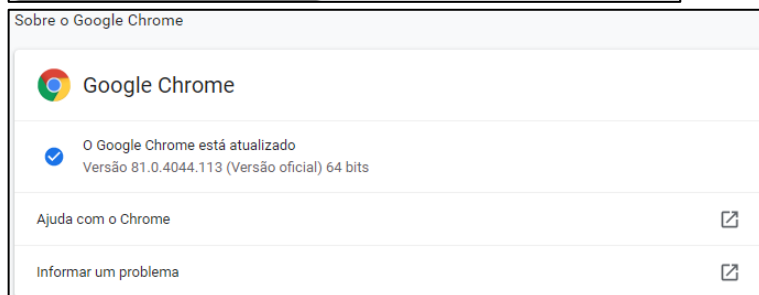
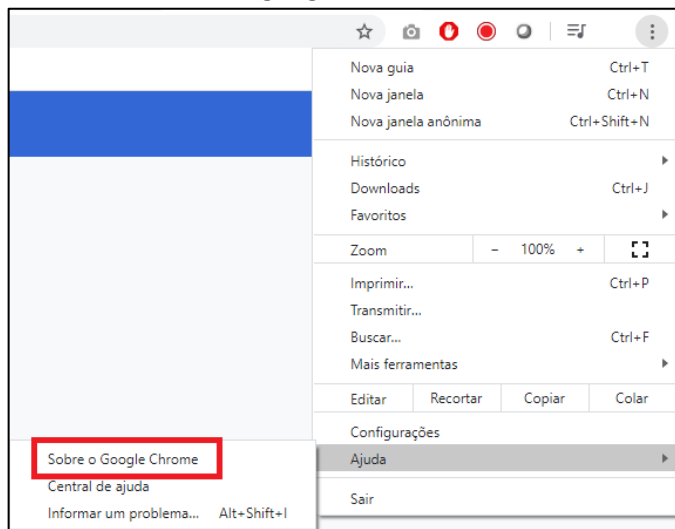


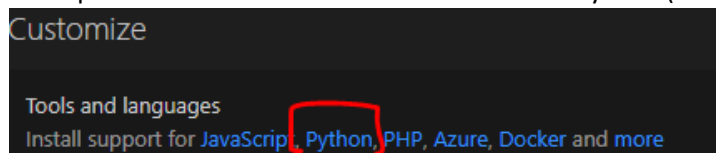
WHATSAPP BOT PROJECT

Preparing VS Code, Python and Selenium on Windows

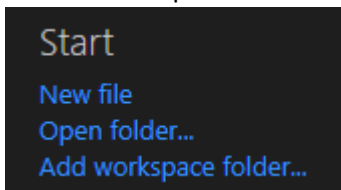
1. Install Python 3.8.2 from <https://www.python.org/>
 2. Check the box "Add Python to environment variables"
- Open cmd.exe
 1. Pip install selenium (to navigate chrome automatically)
 2. Pip install pandas (to read contacts from CSV)
 3. If it doesn't work, check the environment variables step (PATH)
 - Install VS Code from <https://code.visualstudio.com/>
 - Install google chrome webdriver from <http://chromedriver.chromium.org/downloads>
 1. ChromeDriver version must match google chrome's version
 2. To check google chrome version:



- Unzip chromedriver.exe to this folder (same folder as the WhatsappBot.py file)
- Open VS Code and clicked on customize → Python (for first users)



- Start → Open Folder → Navigate to this folder



WhatsappBot.py code – Step by step execution description

- Import selenium's webdriver, time, pandas and os

```
# -----  
# Import libraries  
# -----  
  
print("-----")  
print("Whatsapp Bot V1.0")  
print("-----")  
print("Importing libraries")  
  
from selenium import webdriver          # import the navigator  
from selenium.webdriver.common.keys import Keys # Keys library (to use SHIFT+ENTER)  
import time                             # time library to add delays between commands  
import pandas as pd                     # pandas library to read CSV files (with whatsapp contacts)  
import os                               # file and directory management  
import sys                             # Will be used to flush the debug window
```

- Write custom message and leave a `_NAME_` key to replace with contacts' names

```
# -----  
# Main message (customizable)  
# -----  
  
print("Storing main message")  
  
mainMsg = ["Teste de mensagem personalizada.", "Nome: _NAME_"]  
#mainMsg = "" #uncomment this if you don't want messages, just attachments
```

- Import CSV file with names and phone numbers and convert the generated dataframe to string

```
# -----  
# Import contacts csv as dataframe and convert to string  
# -----  
  
print("Importing CSV")  
  
csvFileName = "Contacts.csv"  
contactsDF = pd.read_csv(csvFileName, error_bad_lines = False, sep=';')  
contactsDF = contactsDF.applymap(str)  
nContacts = len(contactsDF.index)
```

