


简介,看看它的主页就明白了



more languages

home

download

news

features

resources

contribute

donate

contributors

links

download

Current Version: 6.7.3

News

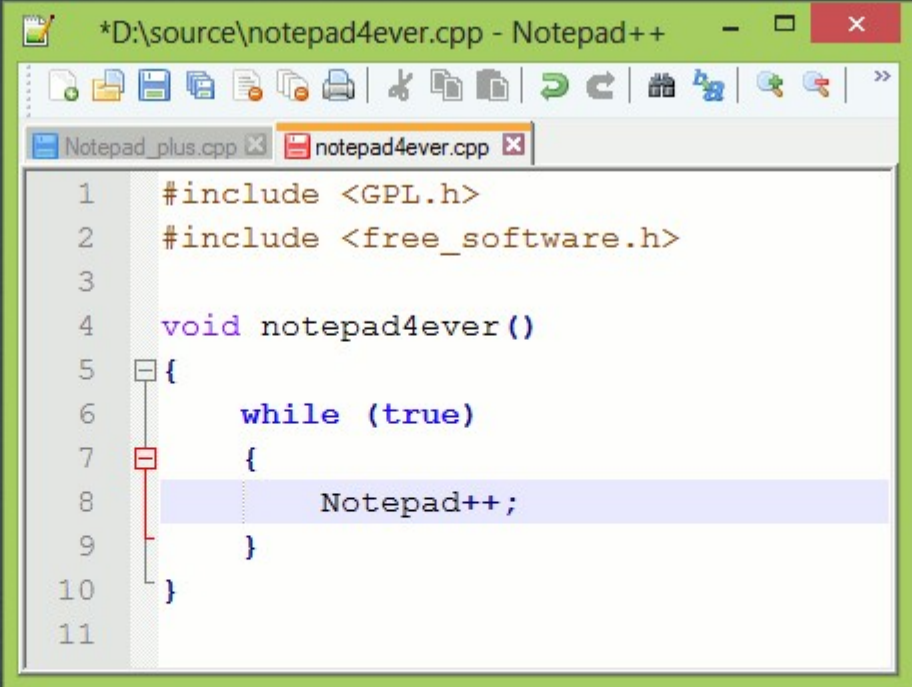
- **Notepad++ 6.7.3 released**
Jan 01 2015
- **Notepad++ 6.7.2 released**
Dec 27 2014
- **Notepad++ 6.7.1 released**
Dec 21 2014
- **Notepad++ 6.7 released**
Dec 14 2014
- **Notepad++ 6.6.9 released**
- **Notepad++ 6.6.8 released**
Jul 27 2014
- **Notepad++ 6.6.7 released**
Jun 24 2014
- **Notepad++ 666 Friday the 13th edition**
Jun 12 2014
- **Response and Apology for Sexist Jokes**
Jun 07 2014
- **Notepad++ 6.6.4 Tiananmen June Fourth Incident Edition**
Jun 04 2014

[more news »](#)

About

Notepad++ is a free (as in "free speech" and also as in "free beer") source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by [GPL License](#).

Based on the powerful editing component [Scintilla](#), **Notepad++** is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size. By optimizing as many routines as possible without losing user friendliness, **Notepad++** is trying to reduce the world carbon dioxide emissions. When using less CPU power, the PC can throttle down and reduce power consumption, resulting in a greener environment.



```
1  #include <GPL.h>
2  #include <free_software.h>
3
4  void notepad4ever()
5  {
6      while (true)
7      {
8          Notepad++;
9      }
10 }
11
```

You're encouraged to [translate Notepad++](#) into your native language if there's not already a translation present in the [Binary Translations](#) page.

贴一段百科上面的介绍([via](#)):

Notepad++是一套非常有特色的[自由软件](#)的纯文字[编辑器](#)(许可证:[GPL](#))，有完整的[中文](#)化接口及支持多国语言编写的功能(UTF8 技术)。它的功能比 Windows 中的 [Notepad](#)(记事本)强大，除了可以用来制作一般的纯文字说明文件，也十分适合当作编写电脑程序的[编辑器](#)。Notepad++ 不仅有语法高亮度显示，也有语法折叠功能，并且支持宏以及扩充基本功能的外挂模组。

Notepad++是一款非常有特色的编辑器，是开源软件，可以免费使用，自带中文。支持的语言: C, C++ , [Java](#) ,pascal, C#,[XML](#),SQL,Ada, [HTML](#), [PHP](#), ASP, AutoIt, 汇编, [DOS](#) 批处理, Caml, [COBOL](#), [Cmake](#), [CSS](#), [D](#), [Diff](#), [ActionScript](#), [Fortran](#), Gui4Cli, [HTML](#), [Haskell](#), INNO, [JSP](#), KIXtart, [LISP](#), [Lua](#), [Make](#) 处理 (Makefile), [Matlab](#), [INI](#) 文件, MS-DOS Style, [NSIS](#), Normal text, [Objective-C](#), [Pascal](#),[Python](#), [Javascript](#), [VerilogHaskell](#)InnoSetup[CMake](#), VHDLAdaCaml、AutoItKiXtart、[Matlab](#)

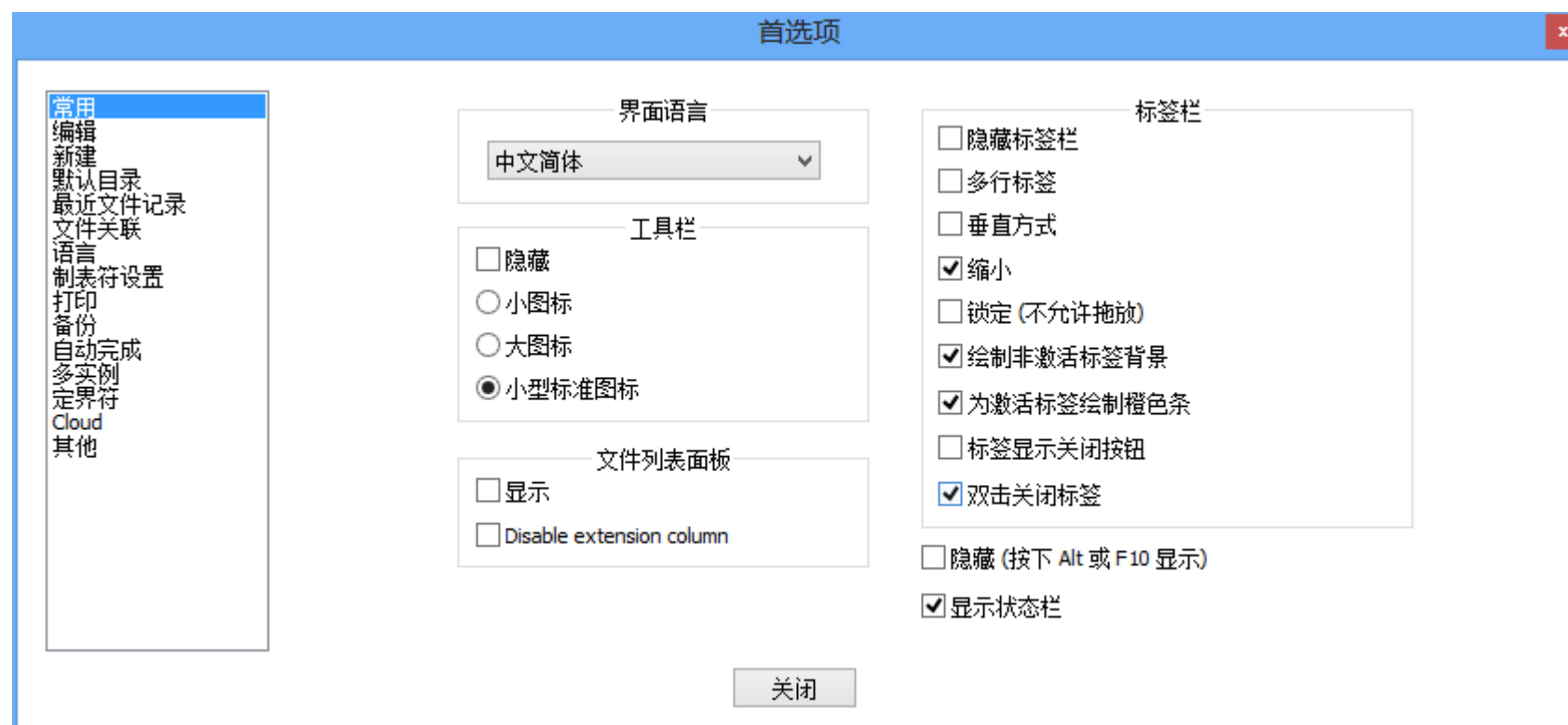
关于各种编辑器的比较什么的,这里我也不说些什么,没有深入的使用,不方便发表什么评论.

