

# 파이썬을 활용한 업무자동화

9회차: 컴퓨터 자동화로 업무자동화 완성하기

# 목차

9회차: 자동으로 웹사이트 다루기 - 심화 (2)/ 키보드, 마우스 제어 - 기본

- 복습
- 웹 자동화 심화: xpath 심화
  - ▶ 11번가 상품 검색 및 리스트 가져오기
- 키보드, 마우스 제어하기: 기본
  - ▶ 기본 라이브러리 소개

지난 강의 때 뭐했지..?

## ■ 자동으로 웹 사이트 다루기 (심화)

- 웹 자동화 심화: 마우스를 활용
  - ▶ 인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기
- 웹 자동화 심화: iframe 다루기
  - ▶ 중고나라 검색결과 가져오기

```
# 위에서 찾은 링크들에 대해 반복문을 돌면서 클릭
ac = ActionChains(driver)
for post in posts:
    # 해당 요소로 이동하여 클릭
    ac.reset_actions()
    ac.move_to_element(post)
    ac.click()
    ac.perform()

    # 게시글이 뜰때까지 기다림
    time.sleep(1)

    try:
        # 좋아요를 링크 텍스트로 찾아서 누름
        # 만약 이미 좋아요가 눌러진 글이라면 해당 링크 텍스트가 없음
        # 따라서 try를 이용하여 처리
        like_btn = driver.find_element_by_link_text('좋아요')
        like_btn.click()
        # 좋아요가 반영될때까지 기다림
        time.sleep(1)
    except:
        print('이미 좋아요를 누른 게시글입니다.')
        like_btn = driver.find_element_by_link_text('좋아요 취소')
        like_btn.click()

    # ESC 키를 눌러 게시글을 닫음
    ac.reset_actions()
    ac.send_keys(Keys.ESCAPE)
    ac.perform()

    time.sleep(1)
```

```
# 검색 결과를 감싸고 있는 iframe 가져오기
iframe = driver.find_element_by_id('cafe_main')

# iframe 안으로 들어가기
driver.switch_to_frame(iframe)

# 원하는 내용을 가져오기 위해 상위의 name을 가진 form을 가져옴
form = driver.find_element_by_name('ArticleList')

# 최하위에 숨어있는 tr을 피하기 위해 xpath로
# 원하는 tr을 명시해서 가져옴
rows = form.find_elements_by_xpath('./table/tbody/tr')
for row in rows:
    try:
        # 제목이 a태그로 표현되어 있어, a 태그를 가져옴
        # tr 태그중에 게시글이 아닌 하나의 선을 표현한 태그가 있기 때문에
        # 해당 태그를 건너뛰기 위해 try를 시도
        # 선을 표현한 태그는 하위에 a태그를 가지고 있지 않기 때문에
        a_tag = row.find_element_by_tag_name('a')
        print(a_tag.text)
    except:
        # 데이터가 없는 경우 반복문을 스킵
        continue

# 브라우저가 바로 꺼지지 않도록 input()을 추가
input()
except Exception as e:
    print(e)
finally:
    driver.quit()
```

# 웹 자동화 심화: XPATH 심화

04

11번가 상품 검색 및 리스트 가져오기

## ■ XPATH 란?

- XPATH는 XML이라는 문서 형태에서 사용하는 경로 표기법
- 하지만 XML뿐만 아니라 HTML에서도 XPATH를 활용할 수 있음
- 이전 시간에 배웠던 것과 같이 단순히 /path/path/path 외에 속성이 많음
  - ▶ 문자열을 포함하는 태그 찾기
  - ▶ id, class, name 외에 속성으로 찾기
  - ▶ 속성값의 일부만 일치하는 조건으로 찾기