- Get names of all images on images folder

```
# -----  
# List all images to attach (if there are no images on the images folder, the code will not attach anything)  
# -----  
  
print("Listing attachments")  
  
nImages = 0  
mainPath = os.path.dirname(os.path.abspath(__file__)) # path of current file  
imgPath = mainPath + "\\Images"                      # path of images folder within current folder  
imgFiles = []                                         # initialize empty list of images  
  
for r,d,f in os.walk(imgPath):  
    for file in f:  
        imgFiles.append(imgPath + "\\" + file)  
        nImages = nImages + 1
```

- Create 2 functions: "clean" phone numbers and retrieve the contacts' first names

```
# -----  
# Function to leave only numbers of string and also remove leading zeroes  
# -----  
  
print("Declaring user defined functions")  
  
def onlyNumbers(fullString):  
    a = ""  
  
    for i in range(len(fullString)):  
        for k in range(10):  
            if fullString[i] == k or fullString[i] == str(k):  
                a = a + fullString[i]  
  
    finalStr = a.lstrip('0')  
    return finalStr  
  
# -----  
# Function to extract first name  
# -----  
  
def firstName(fullName):  
    return fullName.split()[0]
```

- Create a webdriver object and navigate to whatsapp web

```
# -----
# Webdriver object
# -----

print("Opening chrome")

driver = webdriver.Chrome(executable_path = r'./chromedriver.exe')

# -----
# Navigate to whatsapp web
# -----

print("Accessing whatsapp web")

driver.get('https://web.whatsapp.com')
```

- Wait 15 seconds for user to scan QR code

```
# -----
# Wait 15 seconds for user to log in using QR code
# -----

print("Waiting for QR code log-in")

for i in range(15,0,-1):
    sys.stdout.write(str(i)+' ')
    sys.stdout.flush()
    time.sleep(1)
```

- Loop all contacts

```
# -----
# Loop all contacts in the CSV file
# -----

print("Looping contacts")

st = 4      # sleep time in seconds

# Begin loop
for i in range(len(contacts)):

    # Get name, first name and phone number
    cName = contactsDF.at[i,'Nome']
    cFirstName = firstName(cName)
    cNumber = onlyNumbers(contactsDF.at[i,'Telefone'])

    print("-----")
    print(cName)
    print(cNumber)
    print("-----")
```

- Find the html tag related to the WhatsappWeb searchbox and type the number

```
<div class="_2cLHw">Search or start new chat</div>
<label class="_2MSJr">
  <div tabindex="-1" class="_3F6QL _3xlwb">
    <div class="_39LWd" style="visibility: visible"></div>
    <div class="_2S1VP copyable-text selectable-text"
      contenteditable="true" data-tab="3" dir="ltr">...</div> == $0
  </div>
```

```
# Get search box
# <div class="_2S1VP copyable-text selectable-text" contenteditable="true" data-tab="3" dir="ltr"></div>
print("Finding searchbox and writing contact's number")

searchBox = driver.find_element_by_class_name('_2S1VP')
searchBox.click()
time.sleep(st)

searchBox.send_keys(cNumber)
time.sleep(st)
```

- Find the html tag related to the contact using its full name EXACTLY as it is on the phone's contacts list and click it

```

▶<div class="_2wP_Y" style="z-index: 2; transition: none 0s ease 0s; height: 72px; transform: translateY(144px);">...</div>
▶<div class="_2wP_Y" style="z-index: 25; transition: none 0s ease 0s; height: 72px; transform: translateY(0px);">...</div>
▼<div class="_2wP_Y" style="z-index: 0; transition: none 0s ease 0s; height: 72px; transform: translateY(72px);">
  ▼<div tabindex="-1">
    ▼<div class="_2EXPL">
      ▶<div class="dIyEr">...</div>
      ▼<div class="_3j7s9">
        ▼<div class="_2FBdJ">
          ▼<div class="_250oe">
            ▼<span class="_3TEwt">
              <span dir="auto" title=" " class="_1wjpf _3Nfp9 _3FX81">...</span> == $0

```

```

# Click on the result (will use complete full name from contact list)
print("Clicking on the contact")

contactBtn = driver.find_element_by_xpath(f"//span[@title='{cName}']")
contactBtn.click()
time.sleep(st)

```

- Find the html tag related to the message box, write the customized message and send it

```

▼<div class="_3pkxz V42si copyable-area">
  ▶<div class="weEq5" style="will-change: width; ">...</div>
  ▼<div tabindex="-1" class="_1Plpp" == $0
    ▼<div tabindex="-1" class="_3F6QL _2WovP">
      <div class="_39Lwd" style="visibility: visible; ">Type a message</div>
      <div class="_2S1VP copyable-text selectable-text" contenteditable="true" data-tab="1" dir="ltr" spellcheck="true"></div>
    </div>
  </div>
  ▶<div class="weEq5">...</div>
</div>