假设亲爱的你已经自己将软件安装完毕,我们接下来来做一些相关的设定.

Npp(notepad++的简称,后文若不作说明,皆是指 Notepad++)的设置

默认的设置是哪些,有哪些变动我现在也记不清了,直接把自己的一些设定截图放上来就 OK,需要的就比对查阅一下

设置>首选项>常用:



设置>首选项>编辑:

边界宽度 80,这也是有历史原因的(屏幕不够宽,限定在 80 的话代码就不会断行显示),这样便于代码阅读.



设置>首选项>制表符的设置:

选定 python,将使用默认值去掉勾选,然后设定制表符宽度为 4(关于制表符宽度,可以参考一下 Python 的 [PEP-8](#)).

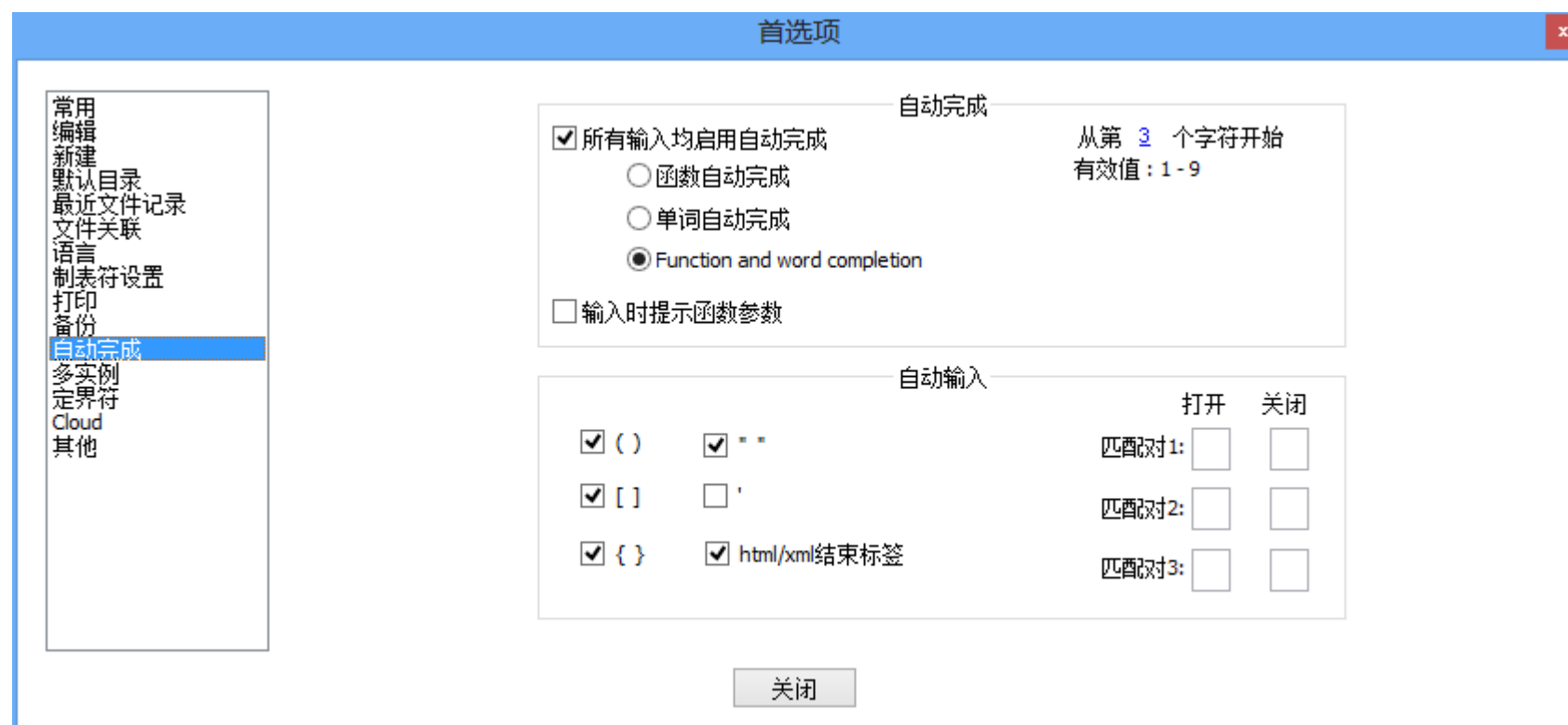
Python 对于缩进是有严格的要求的,用 4 个空格代替 Tab 基本上是标准(反正我是信了).

同样的,你也可以顺手将[Default]项中的制表符宽度设定一下.

