



[Home](#)



[Archive](#)



[Resume](#)



[LLVM#](#)



[PRs](#)



[Gists](#)



[LAgda](#)



[About](#)

---

# 如何正确使用 DEV C++

2016, Oct 28 by Tesla Ice Zhang

我自身作为一个高度依赖 IDE 的人，在拿到 DEV C++之后也是高度开发它的功能。我觉得对于 Oier 来说，用好这个 IDE 是非常重要的。

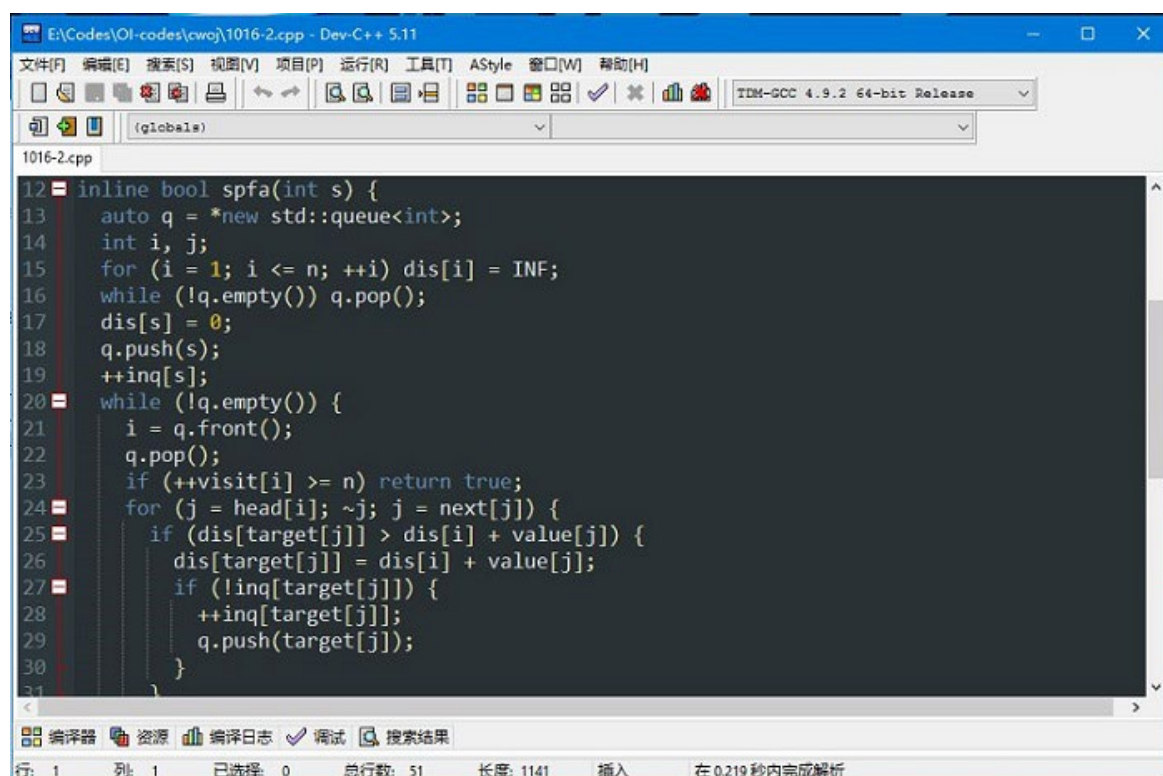
虽然这破 JB 玩意早就停止更新子

## 依赖

---

- [DEV C++ 5.11](#) 最新版

下载安装 DEV C++后，使用 DEV C++打开一个随便的 `cpp` 文件，你看到的应该是这样的界面。（为了节约读者的流量，图片进行了有损压缩，但是字看得清楚）