```

```

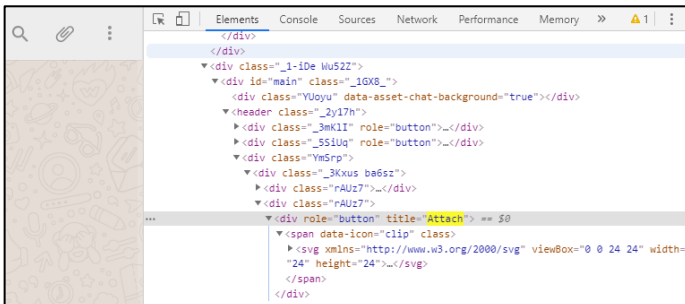
# Create custom message and send the message
if len(mainMsg) > 0:
    # Get msgBox reference
    print("Finding the message box and typing the custom message")
    msgBox = driver.find_element_by_class_name('_1Plpp')
    msgBox.click()
    time.sleep(st)

    # Write message line by line
    for msg in mainMsg:
        msgBox.send_keys(msg.replace("_NOME_", cFirstName))
        msgBox.send_keys(Keys.SHIFT + Keys.ENTER)

    # Press enter (easier than looking for the "send button")
    time.sleep(st)
    msgBox.send_keys(Keys.ENTER)
    time.sleep(st)

```

- Find the html tag related to the attachment button and click it



```

</div>
</div>
▼<div class="_1-iDe Vu52Z">
  ▼<div id="main" class="_1GX8">
    <div class="YUoyu" data-asset-chat-background="true"></div>
    ▼<header class="_2y17h">
      ▶<div class="_3mKlI" role="button">...</div>
      ▶<div class="_5SiUq" role="button">...</div>
      ▼<div class="Ym5rp">
        ▼<div class="_3Kxus b06sz">
          ▶<div class="rAUz7">...</div>
          ▼<div class="rAUz7">
            ▼<div role="button" title="Attach" == $0
              ▼<span data-icon="clip" class=">
                ▶<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" width="24" height="24">...</svg>
              </span>
            </div>

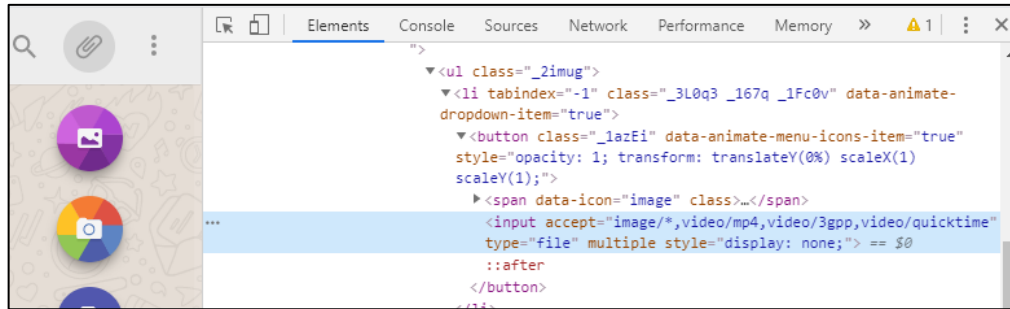
```

```

# Send attachments
print("Sending attachments")
if nImages > 0:
    # Get first attachment button (clip shaped) to click and open attachment options
    clipBtn = driver.find_element_by_xpath("//div[@title='Attach']")
    clipBtn.click()
    time.sleep(st)

```

- Find the html tag related to the “photos and videos”. And loop send_keys with the files’ locations



```
# Get input element
imageBox = driver.find_element_by_xpath("//input[@accept='image/*,video/mp4,video/3gpp,video/quicktime']")

# Attach images (loop)
imgStr = ""
ic = 0
for imgFile in imgFiles: # create an imgStr string with all image file paths separated by a new line (\n)
    ic = ic + 1
    if ic == 1:
        imgStr = imgFile
    else:
        imgStr = imgStr + "\n" + imgFile

imageBox.send_keys(imgStr)
time.sleep(st)
```

- Find the html tag related to the “send attachment” button and click it



```
# Send attachments
sendAttachments = driver.find_element_by_xpath("//span[@data-icon='send-light']")
sendAttachments.click()
```

Sources used

- Whatsapp bot (basic, just sends messages to named groups): <https://www.youtube.com/watch?v=ISYHWfWvp3E>
- Upload files with selenium and webdriver (to send attachments): <https://www.youtube.com/watch?v=ffHC36iTNRg>
- Attach files on whatsapp web <https://www.youtube.com/watch?v=x85ba0VYsYo>