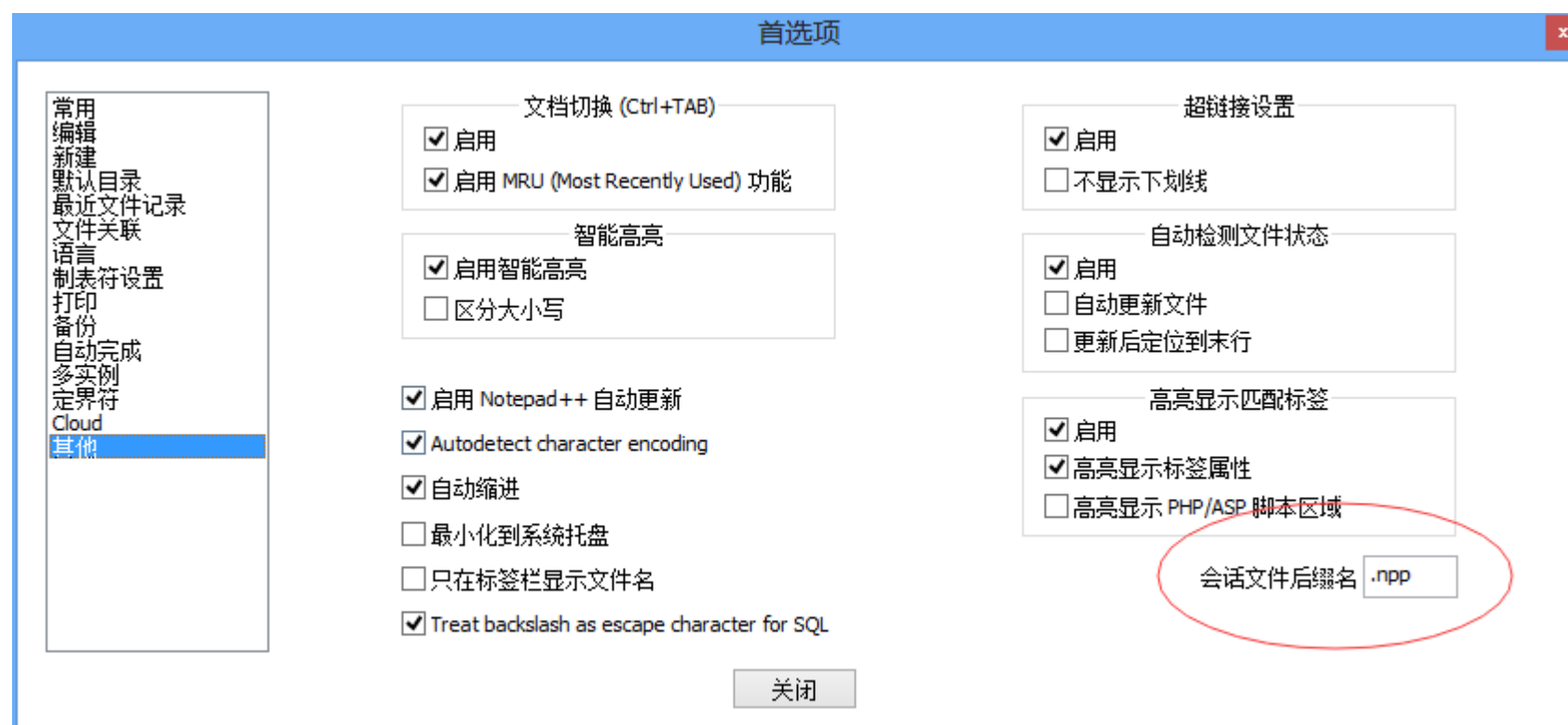


设置>首选项>自动完成:

几个括号自动完成确实是能省不少事,单词和函数自动完成也是不错的(从第 3 个字符开始,免得输入一个单词就开始提示,这样提示也未免太频繁了…)



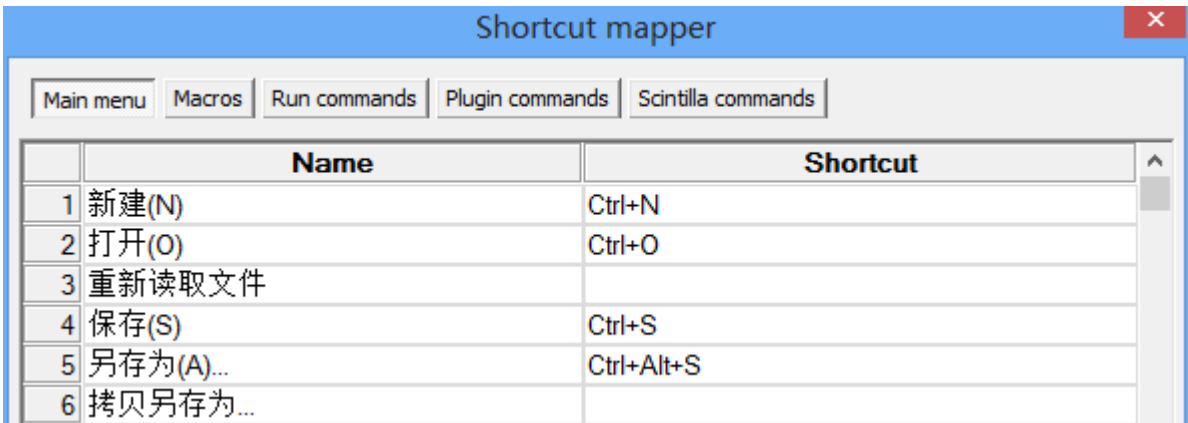
Notepad++支持保存当前的工作到文件,下次直接点击文件就能还原到之前的工作状态(当然,关闭了下次打开也会恢复到最后一次的工作状态).为自己的会话文件注册一个好记好用的后缀名.



Npp 快捷键设定

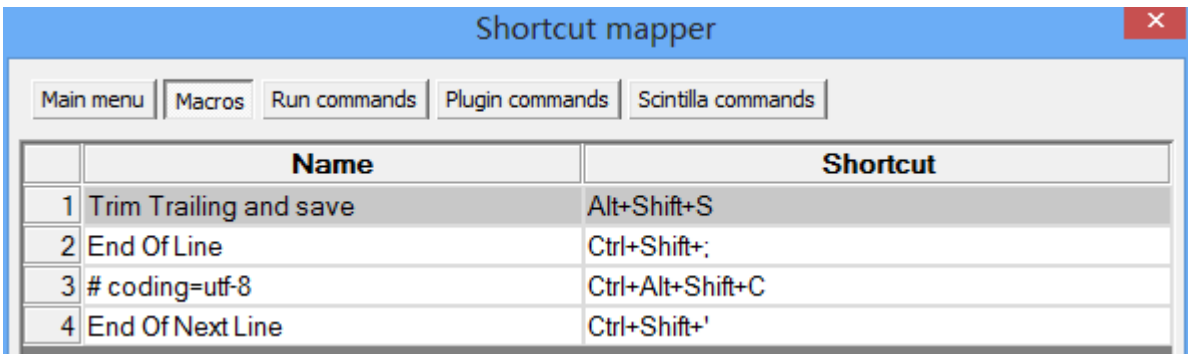
为码农准备的编辑器都会有各式各样的快捷键设定,npp 也不例外.关于 npp 的快捷键设定,大体上分为以下几类

1 俗成的常用快捷键,像 Ctrl+C/V/A/X 等等的一系列地球人都知道的快捷键.

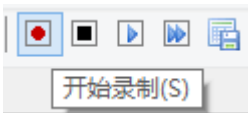
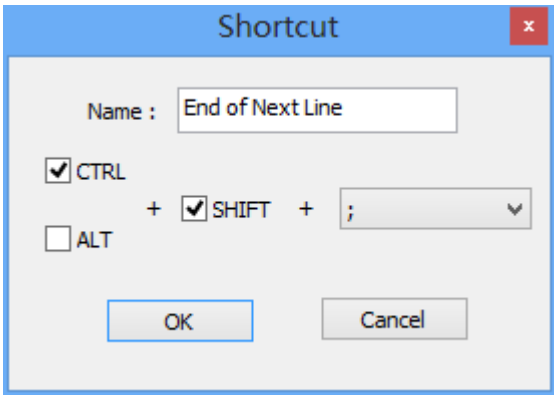


2 定义的各种宏并为之注册的快捷键

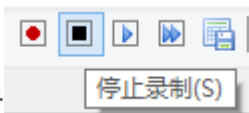
这里我需要推荐一下的就是 Alt+Shift+S ,与 Ctrl+S(保存)不同是在于,该快捷键多做了一个保存前删除所有多余的空格的命令.是我最喜欢的快捷键之一,基本上代替了 Ctrl+S 了.~图中也有作者自己注册的几个编码中常用到的宏,一个是跳转到行尾,一个是编码声明,一个是跳转到下一行,这样就只需要动动手指头而不是动动手腕就能达到同样的效果(一个字,懒).



