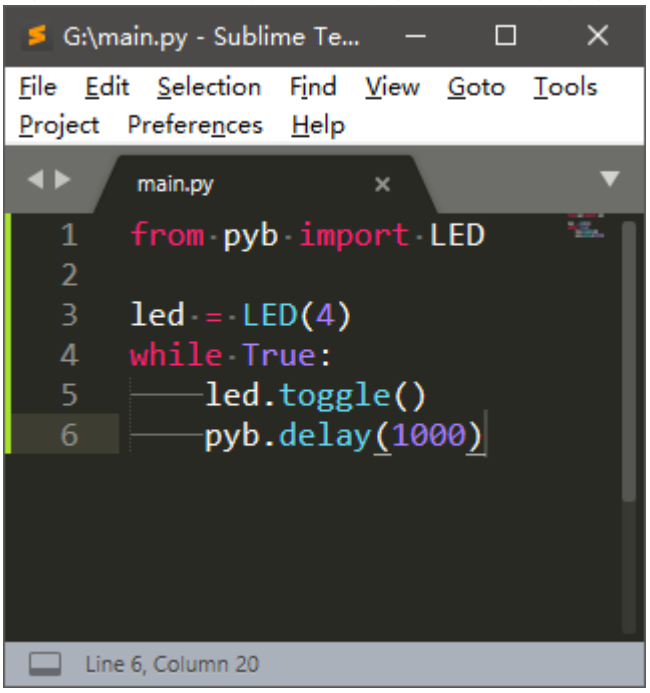
第 7 课，连续闪烁一个 LED 所面临的问题

本课学习内容：

- 通过 whlie(1):来实现循环闪烁 LED(4),但是注意这种方法对于 Pyhton 来说很 Low，记住 Python 语言可以面向过程，也是面向对象编程。C 语言只能面向过程编程。
- 一旦使用了 While True：，REPL 不能用了的原因和如何处理。

首先我们直接写这个功能的脚本吧！我们看看使用 REPL 运行会怎么样？



编写完这个脚本后，我们在 pyboard-tool 中进行了软启动，发现功能一切正常，第 4 个 LED 灯 1s 亮 1s 灭，看似很正常，没有什么毛病。但是你再点一次点击“**软启动**”时你会发现 REPL 好像不理你了。没有任何反应，发现了吗？这是因为系统“卡死”在了你脚本中的 whlie True：中无法响应你的复位指令，现在系统在埋头与死循环中干活。记下来该怎么做，有两个方法：回到 pyboard 生成的移动硬盘中，去掉这个死循环，保存好，复位就行，这时 Pyboard-tool 中按“软启动”还是不行，可以使用用另外两只复位方式才行，按一下 RST 按键或者重新上电复位，但这两种复位还是很麻烦的，在我们 Pyboard-tool 中有个强制 REPL 调试，先点这个，再点软启动，就恢复了 REPL 调试了。

分析上述这种情况的原因，这种直接使用 whlie True：方式不是不可以用，只是这种方式有点太自私了，独占了整个系统，对系统调用不够灵活，不建议使用。提醒记住，这种方法我们不推荐使用，也不建议使用，如果 Pyhton 只能这样安装面向过程编程，那 Python 只是比 C 语言更少的代码除此之外没有什么 NB 的价值，我们要的是面向对象编程，让那些成天打磨 C 语言致死都都不能面向对象编程的程序猿都哭去吧！我们已经开启了在 STM32 中面向对象编程的序幕。