色对应的十六进制代码是 FFA400，然后删掉 AAA，写上 0xffFA40，格式如下

```
key_back_color: 0xffFFA40, 这是没按下时的按键背景色  
hilited_key_back_color: 0xffFFA40 这是按下时的按键背景色，显然我这样做的话，按下、不按都会显示橙色了  
0xffFa40 是不区分大小写的，前面的 0x 表示十六进制，0x 后面的两个 f 是透明度，范围是 00 到 ff 注意，是 00——两个零。
```

注意 color: 冒号与 AAA 之间要有空格，否则部署时程序会崩溃！

更改完成后，在 app 里点击“部署”、或上滑一下 Enter 键，重新选择一下“小鹤”主题即可完成设置。

③ 改变整体按键的颜色布局：

翻到 trime.yaml 文件里的这个地方，大约是在第 140 行代码左右。红色的字是我写的

```
#配色方案  
preset_color_schemes: #预设_颜色_方案  
  default: #默认颜色方案  
    author: flypy.com #作者  
    name: "小鹤" #配色名字（或叫皮肤颜色名字）  
    back_color: 0xfffefefe #候选区背景状态栏  
    border_color: 0xffe9e9e9 #悬浮边框  
    candidate_separator_color: 0x00000000 #候选分割背景  
    candidate_text_color: 0xff5a676e #候选文本  
    comment_text_color: 0xff7b868c #提示  
    hilited_back_color: 0x00000000 #高亮编码背景  
    hilited_candidate_back_color: 0x00000000 #高亮候选背景  
    hilited_candidate_text_color: 0xffff97f70 #高亮候选文本  
    hilited_comment_text_color: 0xff000000 #高亮提示  
    hilited_key_back_color: XHkeyG.png #高亮按键背景  
    hilited_key_symbol_color: 0xffffffff #高亮按键符号  
    hilited_key_text_color: 0xffffffff #高亮按键文本
```



```
hilited_off_key_back_color: XHkeyF.png #高亮按键关闭状态背景
hilited_off_key_text_color: 0xff000000 #高亮按键关闭状态文本
hilited_on_key_back_color: XHkeyF.png
hilited_on_key_text_color: 0xff000000
hilited_text_color: 0xff23948e #高亮编码
key_back_color: XHkeyF.png #按键背景
#key_border_color: 0xffeceff1 #按键边框 取消
key_symbol_color: 0xff919191 #按键符号
key_text_color: 0xff616161 #按键文本
keyboard_back_color: 0xffe9e9e9 #键盘背景
label_color: 0xff000000 #标签
off_key_back_color: XHkeyG.png #按键关闭状态背景
off_key_text_color: 0xffffffff #按键关闭状态文本
on_key_back_color: XHkeyG.png
on_key_text_color: 0xffff97f70
preview_back_color: 0xffff97f70 #按键提示背景
preview_text_color: 0xffffffff #按键提示文本
shadow_color: 0xff000000 #按键文字阴影
text_back_color: 0xccffffff #编码区悬浮窗背景
text_color: 0xff5a676e #编码
ABC: XHkey.png #letterfh 键盘主键背景
ABH: XHgnb.png #letterfh 主键高亮背景
AAA: XHkey.png #主键背景
BBB: XHkey.png
CCC: XHkey.png
AAH: XHgnb.png #主键高亮
BBH: XHgnb.png
CCH: XHgnb.png
aaa: XHkeyF.png #符号键盘数字键背景
bbb: XHkeyF.png
ccc: XHkeyF.png
bdb: XHbdb.png #主键标点背景
bdh: XHgnb.png #主键标点键高亮背景
gnb: XHgnb.png #主键功能键背景
gnh: XHkey.png #主键功能键高亮背景
gns: 0xffffffff #功能键上档文字颜色
kgb: XHkgb.png #空格背景
kgh: XHkgbh.png
kgs: 0xff67777f #空格键上档文字颜色
kgt: 0xff616161 #空格键文字颜色
hcb: XHhcb.png #回车键背景色
hch: XHkey.png #回车键高亮背景
hcs: 0xffffffff #回车键上档文字颜色
hct: 0xffffffff #回车键文字颜色
cdt: 0xffff97f70 #符号键盘当前菜单键文字颜色
yyt: 0xffff97f70 #应用文字颜色
fyt: 0xffd3d7da #翻页键止文字颜色
xnhe: 0xffff97f70 #xnhe 键文字颜色
zytb: 0x00ffffff #al 左右键背景色
```



```
zjs: 0xff919191 #助记颜色
xdb: 0xffe9e9e9 #❖选单界面背景色（菜单界面背景色）
```