Tips:宏的录制与设定,这里我们以上图中第二个宏的录制与快捷键设定为例



点击开始录制,然后点击键盘上的 End 键,然后点击



.然后再点击



,在窗里面设定喜欢顺手的快捷键然后 OK 一下就完

成了, so easy~~

3 令注册的快捷键

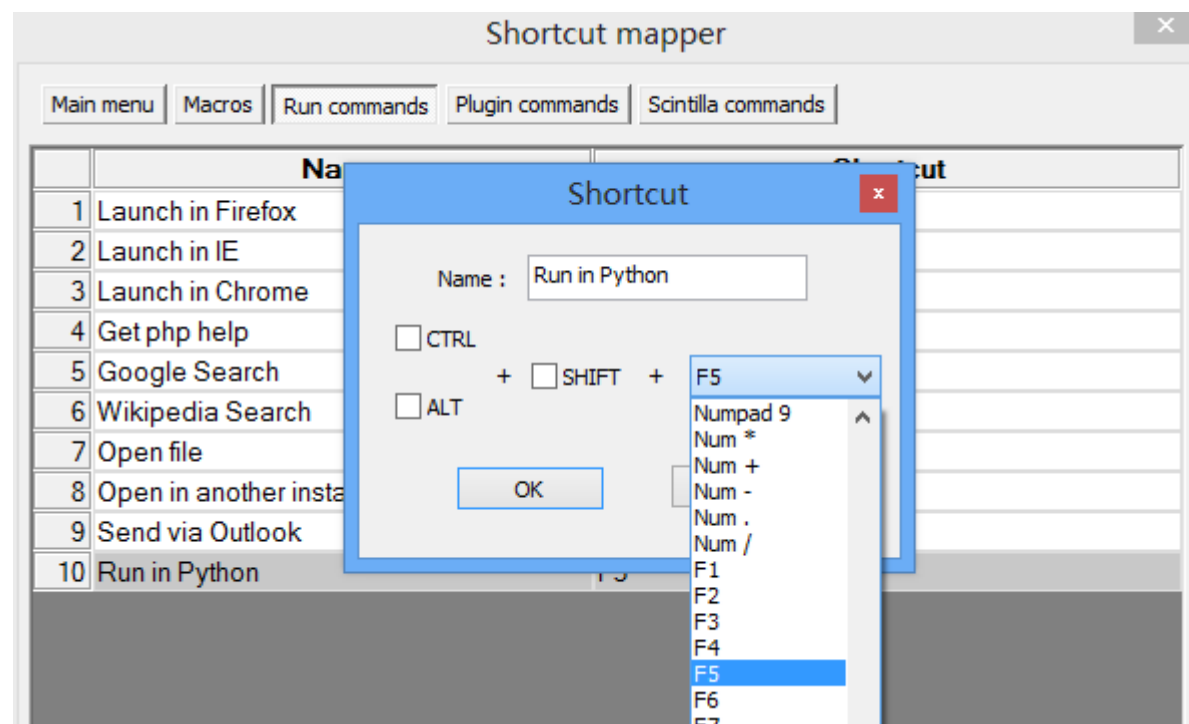
npp 自带的有各种在浏览器中打开当前文件的快捷键,搜索某某的快捷键,可能这里亲爱的你注意到了,第 10 个 Run in Python,这里实际上就是一条命令的调用.后面会详细介绍一下.

Shortcut mapper		
Main menu Macros Run commands Plugin commands Scintilla commands		
	Name	Shortcut
1	Launch in Firefox	Ctrl+Alt+Shift+X
2	Launch in IE	Ctrl+Alt+Shift+I
3	Launch in Chrome	F6
4	Get php help	
5	Google Search	
6	Wikipedia Search	
7	Open file	Alt+F5
8	Open in another instance	Alt+F6
9	Send via Outlook	
10	Run in Python	F5

4 的安装引入的相关的快捷键,和其他的相关的快捷键

这里不做过多描述.

快捷键的指定,只需要选定想要修改的快捷键,双击就会弹窗供你修改,需要注意的是,不要修改成同样的按键,这样可能会有冲突导致按键的时候并不能正常的响应你指尖的操作哟(也可以先去将已经存在的一个快捷键删除掉—指定为 none,或者修改成别的,再回来该这个).



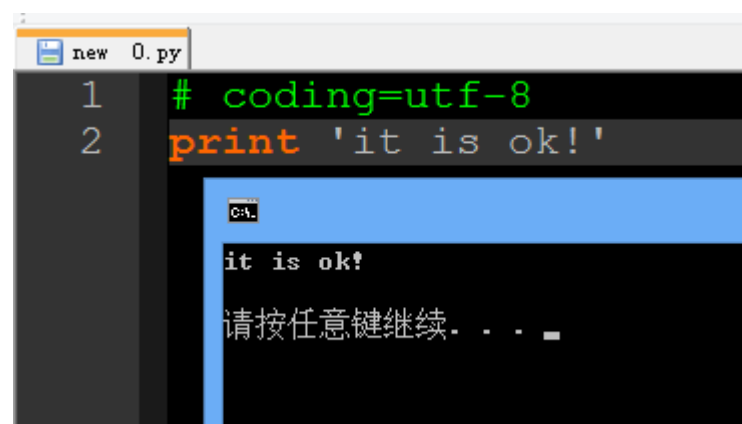
几条神奇的命令

用 Python 运行当前脚本文件:

运行>运行,输入框粘贴如下的命令然后保存,指定一个快捷键(也可以不用,这里我指定为 F5,记得把之前的 F5 给干掉,忘记在哪一栏了,就在上面说的几个里面,当然,也可以指定为其他的不会冲突的)

```
cmd /k python "%(FULL_CURRENT_PATH)" & ECHO. & PAUSE & EXIT
```

试试,保存文件后,F5 按一下就呼出了命令行窗口运行当前的文件,是不是很方便?

A screenshot of a code editor window titled 'new 0.py'. The editor has a dark background with a light blue vertical line on the left. The code is as follows:

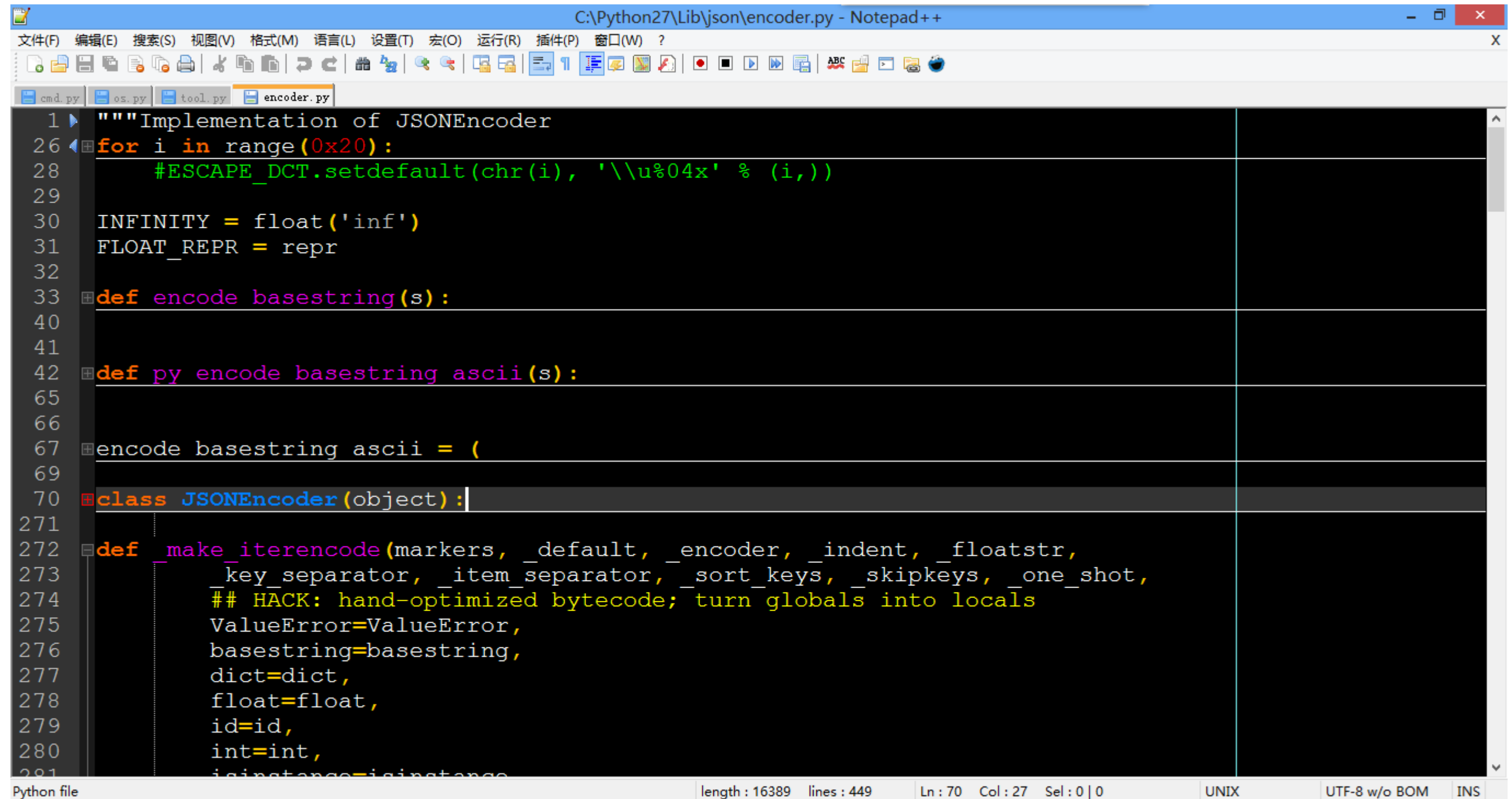
```
1 # coding=utf-8
2 print 'it is ok!'
```

