Day 50 Auto Tinder bot

<https://www.nytimes.com/2014/10/30/fashion/tinder-the-fast-growing-dating-app-taps-an-age-old-truth.html>

<https://tinder.com/>

<https://www.thispersondoesnotexist.com/> auto generates a picture of not a real person.

Step 1 - Setup your account on Tinder

1. If you don't already have an account on [Tinder](https://tinder.com/), set one up now. Make sure you can sign in to your account using Facebook or Google.

If you don't want to use your own details, ask a friend who is on Tinder or just set up a new [Facebook](https://www.facebook.com/) account and use an AI-generated image from <https://www.thispersondoesnotexist.com/>

HINT: You can hit refresh on thispersondoesnotexist to generate new random images.

2. Manually go through the process of swiping on profiles and see which elements you'll need to target with your code.

NOTE: If you don't want to disappoint anyone (I'm sure you are very attractive/good dating material 😜), you can always complete the tutorial and hit "NOPE" on everyone. Least you break people's hearts by matching with them and then telling them "Sorry I just did it for a Python tutorial" 😢.

Step 2 - Navigate to Login Page

In order to avoid dual verification with a phone every time we log in, we'll need to use the Facebook/Google login. The Google login flow requires a lot more steps than Facebook login, so we'll go with Facebook in this challenge.

1. Using Selenium and Python Navigate to the Tinder website (<https://tinder.com/>) and click on LOG IN then LOGIN WITH FACEBOOK. See below:

If successful, you should see a pop-up for Facebook login:

HINT 1: Make sure you've already manually signed-in and verified your phone number with Tinder as we can't automate the phone number verification. You only have to do this once.

HINT 2: If you are getting a NoSuchElementException, make sure you've added some delay between clicking on buttons so that the new element has enough time to load.

HINT 3: You might find it easier to right click on the element and get the XPath to use with Selenium. e.g.

Step 3 - Login with Facebook

1. The Facebook login page opens in a new window. In order for our selenium code to work on the new window, we have to switch to the window in front.

In Selenium, each window has a identification handle, we can get all the window handles with:

1. driver.window\_handles

The above line of code returns a list of all the window handles. The first window is at position 0 e.g.

1. base\_window = driver.window\_handles[0]

New windows that have popped out from the base\_window are further down in the sequence e.g.

1. fb\_login\_window = driver.window\_handles[1]

We can switch our Selenium to target the new facebook login window by:

1. driver.switch\_to.window(fb\_login\_window)

You can print the driver.title to verify that it's the facebook login window that is currently target:

1. print(driver.title)

The full code to switch to the new pop-up window is thus:

1. base\_window = driver.window\_handles[0]
2. fb\_login\_window = driver.window\_handles[1]
3. driver.switch\_to.window(fb\_login\_window)
4. print(driver.title)

If successful the printed title should be "Facebook" and not "Tinder | Match. Chat. Date."

2.  Using what you have learnt about Selenium, fill in the Facebook login form and submit it to log in.

NOTE: Avoid invoking the Facebook Login too frequently, see if you can test your code without logging in, you don't want to appear like a bot to Facebook as there is always the chance that they might disable your FB account. Alternatively, you can try setting up an alternative Facebook account.

If successful, you should see the pop-up window disappear and you're back on the Tinder page. e.g.

3. At this point, you should revert back to the base\_window and verify by printing the title of the Selenium controlled window title.

1. driver.switch\_to.window(base\_window)
2. print(driver.title)

If successful, it should print "Tinder | Match. Chat. Date."

Step 4 - Dismiss all requests

When you first login to Tinder, it will ask if it's ok to get your location, send you notifications and track your cookies. We need to dismiss all of these modal pop-ups.

Location Pop-up:

Notification Pop-up:

Cookies Pop-up:

1. Using Selenium and Python:

- Click ALLOW for location.

- Click NOT INTERESTED for notifications.

- Click I ACCEPT for cookies

HINT 1: Finding the XPath will make it easier to target each element.

HITN 2: Adding some delay before targeting these elements will allow time for them to load up.

Step 5 - Hit Like!

1. The final step is to like some people. Because it's the web version, we don't have to swipe right, all we need to do is to click on the "Like" button. You'll want to add **at least a 1 second delay** between "Likes" so that Tinder doesn't block you because you seem like a bot.

HINT 1: It takes a while for Tinder to load up people near you, this is not a fixed time as it depends on a number of factors. When it's loading, the "Like" button will not be reachable and you will get a NoSuchElementException if you try to find it. Use exception handling to handle this situation and wait 2 seconds before you retry.

HINT 2: Sometimes, as you are swiping, you'll get a match which is a pop-up on the same page. But this will mean that your Like button will be hidden behind the pop up and you'll get a ElementClickInterceptedException. e.g.

You'll need to dismiss this by clicking on "BACK TO TINDER" to continue swiping.

NOTE: On the free tier, Tinder only allows 100 "Likes" per day.

[SOLUTION](https://gist.github.com/angelabauer/5c47e3b2d3f8d6a80c8c92d2892774be)