# 웹 자동화 심화: XPATH 심화

05

11번가 상품 검색 및 리스트 가져오기

## ■ 자동화 단계 분석

1. 11번가 사이트 접속

2. 검색어 입력

3. 검색 결과 리스트 가져오기

1. id의 일부만 일치하는 요소 가져오기

2. '고객응대우수' 판매자인지 확인하기

4. 모든 판매자 리스트 가져오기

1. 특정 속성값으로 태그 찾기

# 웹 자동화 심화: XPATH 심화

06

11번가 상품 검색 및 리스트 가져오기

## ■ 0단계: Selenium 설정

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

driver = webdriver.Chrome('./chromedriver')
```



# 웹 자동화 심화: XPATH 심화

07

11번가 상품 검색 및 리스트 가져오기

## ■ 1단계: 11번가 사이트 접속

### 1.11번가에 접속!

```
try:  
    # 11번가로 접속  
    driver.get('http://11st.co.kr')
```

# 웹 자동화 심화: XPATH 심화

08

11번가 상품 검색 및 리스트 가져오기

## 2단계: 검색어 입력

### 1. 검색어를 초기화

### 2. '자전거' 검색어 전달

```
# 검색어 입력을 위해 kwd를 가져옴
elem = driver.find_element_by_name('kwd')
# 검색어 초기화한 뒤 자전거 입력
elem.clear()
elem.send_keys('자전거')
elem.send_keys(Keys.RETURN)
```

이선반 베란다창고정리!	<input type="hidden" name="adKwdIrcNo" id="adKwdIrcNo" value="1201711224883479455">
	<input type="hidden" name="adPrdNo" id="adPrdNo" value="191763407">
	<input type="hidden" name="targetTab" value="T">
	▼<fieldset id="tSearch">
	<legend>통합검색</legend>
SHOP 백화점 브랜드 패션 뷰티 ...	<input type="text" class="header_inp_txt" title="11번가 통합검색" name="kwd" value id="AKCKwd" onfocus="clearAdUrl();searchManager.init(true);"> == \$0

# 웹 자동화 심화: XPATH 심화

09

11번가 상품 검색 및 리스트 가져오기

## 3단계: 검색 결과 리스트 가져오기

1.id에 thisClick이 포함된 태그를 가져옴

1.xpath: `//li[contains(@id, 'thisClick')]`

2.가져온 리스트에서 제목, 금액, 판매자 부분을 가져옴

어버이날 선물	<pre>&gt;&lt;div class="top_wrap"&gt;...&lt;/div&gt; ▼&lt;div class="total_listing_wrap"&gt;   ▼&lt;ul class="tt_listbox" data-log-actionid-area="hot"&gt;     ... ▼&lt;li id="thisClick_1510196699" class=" ad_sec "&gt; == \$0       &lt;div data-tracking-code="SA11I7201607274240463016"&gt;&lt;/div&gt;       ▼&lt;div class="total_listitem"&gt;         ▶&lt;div class="photo_wrap" name="prdPhoto"&gt;...&lt;/div&gt;         ▶&lt;div class="list_info"&gt;...&lt;/div&gt;         ▶&lt;div class="list_price"&gt;...&lt;/div&gt;         ▶&lt;div class="list_benefit"&gt;...&lt;/div&gt;       &lt;/div&gt;     &lt;/li&gt;     ▶&lt;li id="thisClick_433893654" class=" ad_sec "&gt;...&lt;/li&gt;     ▶&lt;li id="thisClick_1261829258" class=" ad_sec "&gt;...&lt;/li&gt;     ▶&lt;li id="thisClick_1265381234" class=" ad_sec "&gt;...&lt;/li&gt;     ▶&lt;li id="thisClick_1724601078" class=" ad_sec "&gt;...&lt;/li&gt;     ▶&lt;li id="thisClick_1594744351" class=" ad_sec "&gt;...&lt;/li&gt;     ▶&lt;li id="thisClick_1701981556" class=" ad_sec "&gt;...&lt;/li&gt;</pre>
클 등 자전거	

# 웹 자동화 심화: XPATH 심화

10

11번가 상품 검색 및 리스트 가져오기

## ■ 3단계: 검색 결과 리스트 가져오기

1.id에 thisClick이 포함된 태그를 가져옴

1.xpath: `//li[contains(@id, 'thisClick')]`

2.가져온 리스트에서 제목, 금액, 판매자 부분을 가져옴