The output of the script is shown in a separate window below the code, displaying 'it is ok!' and '请按任意键继续. . . _'.

Npp 的主题颜色

Npp 给用户提供了各种各样的主题配色,同样也支持自定义的颜色配置,笔者使用的主题颜色配置,截图一张.

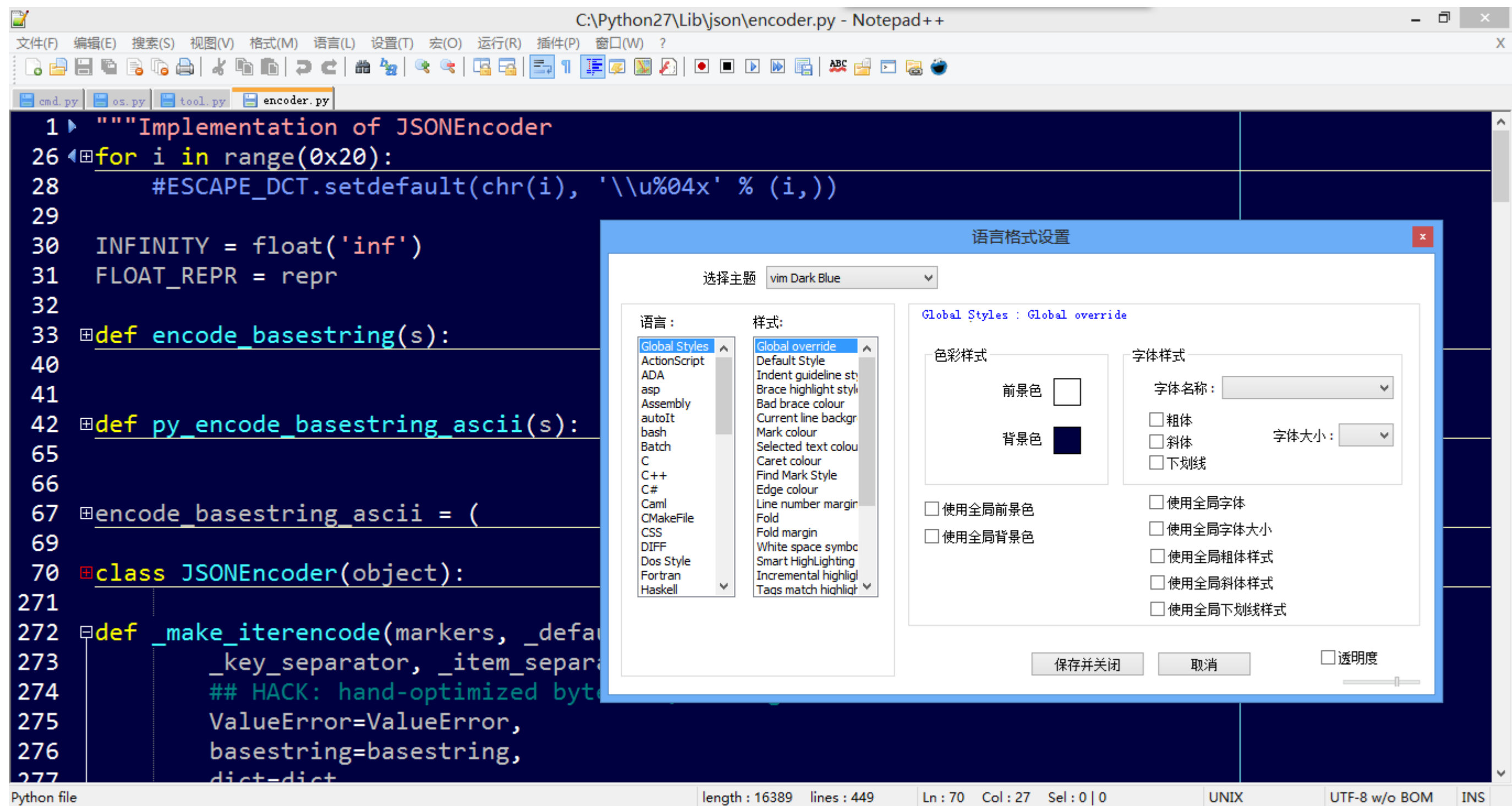
我的主题风格



```
1  """Implementation of JSONEncoder
26  for i in range(0x20):
28      #ESCAPE_DCT.setdefault(chr(i), '\\u%04x' % (i,))
29
30  INFINITY = float('inf')
31  FLOAT_REPR = repr
32
33  def encode_basestring(s):
40
41
42  def py_encode_basestring_ascii(s):
65
66
67  encode_basestring_ascii = (
69
70  class JSONEncoder(object):
271
272  def make_iterencode(markers, _default, _encoder, _indent, _floatstr,
273                      _key_separator, _item_separator, _sort_keys, _skipkeys, _one_shot,
274                      ## HACK: hand-optimized bytecode; turn globals into locals
275                      ValueError=ValueError,
276                      basestring=basestring,
277                      dict=dict,
278                      float=float,
279                      id=id,
280                      int=int,
281                      isinstance=isinstance,
```

Python file length : 16389 lines : 449 Ln : 70 Col : 27 Sel : 0 | 0 UNIX UTF-8 w/o BOM INS

Vim 深蓝色风格



Hello Kitty 粉色风格

这里简单介绍一下如何自定义自己喜欢的颜色.以我自己使用的为例

我使用的颜色配置是在自带的 Deep Black 的基础上修改而来.

设置>语言格式设置>样式

然后修改各种各种.这个留给好奇的你自己慢慢探索样式里面不同的分别代表什么了~

这里顺便提供一下自己已经修改好了的部分,显示的风格见上面的 我的主题风格下面的配图,如果了解一下这样配置具体做了些什么事情,可以看看下面的.

直接打开 Notepad++的安装路径下的主题文件 X:/SomePtah/Norepad++/themes/Deep Black.xml

Ctrl+F 搜索 python ,大概在第 318 行附近.当然,你也可以先折叠代码到第 3 层次,这样看也比较方便.

展开 python 的这一块,这里就可以看到关于 Python 部分的代码主题风格的颜色,字体的相关配置了,然后,我们就可以对它做一些自己爱做的事了.

这里简单说明一下,每一条 <WordsStyle name="DEFAULT" styleID="0" fgColor="FFFFFF" bgColor="000000" fontName="" fontStyle="0" fontSize="" /> 代表了一个配置项,配置了不同的类型的代码的显示:默认,注释,数字,字符串,关键字,类等等的,这里我们主要改的就是 fgColor,bgColor,fontSize 等等.你对改文件的修改就会直接反映到主题颜色,字体大小的改变上了.

