

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。

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。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。

你可以直接看第二章的教程部分，不过我还是建议你按着顺序看。希望你看完之后可以达到以下目的：改变按键的颜色、调整按键的位置，甚至触类旁通。我尽量简单地写。另外建议你在看的同时不妨一边直接操作，以便更快上手。

目录

一、简介同文输入法	1
二、简介小鹤音形 app	4
2.1 内部文件 trime.yaml	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	
2.2.5 了解寄生主题之高飞.trime.yaml 文件	
2.4 提一下 schema.yaml 这个文件.....	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.....	5
3.2 用 Mt 管理器打包输入法	5
四、参考链接	

一、简介同文输入法

同文输入法其实是一个输入法平台，主平台是 **rime**，条件合适时能 **window, mac, IOS, Android** 全平台通用。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、一千键甚至 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。不过经过人们的努力，目前同文的一个另外版本不仅能本地联想，还能做到云联想了.....），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“拼写连算”功能。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](#)

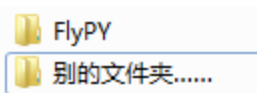
二、简介——小鹤音形 app

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。

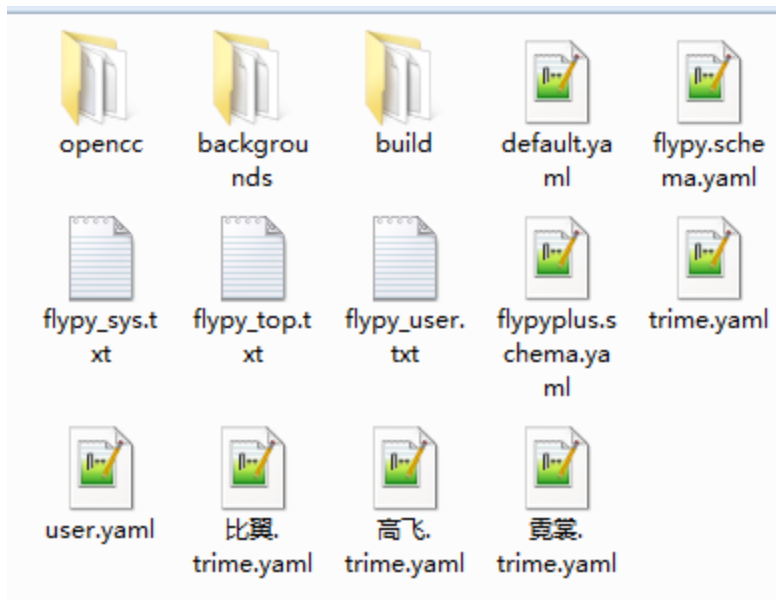
首先从[这里](#)或小鹤双拼的[官方网盘](#)找到并下载此小鹤音形 app，并下载“小鹤音形安卓版安装必读.txt”，



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](#)，



点开之后发现里面总共有这么多的文件：（不必细看，略览即可）



显而易见，里面还有三个子文件夹和至少七个后缀为 `yaml` 的文件以及三个 `txt` 格式的用户词库。
下面展开介绍：

文件夹： `backgrounds` #背景文件夹，内有图片，删除则输入法崩溃，清理手机内存时务必注意保留。

文件夹： `openccc` #删除则不能实现繁体 ⇌ 简体转换

文件夹： `build` #后缀为 `bin` 的文件均存放于此

Bin 文件： `flypy.prism.bin` # `app` 执行的二进制文件，
 `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`、`flypy_sys.txt`、`flypy_top.txt` #使用者可以在这里方便地加词，`user`：用户，`sys`：system，`top` 置顶

Yaml 文件：`default.yaml`
 `flypy.schema.yaml`
 `flypyplus.schema.yaml`
 `installation.yaml`
 `trime.yaml`
 `user.yaml`
 `YY.trime.yaml`

只有后缀为 `yaml` 和 `txt` 的文件可以编辑。`txt` 是用户词库，打开之后会发现自带讲解，所以下面重点讲 `yaml` 文件。
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+方案的码表文件，即开启了隐藏全码的小鹤音形方案的码表文件

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，所以就删掉它了。

[返回目录](#)

2.1 内部文件“trime.yamll”