```
# id가 thisClick을 포함하는 모든 태그를 가져옴
results_list = driver.find_elements_by_xpath('//li[contains(@id,
\'thisClick\')]\')

# 위 가져온 태그들에 대해 반복
for elem in results_list:
    # 제목, 금액, 판매자 부분을 가져옴
    title_tag = elem.find_element_by_class_name('info_tit')
    price_tag = elem.find_element_by_class_name('sale_price')
    mall_tag = elem.find_element_by_class_name('benefit_tit')
    print("[%s] %s : %s"%(mall_tag.text, title_tag.text, price_tag.text))
```

# 웹 자동화 심화: XPATH 심화

11

11번가 상품 검색 및 리스트 가져오기

## 3단계: 검색 결과 리스트 가져오기

1. 태그 안에 고객응대우수 라는 문자가 포함된 span태그를 찾음

1. 해당 태그가 없을 경우 에러가 나기 때문에 try로 감싸야함

2. xpath: `./span[text()='고객응대우수']`

```
# 태그 안에 고객응대우수 라는 문자가 포함된 span태그가 있는 경우
# 고객응대우수 판매자를 출력하고, 그렇지 않은 경우 아무 동작하지않음
try:
    span = elem.find_element_by_xpath(' ./span[text()=\ '고객응대우수\ ' ] ')
    print("고객응대우수 판매자입니다.")
except:
    pass
```

# 웹 자동화 심화: XPATH 심화

12

11번가 상품 검색 및 리스트 가져오기

## 4단계: 모든 판매자 리스트 가져오기

1. data-log-actionid-label 속성값이 sellername인 태그 가져오기

1.xpath: //a[@data-log-actionid-label='sellername']

건강저울나라	<pre>&lt;p class="benefit_tit"&gt;   &lt;a href="#" data-log-actionid-label="sellername"     data-log-body="{\"content_no\":     \"1510196699\", \"content_type\":     \"PRODUCT\", \"content_name\": \"모야2S 3초 접이식 휴대용 전기 자     전거 어버이날 선물\", \"trc_no\":     \"7201607274240463016\", \"ad_yn\": \"Y\", \"ad_typ_gubn\":     \"S\", \"ad_area_gubn\": \"A11\", \"ad_rank\": \"1\", \"product_no\":     \"1510196699\", \"product_name\": \"모야2S 3초 접이식 휴대용 전기     자전거 어버이날 선물\", \"product_price\":     \"1490000\", \"seller_no\": \"MTAyNDA10TgwMTY\", \"ad_type\":     \"HOT\", \"display_order\": 1}" data-log-index="1"     onclick="try{rakeLog.sendRakeLog(this);}catch(e){}     ga('send','event','통합검색(전체상품)','HOT클릭','셀러     명');HeaderComm.MiniMall.goHome.prdNo('1510196699',     HeaderComm.link.target.self); return false;" title=     \"건강저울나라\" data-is-send-log="true"&gt;건강저울나라   &lt;/a&gt;</pre>
자전거가좋다	
고객만족우수	

# 웹 자동화 심화: XPATH 심화

13

11번가 상품 검색 및 리스트 가져오기

## ■ 4단계: 모든 판매자 리스트 가져오기

1. data-log-actionid-label 속성값이 sellername인 태그 가져오기

1.xpath: //a[@data-log-actionid-label='sellername']