后文将会给出我自己的主题风格的下载地址,嫌麻烦的也可以直接下载然后覆盖 Deep Black.xml 文件(注意先备份).这段源码附在下面

```
<LexerType name="python" desc="Python" ext="">
```

```
<WordsStyle name="DEFAULT" styleID="0" fgColor="FFFFFF" bgColor="000000" fontName="" fontStyle="0" fontSize="" />
```

```
<WordsStyle name="COMMENTLINE" styleID="1" fgColor="00FF00" bgColor="000000" fontName="" fontStyle="0" fontSize="" />
```

```
<WordsStyle name="NUMBER" styleID="2" fgColor="FF0000" bgColor="000000" fontName="" fontStyle="0" fontSize="" />
```

```
<WordsStyle name="STRING" styleID="3" fgColor="FFFFFF" bgColor="000000" fontName="" fontStyle="0" fontSize="" />
```

```
<WordsStyle name="CHARACTER" styleID="4" fgColor="C0C0C0" bgColor="000000" fontName="" fontStyle="0" fontSize="" />
```

```
<WordsStyle name="KEYWORDS" styleID="5" fgColor="FF6600" bgColor="000000" fontName="" fontStyle="1" fontSize="" keywordClass="instre1" />
```

```
<WordsStyle name="TRIPLE" styleID="6" fgColor="FF8000" bgColor="000000" fontName="" fontStyle="0" fontSize="" />
```

```
<WordsStyle name="TRIPLEDDOUBLE" styleID="7" fgColor="FFFFFF" bgColor="000000" fontName="" fontStyle="0" fontSize="" />
```

```
<WordsStyle name="CLASSNAME" styleID="8" fgColor="0080FF" bgColor="000000" fontName="" fontStyle="1" fontSize="" />
```

```
<WordsStyle name="DEFNAME" styleID="9" fgColor="FF00FF" bgColor="000000" fontName="" fontStyle="0" fontSize="" />
<WordsStyle name="OPERATOR" styleID="10" fgColor="FFCC00" bgColor="000000" fontName="" fontStyle="1" fontSize="" />
<WordsStyle name="IDENTIFIER" styleID="11" fgColor="FFFFFF" bgColor="000000" fontName="" fontStyle="0" fontSize="" />
<WordsStyle name="COMMENTBLOCK" styleID="12" fgColor="00FF00" bgColor="000000" fontName="" fontStyle="2" fontSize="" />
<WordsStyle name="STRINGEOL" styleID="12" fgColor="FFFF00" bgColor="000000" fontName="" fontStyle="0" fontSize="" />
</LexerType>
```

Npp 的各种插件

Npp 的一个很不错的地方就是它有丰富的插件,开源的好处就是有很多人为之添砖加瓦使之不断完善.这里还是先截个图列出几个常用的插件.

下面分别对其中的一些作一点点简单的补充说明.

JSLint 压缩 JavaScript 代码的一个小插件,作为发布代码或者简单的'加密'还是不错的,实际效果,可以参看一下百度主页源码上面密密麻麻的一片.

JSTool,功效供 JSLint 正好相反,格式化 js 代码的一个小工具

Light Explorer npp 下的一个轻量级的文件浏览器,快捷键 Alt+A(和某个快捷键的设定冲突了,这里需要手动删除那一个).快速方便的呼出文件浏览还是很方便的.

MMIE Tools 默认自带的插件,编码解码的一个玩意儿,自己摸索下~

NppExec

这个倒是可以多介绍一下,作为利用 npp 作为日常 Python 编写的一员,这个插件倒是方便的紧.

留空…下次补上

Npp Export

这个自带的格式化导出插件,可以将当前的代码保留格式导出.效果杠杠滴,看下面.

在 word 里面粘贴出来就是这样的效果,方便的紧~~

```
# coding=utf-8  
print 'it is ok!'
```

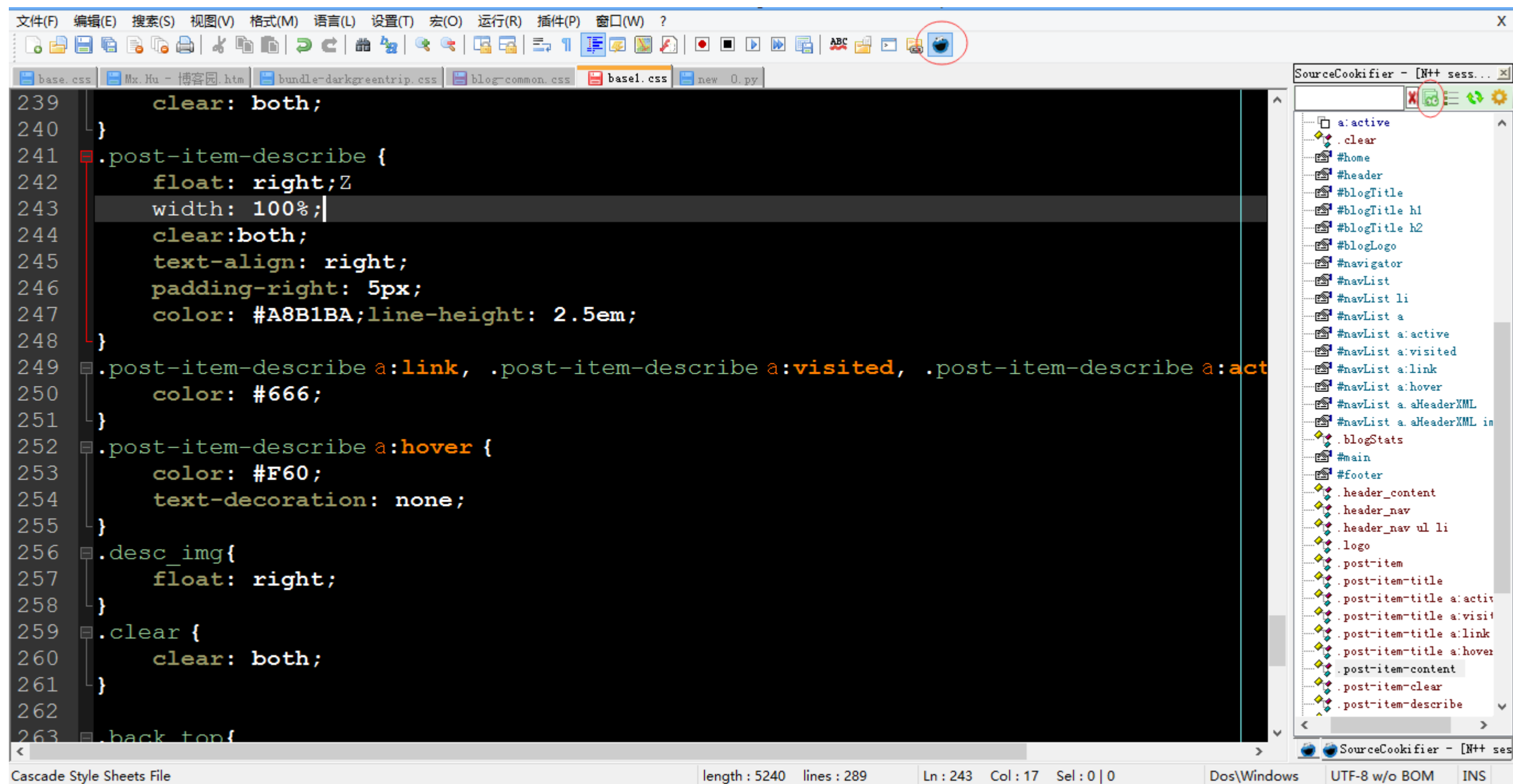
JSON Viewer

同 JSTool 一样的,也是格式化的工具.可以将不规范的 json 格式的字符串格式化.笔者平时编写代码的时候免不了需要同前后台的数据打交道,所以这个格式化数据的插件也是很不错的.