上面的注释应该写得很清楚了，但是考虑到你是初学者，可能对上述很多词汇的意思感到费解，下面逐一介绍一些可能会难以理解的词汇进行解释：左边是词汇，后边是通俗意思，其实看一下英文单词的意思也能懂它的通俗意思了。

高亮——按下按键的时候

高亮按键关闭状态——按键有两种状态，一种是高亮状态，一种是平常状态（或曰：高亮按键关闭状态），比如小鹤音形的 shift 键就有两种状态，长按它就能进入高亮状态，这时候你输入的字母就会全都是大写字母。再长按一下，就恢复到平常状态，通常只有这个键有高亮、平常两种状态，其他按键都只有平常状态。也就是说，上面的代码里，有高亮一词的，都只作用于 shift 键。

上档——字母的上标签文字，比如 qwertyuiop 这些字母上面的 123456789

下标——字母的下标文字，比如——额，小鹤音形的键盘都没有下标，我的[萌新主题](#)就有

改对应的地方后面的颜色编码就是在改按键的颜色了。

更改完成后，在 app 里点击“部署”、或上滑一下 Enter 键，重新选择一下“小鹤”主题即可完成设置。

另外再重复一遍：0xff 中的 0x 代表十六进制；ff 是透明度，范围是 00 到 ff。背景除了用颜色代码，还可以用图片，图片首先放在 background 文件夹里，名字自己写，后缀不能错。下面介绍一下……好像没什么要介绍的了，改讲的都讲完了。剩下的就让读者以后自己去慢慢探究这个 trime.yaml 文件吧。

④增加一个自己的副键盘（增加主键盘呢？直接改主键盘就好啦，不用增加）

第一步、在第 84 行的按键定义里[上户口](#)

```
keyboards: [.....在这里写上你新增加的键盘的名字，英文，例如 symbolsF7, symbolsYD] #键盘配置
重复一遍，一页就是一个键盘，所以你的键盘有多少页，就要写多少个名字，这里我设置了两页
```

第二步、在键盘区（即在 [所有键盘布局开始-所有键盘布局结束](#) 这段代码区域里找一个合适的位置）写你的新键盘下面是示例

```
symbolsF7:
  ascii_mode: 1
  author: "酥梨"
  keys:
    - {click: "", width: 100,height: *TG, key_back_color: tu7.jpg, hilited_key_back_color:
xdb} #tu7.jpg 是我弄的背景图片的名字，如果你直接复制，是会变成后备背景色的，因为你的 background 文件夹里少了这个叫 tu7.jpg 的图片
#另外，这里的*TG——这就是最开始你看不懂的代码，现在点击它看看吧，看看它跳到哪里。下面的*WJGB 同理。
    - {click: K_symbolsF7X, width: 40}
    - {click: K_symbolsYDS, key_text_color: fyt, width: 40}
    - {click: K_back, width: 20}
  name: "同文小鹤必读 7"
  width: 10

symbolsYD:
  ascii_mode: 1
  author: "酥梨"
  keys:
    - {click: K_symbolsF7X, width: 50,height: *WJGB,key_text_color: 0xff333333,
key_back_color: xdb, hilited_key_back_color: xdb}
    - {click: uwkyd, label: "上网阅读",width: 50,height: *WJGB,key_text_color: 0xff333333,
key_back_color: xdb, hilited_key_back_color: xdb}
    - {click: uwkyd2, hkey_text_color: fyt,width: 40 }
```

```
- {click: "uwkyd3", key_text_color: fyt, width: 40} #这里这个 uwkyd3 有引号，有没有都没关系，据说有的话更保险
- {click: K_back, width: 20}
name: "同文小鹤必读"
width: 10
```

第三步、在 [trime 设置](#) 这个地方，按顺序写上你的按键设置（包括其功能、标签等等），如下代码：

```
K_symbolsF7x: {label: "►", send: Eisu_toggle, select: symbolsF7}
K_symbolsYDS: {label: "必读", send: Eisu_toggle, select: symbolsYD}
```

注意到 symbolsF7x 前面有个 K_，这不是你的键盘的名字，你的键盘名叫 symbolsF7X，这是一个就叫 K_symbolsF7x 的命令操作，请看这里 [K_symbolsF7x](#)，这条命令的意思是，点击它，它就拾起一个叫 symbolsF7 的键盘。

最后一步、在主键盘的某个按键里写上跳转到这个副键盘的命令

```
- {click: w, long_click: 2, swipe_up: 2, key_back_color: BBB, hilited_key_back_color: BBH, swipe_down: K\_symbolsYDS, hint: 必读}
```

我在 w 键上设置了下滑操作跳转到 symbolsYDS 这一页。

。 [返回目录](#)

schema.yaml 这个文件

```
# Rime schema settings
# encoding: utf-8

schema:
  schema_id: flypy # 方案名称，这就是空格处显示的字母
  .....
  dependencies:
    - flypydz # 反查方案是 flydz，fly 单字方案。
    .....
  switches: # 可在状态栏上显示的内容，去掉下面的"#"号可开启
    .....
返回目录
```