```
# data-log-actionid-label이라는 속성값이 sellername인 a태그를 전부 가져와 출력
seller_list = driver.find_elements_by_xpath('//a[@data-log-actionid-label=\
'sellername\'])
for seller in seller_list:
    print(seller.text)

except Exception as e:
    print(e)
finally:
    driver.quit()
```

# 웹 자동화 심화: XPATH 심화

14

11번가 상품 검색 및 리스트 가져오기

## 4단계: 모든 판매자 리스트 가져오기

1. data-log-actionid-label 속성값이 sellername인 태그 가져오기

1.xpath: //a[@data-log-actionid-label='sellername']

```
alghost:9th Alghost$ python3 9th-1.py
[건강저울나라] 모야2s 3초 접이식 휴대용 전기 자전거 어버이날 선물 : 1,490,000
[자전거가좋다] [인기자전거모음] 하이브리드/로드/싸이클 등 자전거 : 85,410
[자이크바이시클] [자이크] 일본 접이식 미니벨로 20인치 MTB 카모 : 275,000
고객응대우수 판매자입니다.
[자이크바이시클] [자이크] 일본 빈티지 클래식 자전거 : 241,000
고객응대우수 판매자입니다.
[자전거여행] [알톤] MTB자전거 데오레30단 팀버30D/AE550 : 450,000
```



# 웹 자동화 심화: XPATH 심화

15

11번가 상품 검색 및 리스트 가져오기

## ■ 복습을 위한 페이지징

1. 여러 페이지를 순회하면서 수집
2. 최하단에 있는 페이지징 링크를 통해 자동화!

- 주의

- ▶ 상품 검색 결과에서 모든 결과를 확인하면 자동 스크롤됨
- ▶ 이는 특정 상품(인기 상품?)을 보여주기 위해 스크롤을 변경
- ▶ 이로 인해 `click()`이 제대로 되지 않음

# 웹 자동화 심화: XPATH 심화

16

11번가 상품 검색 및 리스트 가져오기

## ■ 복습을 위한 페이징

1. 여러 페이지를 순회하면서 수집

2. 최하단에 있는 페이징 링크를 통해 자동화!

```
curr_page = 1
while True:
    # 스크롤을 내리며 자동으로 스크롤이 일어나 스크롤이 제대로 안됨
    # 이를 해결하기 위해 문서 맨 마지막으로 갔다가 올라옴
    driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
    time.sleep(0.5)
    driver.execute_script("window.scrollTo(0, 0);")

    # id가 thisClick을 포함하는 모든 태그를 가져옴
    results_list = driver.find_elements_by_xpath('//li[contains(@id,
\'thisClick\')]')
    ...

    curr_page += 1
    page_link = driver.find_element_by_link_text(str(curr_page))
    page_link.click()
    # 페이지가 동적으로 변경되기 때문에 기다려야 함
    time.sleep(2)
```

# 키보드, 마우스 제어하기: 기본

17

기본 라이브러리 소개

## ■ pyautogui 라이브러리

- 키보드와 마우스를 프로그래밍으로 제어할 수 있는 기능을 제공
- 원리
  - ▶ 모니터의 해상도에 맞춰 픽셀로 좌표를 사용
  - ▶ 왼쪽 최상단의 좌표가 0,0
  - ▶ x는 증가할수록 오른쪽으로 이동
  - ▶ y는 증가할수록 아래로 이동
- 이러한 원리로 인해 복잡한 자동화는 많은 노가다를 필요로함

# 키보드, 마우스 제어하기: 기본

18

기본 라이브러리 소개

## ■ pyautogui 라이브러리

- 레퍼런스: <https://pyautogui.readthedocs.io/en/latest/>
- 설치(Windows)
  - ▶ pip install pyautogui
- 설치(MacOS)
  - ▶ pip3 install pyobjc-core
  - ▶ pip3 install pyobjc
  - ▶ pip3 install pyautogui

# 키보드, 마우스 제어하기: 기본

기본 라이브러리 소개

## ■ 기본 라이브러리 소개

함수명	설명	
size	화면의 크기를 반환	res = pyautogui.size()
moveTo	원하는 위치로 마우스를 이동 (절대좌표)	pyautogui.moveTo(10, 50)
moveRel	원하는 위치로 마우스를 이동 (상대좌표)	pyautogui.moveRel(None, 100)
click	클릭, 아래 옵션을 통해 횟수와 버튼 지정 clicks, interval, x, y, button 옵션가능	pyautogui.click(clicks=2)
typewrite	키를 입력	pyautogui.typewrite('keyboard')
press	특수키를 입력할 때 사용	pyautogui.press('enter')
locateCenterOnScreen	그림과 일치하는 위치의 좌표 반환	x, y = pyautogui.locateOnScreen('a.png')

# 키보드, 마우스 제어하기: 기본

20

기본 라이브러리 소개

## ■ 기본 라이브러리 사용

- 기능 맛보기