SourceCookifier

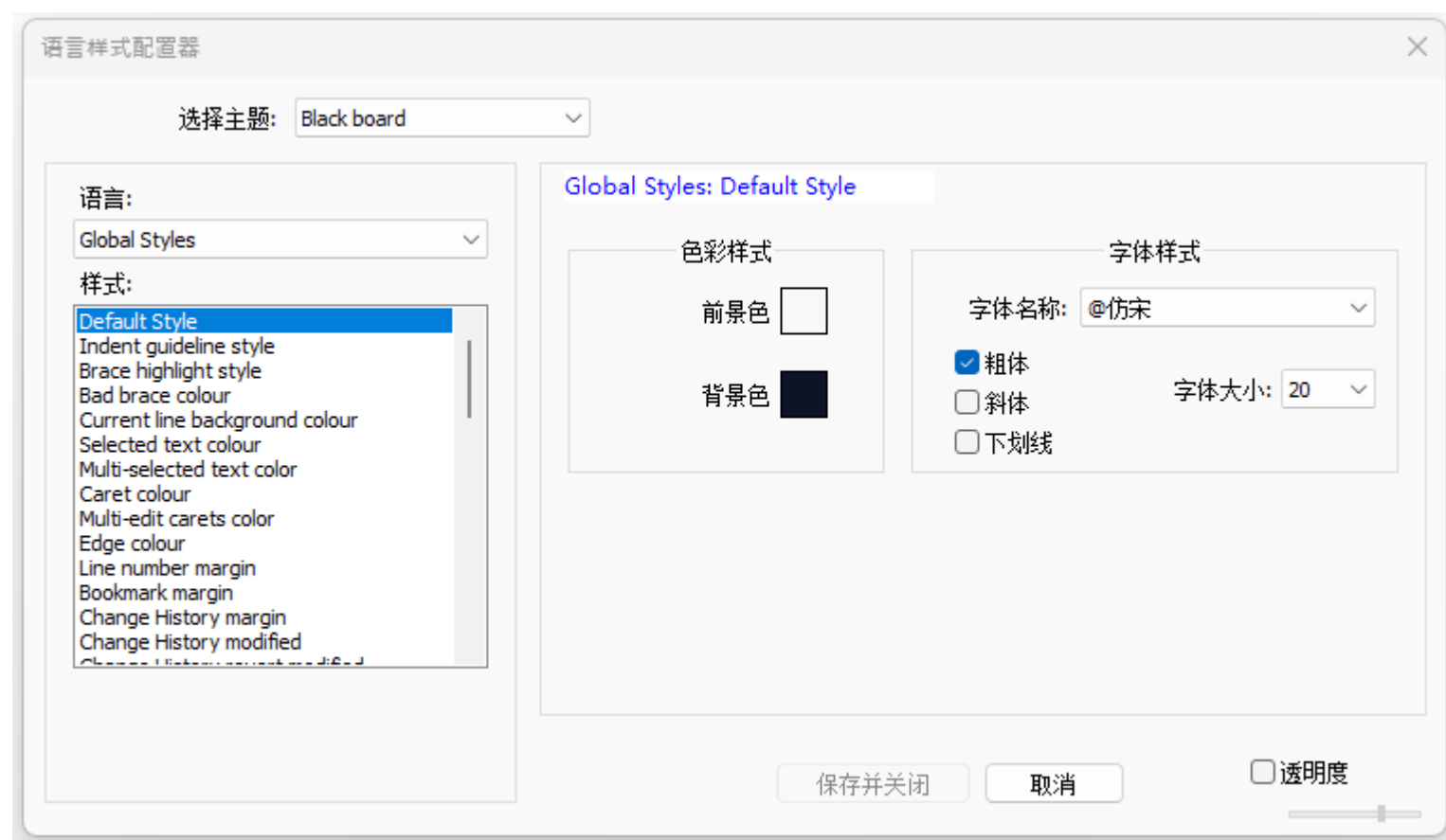
这个插件也很不错,我还是接着截图说明一下,最右边的那一坨黑漆漆的丑图标就是该插件的图标了.确实是够不好看的….

设置上,看见右侧圈出来的了么?选择 N++ session module. 这样在打开的所有文件里面,当看到一个陌生的调用的话,Ctrl+鼠标左键 点击函数名称,就会跳转到函数的定义处,是不是很炫酷,就是这么吊炸天,这对于快速熟悉一段程序真实帮助颇多啊.(右边的侧窗关闭了也是可以的哟)



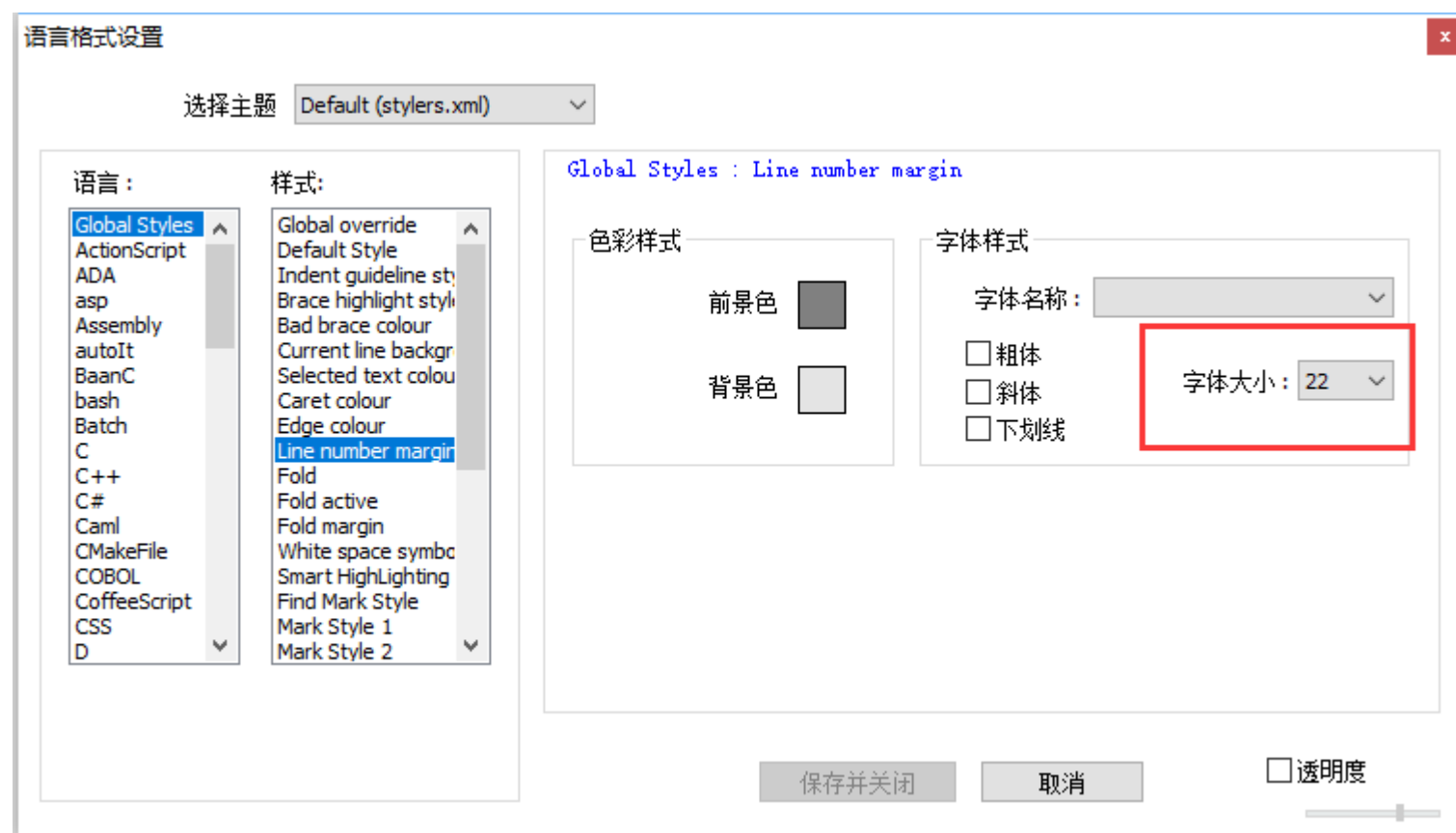
Notepad++页面设置

设置☒语言格式配置器☒语言 (Global Styles) ,样式 (Defenlt Style) 选择主题 (Black board)



notepad++ 调整行间距

在“设置” - “语言格式设置”里面，找到 style 里面的 Line number margin 一项，调整字体大小就可以调整左边标号的大小，然后文本内容的行间距即可任意调整。



