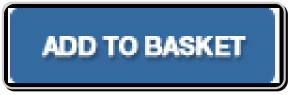
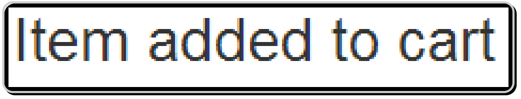
```
# works on all platforms
import os
import shutil
import time
# get the directory containing your running .sikuli
myPath = os.path.dirname(getBundlePath())
if not myPath in sys.path: sys.path.append(myPath)

scrDir = "C:\Users\Public\Pictures\SikluiScreenshots"
logDir = "C:\Users\Public\\Pictures\SikluiScreenshots\Log.txt"

#This fucntion writes the test result to a file in the screenshot folder
def writeResult( testName, result ):
    localtime = time.localtime(time.time() )
    timeString = time.strftime ('%Y/%m/%d - %H:%M:%S')
    inp=open( logDir , 'a')
    inp.write( '( %s )' %timeString )
    inp.write('%s - ' %testName )
    inp.write( '%s \n' %result )
    inp.close( )

#Function to find the "Add to basket" button in the webpage and
```

```
#wait for the confirmation.
def addItem():
        click(Pattern().similar(0.80))
        wait( 5 )
        wait(Pattern().similar(0.90),60 )

#Function to open the specified internet browser
def  openBrowser(browserType):
    appObj = None

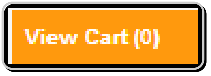
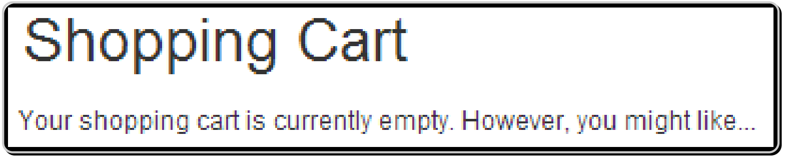
    if( browserType == "IE"):
        appObj = App.open( "C:\Program Files (x86)\Internet
Explorer\iexplore.exe")
    elif( browserType == "FireFox"):
        pass
    elif( browserType == "Chrome"):
        appObj = App.open( "C:\Program Files
(x86)\Google\Chrome\Application\chrome.exe" +  " --start-maximized")
    else:
        raise "openBrowser: unknown browser type=" + browserType

    if appObj == None:
```

Item added to cart

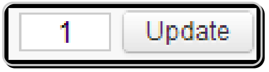
```
        raise "openBrowser: browserType=" + browserType + " appObj == None"

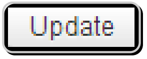
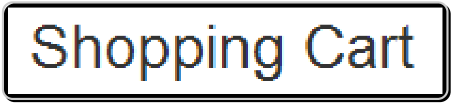
    return appObj

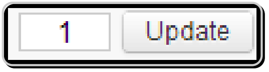
#Testing system functionality to detect an empty shopping cart
def emptyCart( dir ):
    screenshotsDir = dir
    wait( 2 )
    img = capture(SCREEN)
    shutil.move(img, os.path.join(screenshotsDir, ))
    click( find ( 
View Cart (0)
 ) )
    wait( 5 )
    img = capture(SCREEN)
    if exists( 
Shopping Cart
Your shopping cart is currently empty. However, you might like...
 ):
        shutil.move(img, os.path.join(screenshotsDir, ))
        writeResult( "emptyCard" , "Succeded" )
    else:
        shutil.move(img, os.path.join(screenshotsDir, ))
        writeResult( "emptyCard" , "Failed" )

#Testing system functionality to remove an item from the shopping cart
```

```
def removeItem( dir, app):
    screenshotsDir = dir
    addItem()
    addItem()
    addItem()
    addItem()
    click( find(View Cart) )
    wait(Shopping Cart, 60)
    img = capture(SCREEN)
    shutil.move(img, os.path.join(screenshotsDir, 🖼))
    prev = Region(701,415,442,59).text()
    remove = Region(701,415,442,59).right( 600 ).find(Pattern(X
).similar(0.80))
    click( remove )
    wait( 10 )
    current = Region(701,415,442,59).text()
    img = capture(SCREEN)
    if prev!=current:
        shutil.move(img, os.path.join(screenshotsDir, 🖼))
        writeResult( "removeItem", "Succedded" )
    else:
        shutil.move(img, os.path.join(screenshotsDir, 🖼))
        writeResult( "removeItem" , "Failed" )

#Testing system functionality to change the quantity of an item
def modifyQuantity( dir, app):
    img = capture(SCREEN)
    shutil.move(img, os.path.join(dir, ))
    quantity = find( Pattern(1 Update).targetOffset(-30,-3) )
    click( quantity )
    type( Key.DELETE )
    wait(2)
    type( Key.BACKSPACE )
    wait(2)
    type( "4" )
    prev = quantity.right( 500 ).text()

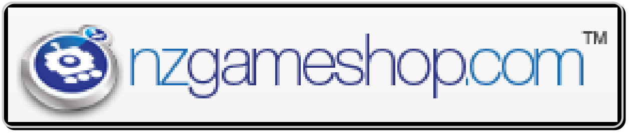
    quantity.click( Update )
    wait( Shopping Cart )
    wait(5)
    current = quantity.right( 500 ).text()
    img = capture(SCREEN)
    if prev!=current:
        shutil.move(img, os.path.join(dir, ))
        writeResult( "modifyQuantity" , "Succeded" )
    else:
        shutil.move(img, os.path.join(dir, ))
        writeResult( "modifyQuantity" , "Failed")

#Testing system functionality to detect a quantity set to zero
def changeQuantity0( dir, app ):
    screenshotsDir = dir
    img = capture(SCREEN)
    shutil.move(img, os.path.join(screenshotsDir, ))
    r = Region(643,401,515,254)
    prev = r.text()
    quantity = r.right( 500 ).find(Pattern(
).targetOffset(-31,-1))

    click(quantity)
    type( Key.DELETE )
    wait(2)
    type( Key.BACKSPACE )
    wait(2)
    type( "0" )
    quantity.click( Pattern( ).similar(0.91) )
    wait(5)
    current = r.text()
    if prev!=current:
        img = capture(SCREEN)
        shutil.move(img, os.path.join(screenshotsDir, ))
        writeResult( "changeQuantity0" , "Succedded" )
    else:
        img = capture(SCREEN)
        shutil.move(img, os.path.join(screenshotsDir, ))
        writeResult( "changeQuantity0" , "Failed" )
    click(Pattern(Clear Cart).similar(0.81))

#Fucntion to close whichever browser that is open
def  closeBrowser( app):
        closeButton = find( X )
        closeButton.highlight( 5 )
        click( closeButton )

#Function to run the four tests above
def runTests( app):
    emptyCart( scrDir )
    removeItem( scrDir, app )
    modifyQuantity( scrDir, app )
    changeQuantity0( scrDir, app )

#Function to open a browser and run tests
def testWithBrowser( browser):
    try:
        appS = openBrowser( browser)
        wait( 5 )
        type( 'd', KEY_ALT )
        type( "www.nzgameshop.com" + Key.ENTER )
        wait( Pattern().similar(0.72) , 60 )
        runTests( appS )
        closeBrowser( appS )

    except Exception,e:
        print "testWithBrowser: caught exception", e
#Main function
def main():
    #testWithBrowser( "IE" )
    testWithBrowser( "Chrome")

main()
```