```
import pyautogui

# 사이즈를 가져옴
screenWidth, screenHeight = pyautogui.size()
# 사이즈를 확인
print(screenWidth, screenHeight)

# 중앙으로 마우스 이동
pyautogui.moveTo(screenWidth/2, screenHeight/2)

# 오른쪽으로 마우스 이동
pyautogui.moveRel(100, None)

# 터미널 창으로 이동
pyautogui.moveTo(100, screenWidth/2)

# 키 입력
pyautogui.typewrite('keyboard test')

# 특수 키입력
pyautogui.press('enter')
pyautogui.press('enter')
```

# 키보드, 마우스 제어하기: 기본

21

기본 라이브러리 소개

## ■ 기본 라이브러리 사용

- 기능 맛보기
  - ▶ 화면 일부를 캡처하여 a.png로 저장
  - ▶ 파이썬 파일과 같은 폴더로 a.png 이동
  - ▶ 주의: 캡처한 이미지가 작을수록 성능 및 정확도가 떨어짐

```
# 이미지로 이동
point = pyautogui.locateCenterOnScreen('a.png')

if point:
    # MacOS의 Retina를 사용하는 경우 가상 픽셀로 해상도를 나타내기 때문에
    # /2가 필요
    pyautogui.click(x=point[0]/2, y=point[1]/2, clicks=2, interval=0.1)
    # 일반적인 경우에는 /2를 할 필요가 없음
    pyautogui.click(x=point[0], y=point[1], clicks=2, interval=0.1)
```

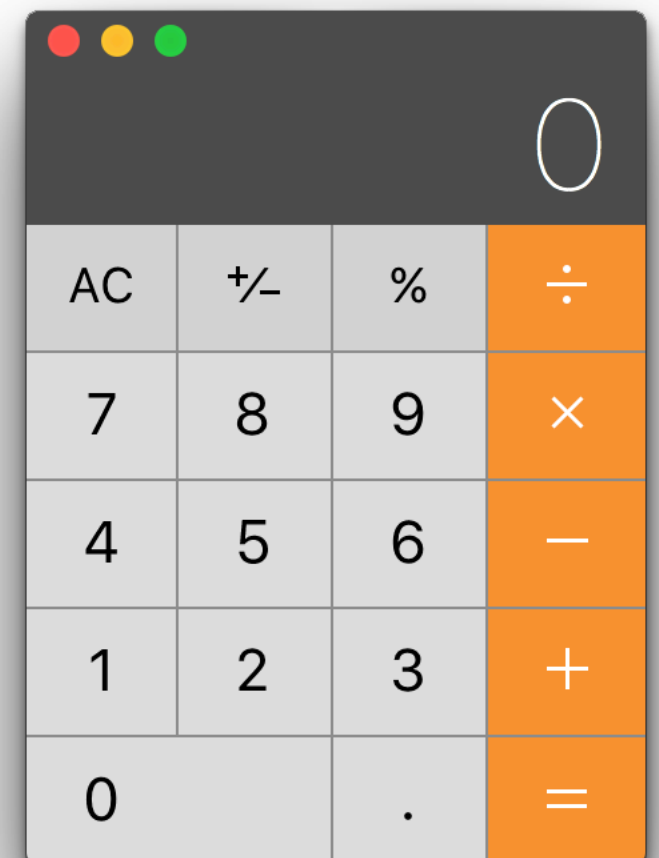
# 키보드, 마우스 제어하기: 기본

22

기본 라이브러리 소개

## ■ 기본 예제 따라하기

- 계산기 프로그램을 실행시켜 버튼 누르며 연산하기
  - ▶ 필요한 버튼을 이미지로 캡처하여 클릭하기
  - ▶  $5 \times 9 =$  실행하여 결과 확인해보기
- 주의
  - ▶ 이미지로 위치를 찾을 때 못찾을 경우 에러 발생
  - ▶ 따라서 try~except로 예외처리 필요





# 키보드, 마우스 제어하기: 기본

23

기본 라이브러리 소개

## ■ 기본 예제 따라하기

- 계산기 프로그램을 실행시켜 버튼 누르며 연산하기
  - ▶ 필요한 버튼을 이미지로 캡처하여 클릭하기
  - ▶  $5 \times 9 =$  실행하여 결과 확인해보기