Mt 管理器简介

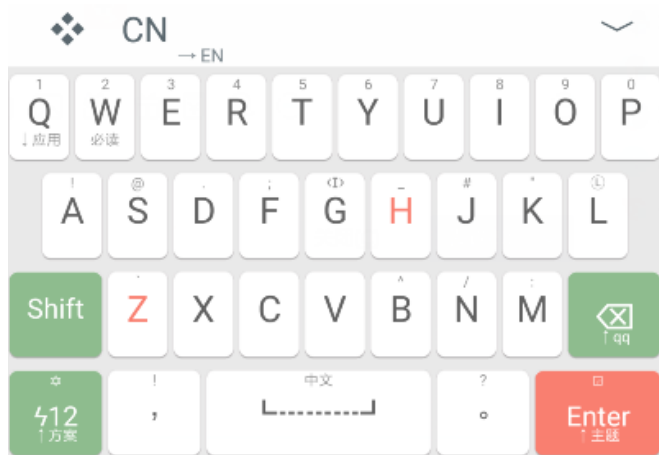
<https://baike.baidu.com/item/MT%E7%AE%A1%E7%90%86%E5%99%A8/19053609?fr=aladdin>

用 Mt 管理器打包输入法

<https://jingyan.baidu.com/article/1e5468f9773b7d484961b7d6.html>

[返回目录](#)

萌新主题：



可以看到 Q 键 W 键等键有下标。

下面介绍一下我的萌新主题（以下两个界面的设计是为了方便在手机上编辑这个 `trime.yaml` 文件）什么？怎么左上角的那个向下的箭头变成了“酥”字？不习惯的话，可以改的，但是就涉及到 `schema.yaml` 这个文件了，不展开讲😂



长按 g 键进入编辑键盘界面，下图：



点击 4 12 键进入数字键盘界面，下图：



编辑键盘的功能：

上下左右移动，跳到行头、跳到行尾、全选、剪切、复制、粘贴、撤回、撤出、制表符 `tab`、删除、退格、回车。

数字键盘功能：

上滑、下滑、左滑、右滑，长按 4 为剪切、长按 8 为剪切、长按 4 为复制、长按 6 为粘贴、长按 5 为选中

[返回目录](#)

最后

下面给大家看一下小鹤双拼输入法的作者对同文的评价

同文有什么吸引到我呢，首先就是次选键，这在之前的百度输入法上是没有的，正好匹配小鹤的最多二重设计，从此打字就不用再去点候选窗中的候选字词了；其次就是其能关闭候选窗的功能，让键盘占用空间减少的同时也让打字更清爽，配合嵌入式编码窗，两个字：完美！

虽然同文目前在我心里有不可替代的地位，甚至认为完美，但还是有些瑕疵总有点膈应，不吐不快，列表住下，以期完善。

同文弱项：截至同文 3.01-110

- ①码表不支持原始序 ☆☆☆☆
- ②用户词库和主词库重码时的排序方法稍复杂 ☆☆
- ③主题内不支持多主键盘 ☆☆☆
- ④主题不能定义在按键上直接切换 ☆☆
- ⑤不支持全码空码时自动清空 ☆☆☆☆

- ⑥不支持指定默认主题 ☆☆☆
- ⑦主辅键盘高度很难调平（怀疑辅键盘最下排的行间距不等高造成） ☆☆☆
- ⑧不能自定扩展字集 ☆☆

散步的鹤

2017 年 8 月 20 日

另外，直至现在（2018 年 4 月 6 日）同文也没有实现对上述弱项的改进。

最新进展：终于写完了，起初写的时候是想用尽量短的语言描述同文输入法，给读者一个大体的框架，但是写着写着……越写越多了（崩溃 😓）可能同文真的不适合大众吧。另外，就在笔者完成这篇介绍没几分钟，同文的作者就更新了同文崩溃 😓 ——内部的结构发生了重大更新。据最近了解，同文输入法的内部文件夹——对应小鹤音形的内部文件夹 FlyPY 结构发生了改变，不再是单纯地只有 background 子文件夹、opencc 子文件夹，而是增加了一个 build 子文件夹，而且部署速度更快。读者可自行尝试。

本人邮箱：zodensu@gmail.com

参考链接

[同文官网](#)

[Rime 说明书](#)

[RIME 新手指南](#)

[trime.yaml 详解](#)

[schema.yaml 详解](#)

[豆泥丸同文简介](#)

[有萌新主题的小鹤音形下载链接](#)

[小鹤双拼作者评价](#)

[返回目录](#)