这是改按键、皮肤颜色的重要文件。我们叫它主题文件。

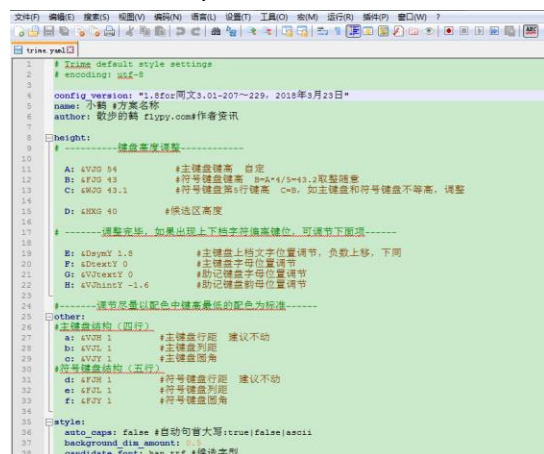
先让读者看看用相应软件打开此文件是个什么样子。

手机端用 Quikedit 这个 app 打开 trime.yamll 文件，电脑端可以用任意可高亮代码的文本编辑器打开，这里用 notepad++ 打开：

里面是这样的：（文字看不见没关系，知道这个文本编辑器能使代码高亮就行）



电脑端用 notepad++ 打开的样子：（同上）



下面展开讲解代码（我甚至不知道这算不算代码，因为着实非常简单，可以说稍微会点英语的人从头看到尾一遍，我想应该都能大致理解里面的内容）

下面带有井号键的绿色文字，是小鹤原注释；

红色斜体的并且带有井号的是我写的注释。

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 #方案名字
author: 散步的鹤 flypy.com #作者资讯
# -----键盘高度调整-----#
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 $C\approx B$ ，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，嗨~我们又见面啦
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

后面的代码暂时从略

[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 abcd 是什么，冒号后面又是什么，为什么会有&，什么是主键盘、副键盘和符号键盘……知道的话很简单，改动一下里面的数字，然后部署一下 app，你就能看到相应界面的变化了，如果看不到变化，那说明暂时是没用的。看不懂没关系，先接着往下看。

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。

└─ yamI 之小目录 ─┐

[主键盘](#)

[副键盘](#)

[符号键盘](#)

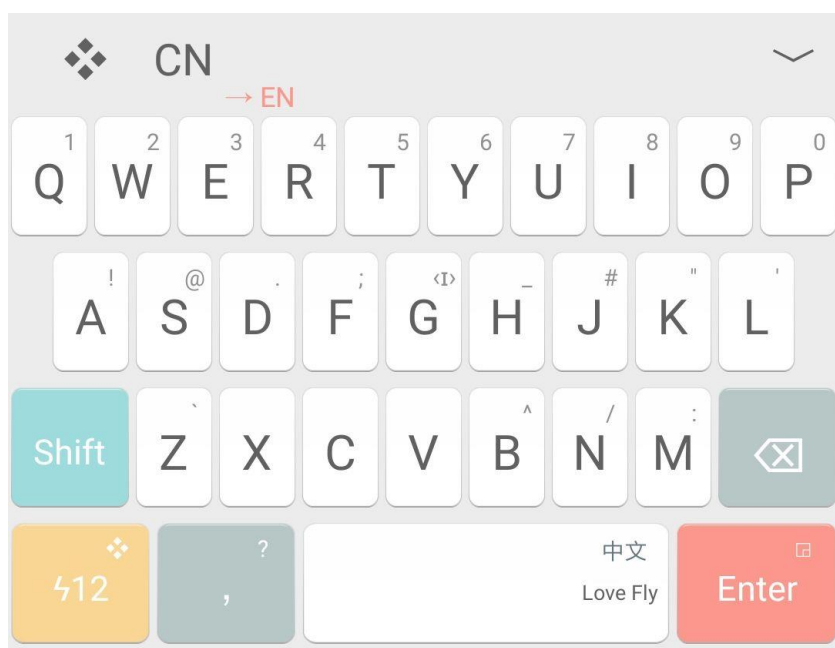
[菜单键盘](#)

└─ [补丁](#) ─┐

注意一边看你输入法界面，一边对比下图。

下面[主键盘](#)