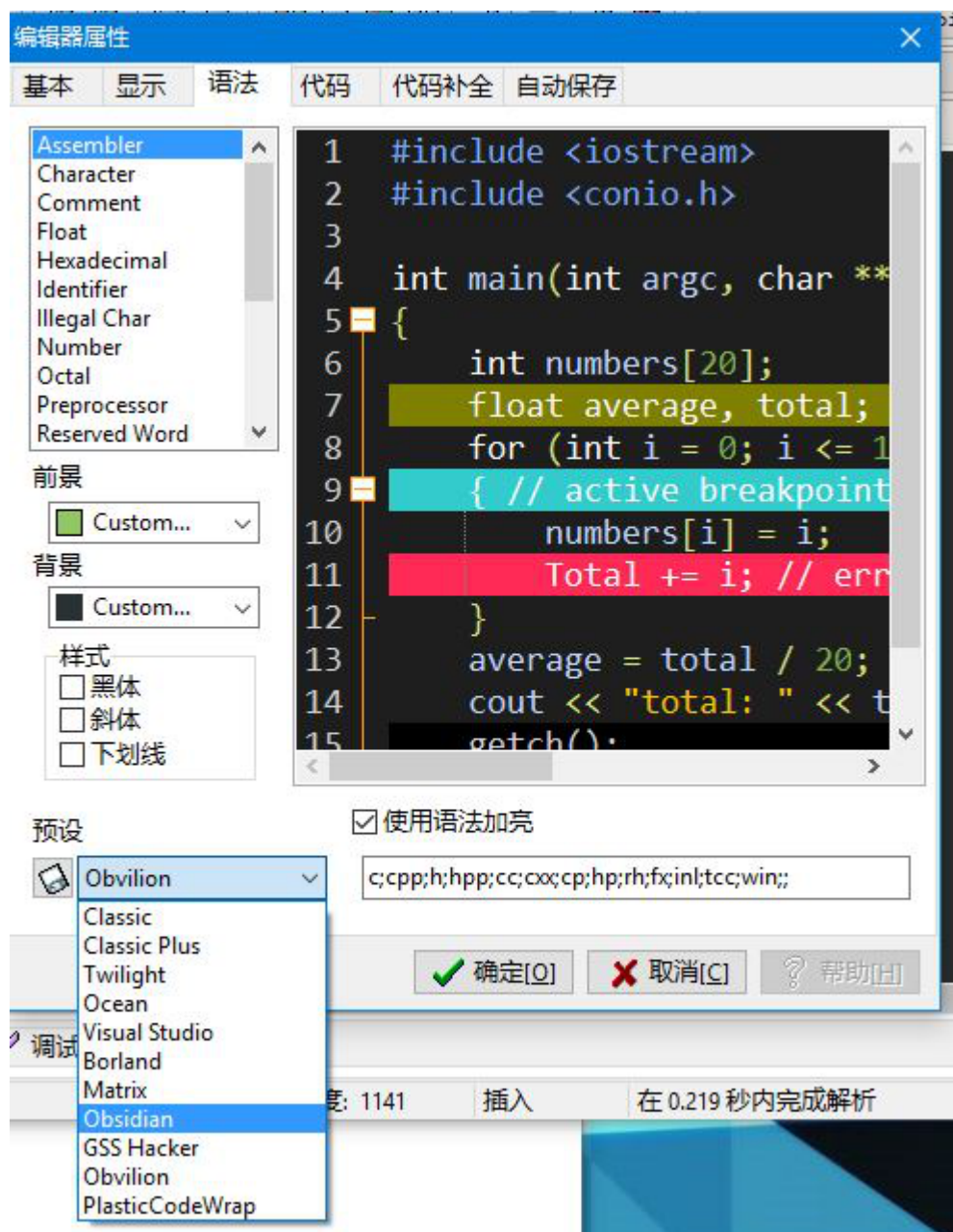
重点是确认工具栏有 AStyle 选项。

相信编译运行什么的同学们肯定都会了，我就不说了。

## 主题

好了那么可以看到，这个黑色主题是比较好看的。默认是白色，那么怎么把黑色主题调出来呢？

先打开：工具 -> 编辑器选项 -> 语法，看到那个『预设』了吗？选择 Obsidian 黑曜石主题，就是我那样了。



其他的主题同学们也可以试试。然后换完主题之后你会发现当前行会以一个非常非常亮的颜色高亮出来，十分辣眼睛。于是我们需要改一改当前行高亮。打开：工具 -> 编辑器选项 -> 基本，并像这样选择。



另外这个界面还有不少的 Preference 可以选择，比如你想让 DEV 帮你自动清除行尾的空格，就勾上去除结尾空白，然后显示函数提示和显示编辑器提示 坠吼勾上，非常好用，可以在你打代码的时候看到函数的形参列表。左下角还可以调整 Tab 大小和空格，不再赘述。