学会这几个插件，让你的 Notepad++ 使用起来更丝滑

搞程序开发的小伙伴相信对 Notepad++ 都不会陌生，是一个占用空间少、打开启动快的文件编辑器，很多程序员喜欢使用 Notepad++ 进行纯文本编辑或者脚本开发，但是 Notepad++ 的功能绝不止于此，下面我们就来讲讲如何使用 Notepad++ 配合插件来提高工作效率。

1、Compare

这是一个差异对比插件，有个小伙伴会说和 BCompare 有什么区别？区别是这个插件不仅支持文件间的差异对比，还支持 git 管控下的同一个文件不同提交记录下的差异对比。作为一个插件，Compare 即开即用，用完就走，比 BCompare 使用起来更加方便。

插件功能

我们先来看一下文件间差异对比展示，对比时，可以通过右边的差异定位条快速定位到差异行，差异行以及差异字符不同深度颜色高亮显示。

文件间差异对比展示

再看一下 git 管控文件版本间差异对比展示，可以看到和最近的 commit 记录比较，我们又进行了文件修改，删除了方法注释。

git 管控文件不同记录差异对比

还可以在插件的 Settings 里面对文件展示、标记色设置个人偏好：

2、Explorer

这个插件可以实现在 Notepad++ 左边显示一个文件资源管理器，和 windows 系统一样的树状目录结构，便于我们快速定位文件和修改，再也不用去系统文件管理器先一个一个找到文件位置再打开文件了，妈妈再也不用担心我找不到作业在哪里啦！

插件功能

需要注意的是，这里如果打开 NotePad++ 不支持的文档，就会以乱码格式显示，所以只能算是个轻型文件系统，用来打开 Notepad++ 支持格式的文档。

3、NppExport

这个插件主要是用来进行代码导出和代码复制，有的小伙伴可能说我左手 C 右手 V，它不香吗？为啥还要脱裤子放~，用它来导出/复制呢？且听我娓娓道来。

插件功能

我们平时 CV 拷贝-粘贴都是复制的代码内容，无法保留代码颜色、字体格式等信息，理论上日常使用也够了，但当需要做项目成果 PPT 等文档或将代码 demo 用作 web 展示时，就需要表达更多的信息，需要代码不只是只有内容，还需要格式、颜色等，这时候我们就可以使用这个插件来实现。

3.1 Export to RTF

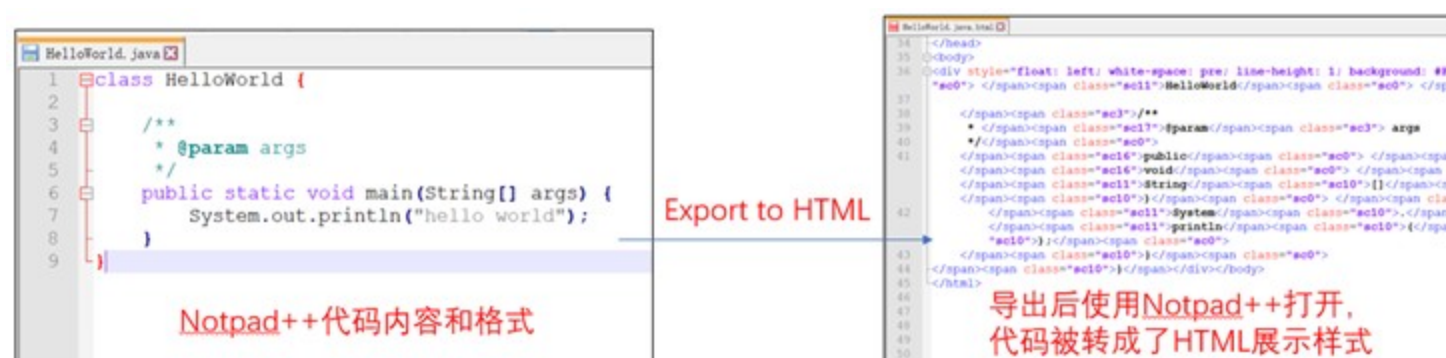
导出为可以使用 Word 打开的富文本文件。可以看到，代码颜色和格式都完整地保留了：



Export to RTF

3.2 Export to HTML

导出为 HTML 展示格式的文件。对于前端小伙伴来说，这个功能还是比较实用的，可以将展示内容直接一键转换，不用再搞一大堆标签来手动实现：



Export to HTML

导出后在 Chrome 上打开，展示效果是这样滴：

3.3 Copy RTF to clipboard

这个和 Export to RTF 其实差不多，只是 Export to RTF 是保存为文件，Copy RTF to clipboard 是将富文本拷贝到剪切板。

3.4 Copy HTML to clipboard

这个和 Export to HTML 差不多，只是 Export to HTML 是保存为.html 文件，Copy HTML to clipboard 是将带有 html 标签的文本内容拷贝到剪切板。

3.5 Copy all formats to clipboard

这个功能相当于 Copy RTF to clipboard 和 Copy HTML to clipboard 两个的综合，实现将文本转为带标签的 html 格式，然后还保存背景色等格式。

4、NppFTP

这是一个远程文件编辑插件，可以实现在 windows 本地编辑远程服务器文件，也是我使用最频繁的一个插件。经常在 Linux 服务器上做开发的小伙伴，相信都应该知道 Linux 服务器上的 VI 编辑器，VI 编辑器用来少量改动文件还是极好用的，但是如果文件改动点比较多，那么使用 VI 编辑器就比较费劲，需要在各种命令间来回切换，还要频繁移动光标定位。这时候，我们就需要用到下面这个插件来提高我们的工作效率。

插件功能

打开 NppFTP 插件后，可以看到 Notepad++ 多出两个模块，如下图所示。FTP 文件管理器主要是在连接到远程节点后进行文件目录树展示用的，操作反馈输出窗口则是在我们对文件进行编辑保存后，反馈操作结果的。

NppFTP 功能模块

然后我们需要通过文件管理模块上方的菜单栏来：设置远程服务器节点连接信息，远程节点连接两个操作，如下图：



连接后，会自动进入到节点登录用户的根目录：

然后，我们就可以定位到我们需要修改的文件，进行文件修改，保存。保存成功后，可以在反馈输出窗口看到，提示文件已经 upload succeeded。这个时候如果你再去服务器上打开这个文件的话，可以看到，文件已经被修改过了。

5、Json Viewer

这个插件可以使字符串格式的 json 转成标准 json 格式，让你不用再吃力阅读了。

对字符串 json 分别进行 Show JSON Viewer 操作和 Format JSON 后获取到标准 json 格式文件：

JSON 格式化

以上这些插件在使用 Notepad++进行生产开发时配合使用，能够极大地提高开发效率。

如果，有的小伙伴还有其他好的插件，可以在评论区留言，做一波分享~

最后，点个关注再走吧，避免错过更多干货~

发布于 2021-05-16 16:13