```
import pyautogui

try:
    x, y = pyautogui.locateCenterOnScreen('5.png')
    pyautogui.click(x/2, y/2)
    x, y = pyautogui.locateCenterOnScreen('x.png')
    pyautogui.click(x/2, y/2)
    x, y = pyautogui.locateCenterOnScreen('9.png')
    pyautogui.click(x/2, y/2)
    x, y = pyautogui.locateCenterOnScreen('= .png')
    pyautogui.click(x/2, y/2)
except:
    print('No target')
```

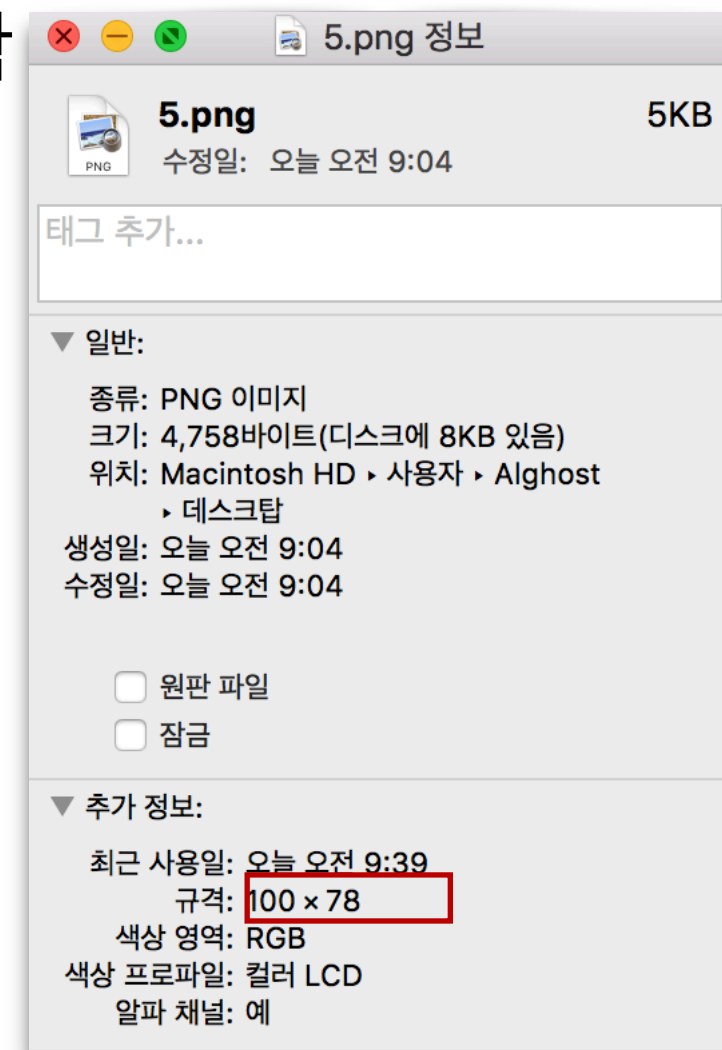
# 키보드, 마우스 제어하기: 기본

24

기본 라이브러리 소개

## ■ 기본 예제 따라하기 - 활용

- 기본 예제에서 확인한 문제
  - ▶ 이미지로 해당 버튼을 찾는데 많은 시간 소요 (무조건 발생)
  - ▶ 따라서 이미지로 버튼을 찾는 동작을 최소화 해야함
- 어떻게 해결?
  - ▶ 특정 버튼을 찾고 상대 경로로 다른 번호를 클릭
  - ▶ 캡처한 버튼의 크기를 가지고 픽셀을 추측
  - ▶ 규격: 100 x 78 (약 100 x 80으로 계산)