## 自动整理代码

屌粉（屌粉指 JetBrains 粉，像我这种的）都肯定很熟悉那个和 QQ 快捷键冲突的 `Ctrl + Alt + L` 自动整理代码的功能，简直神器啊，网上复制下来的丑陋不堪的代码，用这个工具一整，马上变得高度可读了。

其实 DEV C++ 也有一个自带的外部工具可以干这个事。比如你现在从网上复制一份题解代码：



```

13 auto q = *new std::queue<int>;
14 int i, j;
15 for (i = 1; i <= n; ++i) dis[i] = INF;
16 while (!q.empty()) q.pop();
17 dis[s] = 0;
18 q.push(s);
19 ++inq[s];
20 while (!q.empty()) {
21     i = q.front();
22     q.pop();
23     if (++visit[i] >= n) return true;
24     for (j = head[i]; ~j; j = next[j]) {
25         if (dis[target[j]] > dis[i] + value[j]) {
26             dis[target[j]] = dis[i] + value[j];
27             if (!inq[target[j]]) {
28                 ++inq[target[j]];
29                 q.push(target[j]);
30             }

```

看这丑陋的缩进，是不是感到一股绝望的气息？按下 **Ctrl+Shift+A**，你会看到 DEV 帮你整理好了代码。

```

19     q.push(s);
20     ++inq[s];
21     while (!q.empty()) {
22         i = q.front();
23         q.pop();
24         if (++visit[i] >= n) return true;
25         for (j = head[i]; ~j; j = next[j]) {
26             if (dis[target[j]] > dis[i] + value[j]) {
27                 dis[target[j]] = dis[i] + value[j];
28                 if (!inq[target[j]]) {
29                     ++inq[target[j]];
30                     q.push(target[j]);
31                 }
32             }
33         }
34         inq[i] = 0;




```

至于这个整理代码的工具的配置，可以打开 **AStyle -> 格式化选**

项，把括号风格改成别的（比如大括号不换行就是 Java，换行就是 Allman，别的都不用管，整理出来的代码辣眼睛）。另外我推荐的 switch case 语句的缩进方式是这样的：



```
switch (ass_we_can) {
case 1:
    puts("Ah I'm fucking coming");
    break;
case 2:
    break;
default:
    break;
}
```



如果你想这样的话，那么你可以把右上角缩进以下类型的代码中的 switch 给取消掉。

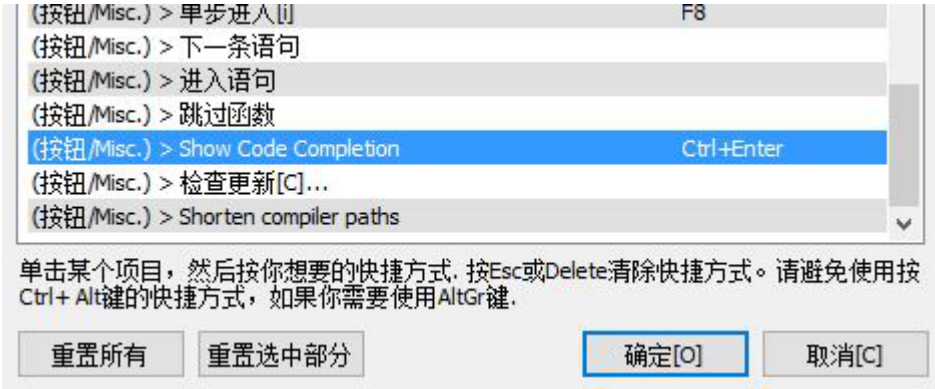
我把这个快捷键改成了  +  + ，和自己最熟悉的 JetBrains 家的快捷键一样。