打开输入法，初始界面如下：

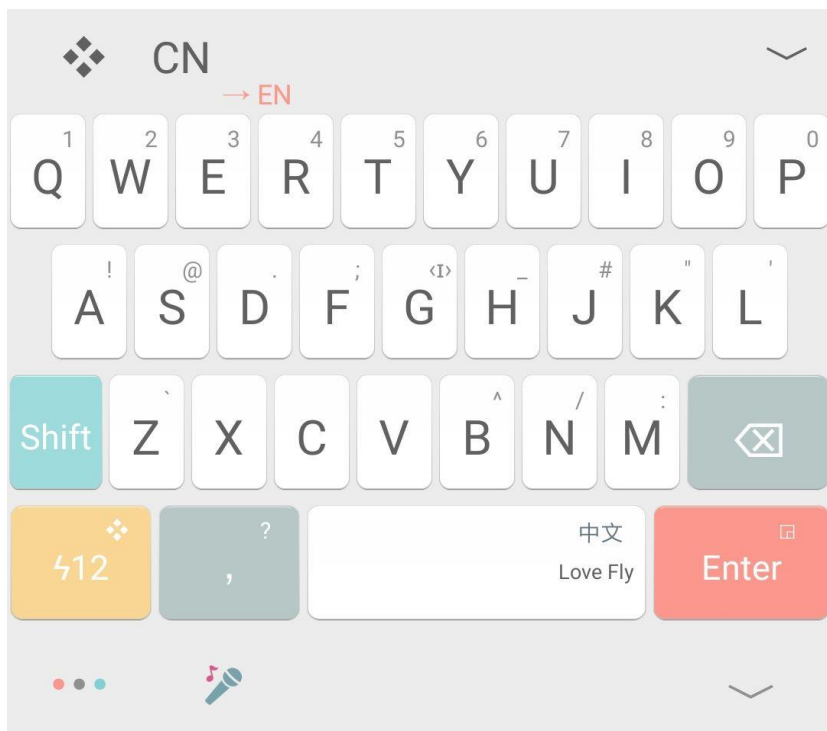


（长按 d 键就能打出句号了）

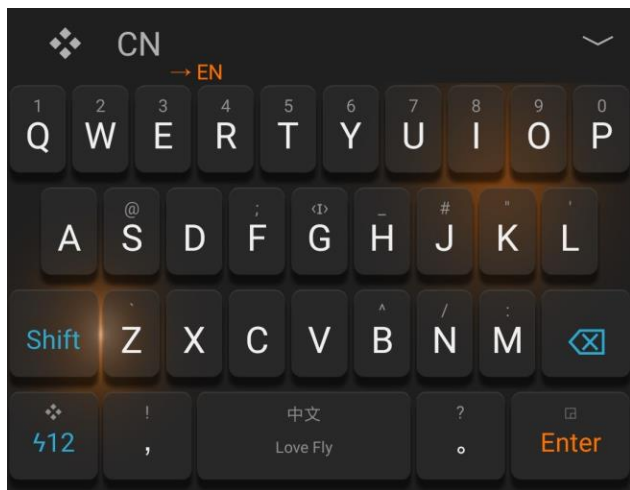
然后上滑一下 Enter 键， 看到四个主题。每个主题都有很多配备的颜色，简称配色。



比如比如上图中最下面的“高飞”主题长这样子，下面多出的一行是为了在打字的时候平衡手机重量：



再比如下面这个是 2019 年元旦的新配色（在“霓裳”主题中打开，所以末排即有逗号也有句号）：



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——原来是作者把方案的名称改为了“Love Fly”而不是“小鹤音形”。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，你甚至可以换成自己喜欢的图片为按键背景颜色）

除了主键盘之外的键盘都叫副键盘，主键盘是你一打开输入界面首先弹出来的键盘。另外，上面几张图其实都是主键盘，每一个主题都有且只能有一个主键盘（不过作者认为这其实是一个缺点）。

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，它们都是副键盘。

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。

[\(补丁是什么?\)](#)

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。

（总之看不懂完全没关系，如果你只是想改动一下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，这就是你需要了解的全部代码，井号后面是注释）[返回目录](#)

比较常见的原因是配置文件出现语法错误了，需要检查空格缩进、引号配对、是否错用 tab 等等。哪怕是一个空格出错了，也有可能使整个配置文件失效，初学者需要格外小心。为了让读者注意，这句话会在后面不断重复。