  - ▶ 주의: Retina를 사용하는 MacOS는 /2 필요



# 키보드, 마우스 제어하기: 기본

25

기본 라이브러리 소개

## ■ 기본 예제 따라하기 - 활용

- 기본 예제에 나 하인한 문제

- ▶ 이미지

- ▶ 따라

- 어떻게

- ▶ 특정

- ▶ 캡처

- ▶ 규칙

- ▶ 주의

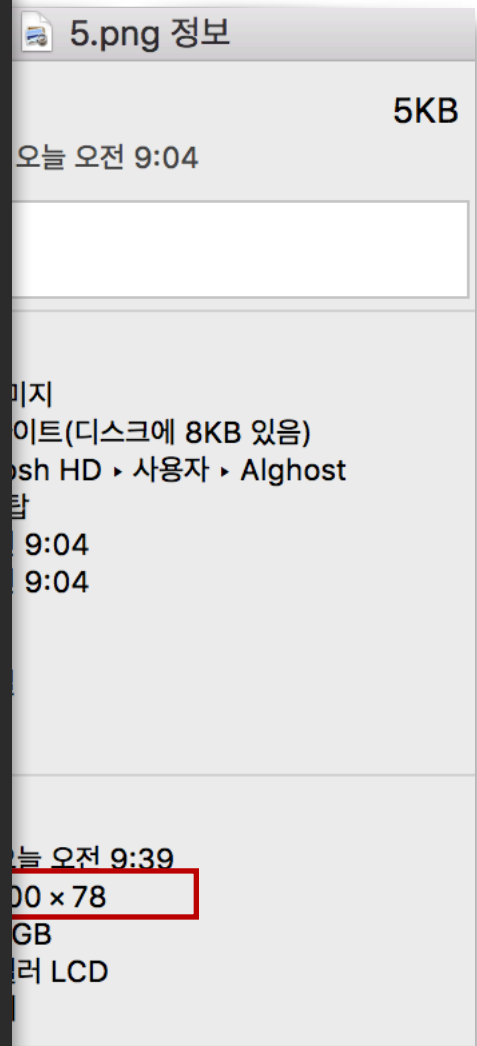
```
import pyautogui

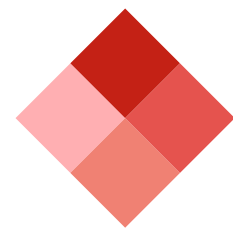
try:
    x, y = pyautogui.locateCenterOnScreen('5.png')
    # click 5
    pyautogui.click(x/2, y/2)

    # click x
    pyautogui.moveRel(100, -40)
    pyautogui.click()

    # click 9
    pyautogui.moveRel(-50, None)
    pyautogui.click()

    # click =
    pyautogui.moveRel(50, 120)
    pyautogui.click()
except:
    print('No target')
```





# Good Bye

See you next time