## 代码补全

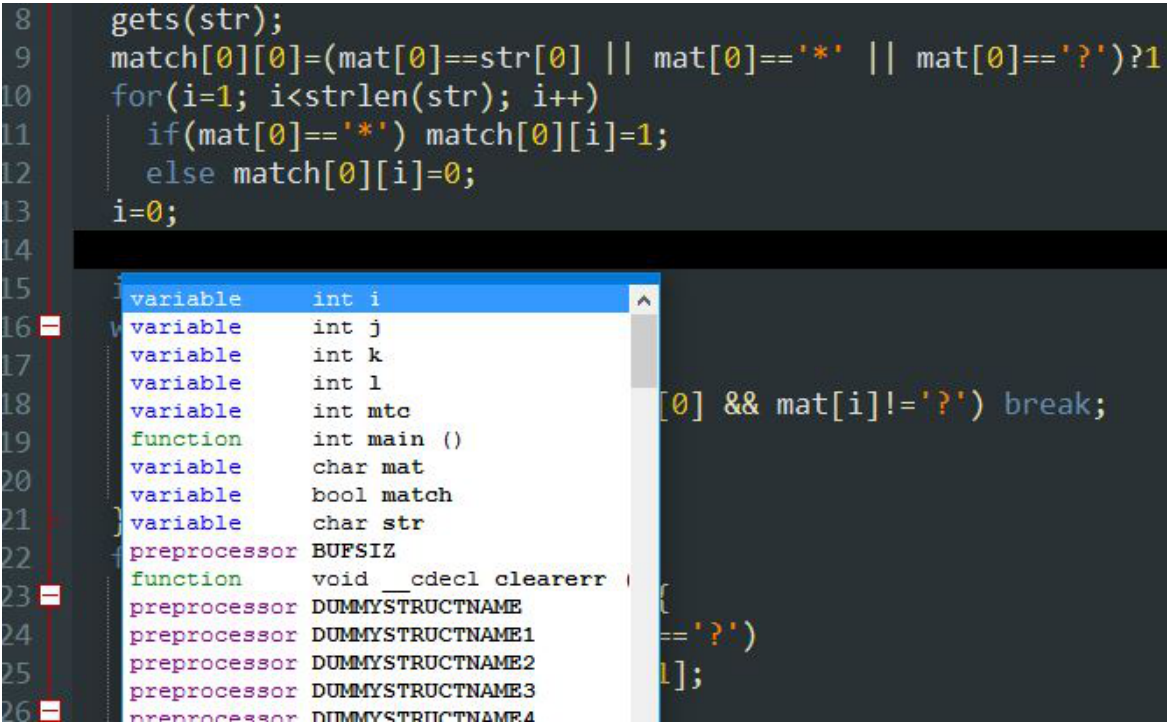
---

DEV 其实最新版已经支持了代码补全，只是因为缺省的快捷键是  + ，简直反人类啊（虽然和 JetBrains 家的默认是一样的），和切换输入法快捷键完美冲突。

打开工具 -> 快捷键选项，找到最下面的 **Show code completion**，把快捷键改成别的（我选择的是  + ，就是选中你要更改的那项，然后直接按下你想要的组合键，自动就改了），然后确定。



然后退出，在编辑器里面按下你设置的组合键，可以看到代码提示已经出来了。



这个代码提示是非常智能的，它会根据你包含的头文件和你光标前面的字符决定提示出来的东西。如果你只包含了 `stdio.h`，那么你先输入 `pr`，提示出来的就只有 `printf`。



但是如果你还包含了 `algorithm` 头文件，你再输入 `pr`，它再提示出来的就是 `prev` 了。

而且它只能提示出它已经缓存好了的头文件的内容，如果你在安装

的时候只缓存了常用头文件的话，你包含**bits/stdc++.h**，就提示不出来了。但是你定义的变量还是可以提示。

大部分 OI 常用的头文件都有提示的，放心吧。

它可以提示出来的有：

- 你定义的变量、函数名
- 函数形参
- 包含的已缓存头文件里的函数
- 对象实例的方法、构造方法
- 宏定义

看上去可以实际上不能提示的：

- 语言保留字，比如 `break`、`return`、`continue`
- 头文件
- 预处理语句

如果你想让它提示保留字出来的话，我有一个解决方案，那就是写这么一个宏定义（以 `return` 为例）：

```
#define return return
```

然后 `return` 就能被作为一个宏而提示出来了。

那么……

## 本垃圾说完了

---

祝大家用 DEV 用得愉快！



另外记住和平不 D 五项原则。

Tweet this



Top

创建一个 [issue](#) 以申请评论

Create an [issue](#) to apply for commentary

## 协议/License

本作品 [如何正确使用 DEV C++](#) 采用 [知识共享署名-非商业性使用-禁止演绎 4.0 国际许可协议](#) 进行许可，基于 <https://ice1000.org/2016/10/28/DevCpp/> 上的作品创作。  
This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](#).



© 2016-2019 Tesla Ice Zhang  
[RSS Feed](#)