看看 trime.yaml 文件的整体框架

[1234 再来一次](#):

我们用 Quikedit 打开 trime.yaml 这个默认主题的框架；

下面就是 trime.yaml 的整体框架啦，
可以看到有三千多行代码。
尽管代码多，但总体结构并不复杂，由以下绿色字体组成。

```
1  # Trime default style settings
2  # encoding: utf-8
3
4  config_version: "3.0for同文3.1.2-20190107, 2019年1月11日"
5  name: 小鹤 #主题名称
6  author: 散步的鹤 flypy.com#作者资讯
7  #键盘高度调整
8  height:
28 #其他的键盘配置
29 other:
39 #输入法风格
40 style:
111 #后备配色
112 fallback colors:
150 #配色方案
151 preset color schemes:
1024 preset keyboards:
3437 #安卓按键
3438 android keys:
3520 #预设按键
3521 preset keys:
3791 #结束
3792
3793
```

Fallback colors 是后备配色，当你的配色方案没设置相应的配色时，系统会自动从后备配色里取色
你要想设置具体的配色，就要在 preset color schemes 里设置。

你要想设置按键，就要在 Preset keyboards 里设置。

你要想设置按键，还必须要在 Preset keys 里设置。

Android keys 为安卓按键，不需要管。

这就是小鹤音形的 trime.yaml 文件的整体框架，

下面再来看一下寄生主题的框架：

```

1  # Trime default style settings
2  # encoding: utf-8
3
4  config version: "for小鹤主题2.3版"
5  name: 高飞
6  # 本主题依附于小鹤主题使用，下面__incl
7  # 如使用本主题，小鹤主题名应为: trime.
8  # 此主题内增加配色即可达到借用小鹤主题
9  #
10 author: 散步的鹤 flypy.com#作者资讯
11  height:
20  # -----结束-----
21  style:
26  android keys:
28  preset keys:
30  fallback colors:
33  #配色方案
34  preset color schemes:
177 preset keyboards:
333

```

哎，发现和前面是一样的。但是代码的行数少了很多。这就是寄生主题的好处。不过之所以叫寄生主题，是因为运用了大量的补丁技巧。

[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“[看看 trime.yaml 的整体框架](#)”这节内容吧。

以主键盘为例

我们直接来操作吧：

① 改 Enter 键和 a 键的位置：

找到 preset_keyboards:→default 键盘，然后找到下列按键：

```

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，嗨~

name: "26 键 2"
width: 10

```

把它们的顺序改变一下就行→

```
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
```

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。

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）。

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}
```

把里面这行

```
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: 0xffFFFA40, 这是没按下时的按键背景色
```

```
hilited_key_back_color: 0xffFFFA40 了 hilited 是高亮的意思，
```

0xffFFFa40 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，取值范围从 00 到 ff，不懂十六进制的同学可以百度一下，这里不展开讲了。

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。

补充说明：

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。

其中的 width 这个命令若放在【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。

```
- {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
```

比较常见的原因是配置文件出现语法错误了，需要检查空格缩进、引号配对、是否错用 tab 等等。哪怕是一个空格出错了，也有可能使整个配置文件失效，初学者需要格外小心。

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。**红色的字是注释**

```
#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: 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
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 #a1 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）

```

比较常见的原因是配置文件出现语法错误了，需要检查空格缩进、引号配对、是否错用 `tab` 等等。哪怕是一个空格出错了，也有可能使整个配置文件失效，初学者需要格外小心。

关于上面的注释，如果你对上面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

改变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……这个 `trime.yaml` 文件基本就到这了，剩下的就让读者以后自己去慢慢探究。

④增加一个自己的副键盘（增加主键盘呢？目前同文主键盘只能有一个，所以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，输入法里的一页（或者说一个界面，一个你打字时所看到的界面）就是一个 键盘，所以你的键盘有多少页，就要写多少个名字，这里我新增了两页 `F7` and `YD`。

第二步、在键盘区（即在 `Preset_keyboard` 下数的这些代码区域里找一个合适的位置）写你的新键盘，

同时，我还是要强调：务必注意空格缩进、引号配对、是否错用 `tab` 等等。哪怕是一个空格出错了，也有可能使整个配置文件失效，初学者需要格外小心呐~

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，系统找不到，所以自动用后备配色相应的代码代替
#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，这是一个就叫作 K_symbolsF7x 的命令操作，请看这里 [K_symbolsF7x](#)，这条命令的意思是，点击它，它就拾起另一个叫 symbolsF7 的键盘。

命令的跳转是这样的：

你在 preset_keyboard 里写了一个键盘，然后它的名字放在户口本里——也就是 style 下数的 keyboard 里。

除此之外你还要在 preset_keys 里添加一个呼出按键，呼唤出你在 preset_keyboard 里的键盘。

当你在 preset_keyboard 里的其他键盘按下这个呼出按键时，就能呼出你新写的键盘了（或者说，反到了新增的一页）


