```
                                         需要使用keys模块，结合send_keys()方法使用
                                                          send_keys(Key.ENTER*5)
                                         模拟回车的操作
                                                          send_keys(Key.TAB*3)
                                         模拟TAB键的操作
                          模拟键盘的操作                   send_keys(Key.BACK_SPACE*10)
                                         模拟删除键的操作
                                                          send_keys(Key.CONTROL,'a')
                                         模拟ctrl+a的操作
                                                          send_keys(Key.CONTROL,'c')
                                         模拟ctrl+c的操作
                                                          send_keys(Key.CONTROL,'v'*10)
                                         模拟ctrl+v的操作

                              三种等待的方式：
                                         sleep()
                                         等待设置中最简单的方法
                                                    不够智能
                              强制等待
                                         缺点      如果元素在指定时间范围内没有加载出来，程序会报错
                                                    如果页面加载很快，必须等够设置的时间，才能执行后续的脚本，浪费时间

                              静态等待
                              implicitly_wait()
                                                    设置一个最长等待时间
                                         原理      在时间范围内，如果页面加载完成，直接执行后续脚本，不会等够最长设置的时间
                              隐性等待                如果在最长等待时间范围内，页面还是没有加载完成，则报错

                                         缺点      框架兼容不是很好
                                                    浪费时间    操作页面元素没有必要等待页面所有元素全部加载完再操作

    Selenium_day3                         用法      在浏览器驱动的周期内，只需要设置一次即可

                              动态等待
                              WebdriverWait()
                              需要结合until或until_not方法使用
                                                          expect_contiditon
             等待的设置                                              1、判断符合某一个条件的所有元素是否全部加载出来，如果是则通过    presence_of_all_elements_located

                                                              2、判断元素的value值是否为期望的某一段文本，如果是则通过    text_to_be_present_in_element_value

                                                              3、判断某一个元素是否可以点击，如果可以点击则通过    element_to_be_clickable
                              显性等待     同时还需要构造一些等待判断的条件    判断条件
                                                              4、判断框架是否出现，如果出现则直接切入的方法    frame_to_be_available_and_switch_to_it

                                                              5、判断符合某一个条件的第一个元素是否加载出来，如果是则通过    presence_of_element_located

                                                              6、判断元素上是否出现某一点文本，如果是则通过    text_to_be_present_in_element

                                                              7、判断警告窗口是否出现，如果出现则直接切入    alert_is_present

                                                    需要设置一个最长等待时间
                                         原理      每间隔0.4s去检查判断的场景是否成立，如果成立，则执行后续脚本
                                                    如果超出最长等待时间场景还是不成立，也会报错

             函数的调用
```