```

35 # -----输入法风格开始
36 style: 户口本
87 keyboards: [.default, default, defaultVJ, letter,
108 # -----输入法风格结束
109 # -----户各配色开始
110 fallback_colors:
143 #配色方案开始
144
145 preset_color_schemes:
1016 # 键盘布局# -----
1017 preset_keyboards:
3429 android_keys:
3510 preset_keys:

```

所以，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](#)

2.2.5 高飞寄生主题文件

寄生主题的关键在于引用已有内容。寄生默认主题 trime.yaml 时，需要引用它的已有内容。

寄生主题文件也是一个完整的主题文件，所以默认主题有的框架，它也必须要有。于是我们直接引用默认主题的框架，看下面：

```

style:
  __include: trime:/style
  __patch:
    color_scheme: xuancai #默认配色方案
    key_height: *FJG
android_keys:
  __include: trime:/android_keys
preset_keys:
  __include: trime:/preset_keys
fallback_colors:
  __include: trime:/fallback_colors

```

该主题的风格直接引用了默认主题的 style，引用格式见上面的代码。先__include 过来，如果需要更改，就再__patch 新的更改内容。比如高飞主题的高度需要变化，我们就打一个高度补丁，叫 key_height，高度为*FJG，然后再在高度里重新定义高度 height: 而*FJG 就在下面

```

height:
# -----键盘高度调整-----

A: &VJG 56 #主键盘键高
B: &VJM 44.5 #主键盘末行键高
C: &FVG 67 #辅助键盘键高 C=(A*4+B)/4=67 与主键盘高度不等时调整
D: &FVM 68.4 #辅助键盘末键高≈C
E: &FJG 53.7 #符号键盘键高 E=(A*4+B)/5=53.6 与符号键盘高度不等时调整

```

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
    .....老生常谈之：需要检查空格缩进、引号配对、是否错用 tab 等等。哪怕是一个空格出错了，也有可能使
    整个配置文件失效，初学者需要格外小心。
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>

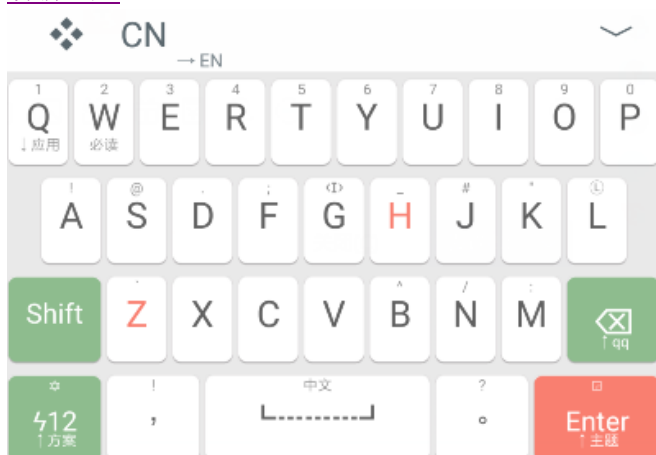
用 Mt 管理器打包输入法

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

[返回目录](#)

小彩蛋：

[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）和默认主题的区别就是小键盘界面增加了很多长按、滑动的功能，见下两图。主要是为了方便在手机上打赛文和编辑代码

长按 g 键进入编辑键盘界面，下图：



点击 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、openc 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。（2018 年 4 月 6 日）

最新进展二：现在的同文已经有 **build** 了，层次更分明。云同步、高度灵活可调 等新功能的无障碍版正在 qq 群之：同文堂里火热进行。期待新功能的诞生。2019-01-15

本人邮箱：1045865146@qq.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](#)