

```
In [ ]: Instagram => requests.session, cookies, headers
Meta, Graph API => Hack; Dev.;
Content(Node) - PK
HTTP(HTML); API(OAuth - 사용자인증); CSRF(서버인증) Token
CSRF(index.html); HTML(js(*), css, ...)
Selenium => BotX, AgentX, CralwerX, => 테스트 자동화
; JS Engine(Browser) Execute
UI/UX; 입력(키보드, 마우스); find_element(s) <= By.CSS_SELECTOR(*), XPATH
; locator => element(DOM;id); DOM; Clickable X, Enable X, Selected X
; Wait(Implicit, Explicit) => Test/Macro
; DOM(DOM(DOM)), Window_handle(창/탭)
```

```
In [ ]: TXT => raw text; Preprocessing
한국말: 음절, 어절, 단어, 형태소, 품사, 어근/접사, 어간/어미, 구, 절, 문단, 문서,...
        분절
Computational NLP: 4대 중의성(Ambiguity)
=> SPTokenizer(BPE)
=> X = 숫자 인코딩, X1 => 숫자 인코딩(feature)
=> 의미적 중의성(단어, 단어들..) = Embedding(Lexicon, LM, ..)
                                (Transformer)BERT, GPT(챗봇;QnA)
=> Embedding(NN, MLP, DNN, AR, GAN...)
```

```
In [20]: # 인터파크 예매(Selenium; Requests) - OCR
# CAPTCHA -> JS, 마우스움직임;
# URL-URL(RefererX)-URL3; Random Surfer;Damping Factor;
from selenium.webdriver import Chrome
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from requests.compat import *
import re
import json
```

```
In [2]: driver = Chrome()
```

```
In [5]: urls = {
        'home':'https://ticket.interpark.com'
    }

# 계정 불러오기
with open('interpark.txt', 'r') as f:
    user = json.load(f)
```

```
In [10]: # 로그인 체크

driver.get(urls['home'])
for c in driver.get_cookies():
    if not c['name'].startswith('_'):
        print(c['name'])

cto_bundle
ab.storage.sessionId.cd97b079-ff05-4967-873a-324050c2a198
ab.storage.deviceId.cd97b079-ff05-4967-873a-324050c2a198
pcid
OAX
```

```
In [44]: urls['login'] = 'https://accounts.interpark.com/authorize/ticket-pc?origin=https%3A%2F%2F'
driver.get(urls['login']) # DOM
```

```
In [45]: try:
    wait = WebDriverWait(driver, 3, .1)
    wait.until(
        EC.visibility_of_element_located(
            (By.CSS_SELECTOR, '#userId')
        )
    )

    wait.until(
        EC.visibility_of_element_located(
            (By.CSS_SELECTOR, '#userPwd')
        )
    )

    wait.until(
        EC.visibility_of_element_located(
            (By.CSS_SELECTOR, '#btn_login')
        )
    )

    uid = driver.find_element(By.CSS_SELECTOR, '#userId')
    upw = driver.find_element(By.CSS_SELECTOR, '#userPwd')
    ubtn = driver.find_element(By.CSS_SELECTOR, '#btn_login')

    uid.clear()
    uid.send_keys(user['id'])

    upw.clear()
    upw.send_keys(user['pw'])

    ubtn.click()
except:
    raise Exception('Error')
```

```
In [ ]: %%writefile interpark.txt
{
    "id": "본인아이디",
    "pw": "비밀번호"
}
```

```
In [46]: driver.get(urls['home'])
incookie = [c['name'] for c in driver.get_cookies()
            if not c['name'].startswith('_')]

```

```
In [47]: driver.get(urls['home'])
outcookie = [c['name'] for c in driver.get_cookies()
            if not c['name'].startswith('_')]

```

```
In [48]: list(zip(incookie, outcookie))
```

```
Out[48]: [('ab.storage.sessionId.cd97b079-ff05-4967-873a-324050c2a198',
            'ab.storage.userId.cd97b079-ff05-4967-873a-324050c2a198'),
            ('TSession%5FID', 'interparkID'),
            ('IMem%5FNO', 'TSession%5FID'),
            ('HMem%5FNO', 'ab.storage.sessionId.cd97b079-ff05-4967-873a-324050c2a198'),
            ('MemHash', 'GNBBookCnt'),
            ('interparktime', 'ab.storage.deviceId.cd97b079-ff05-4967-873a-324050c2a198'),
```

```
( 'memRealName', 'RegionCode'),
( 'iaf', 'imfs_pcid'),
( 'TSecLogin', 'idt_iss'),
( 'ab.storage.userId.cd97b079-ff05-4967-873a-324050c2a198', 'pcid'),
( 'interparkID', 'cto_bundle'),
( 'TMem%5FNO', 'SAVEID'),
( 'GNBBookCnt', 'sector'),
( 'interparkstamp', 'OAX')]
```

```
In [56]: def checkLogin(uid, upw, redirect=urls['home']):
    if not driver.get_cookie('TKLogin'):
        driver.get(urls['login'])

    try:
        wait = WebDriverWait(driver, 3, .1)
        wait.until(
            EC.visibility_of_element_located(
                (By.CSS_SELECTOR, '#userId')
            )
        )

        wait.until(
            EC.visibility_of_element_located(
                (By.CSS_SELECTOR, '#userPwd')
            )
        )

        wait.until(
            EC.visibility_of_element_located(
                (By.CSS_SELECTOR, '#btn_login')
            )
        )

        fid = driver.find_element(By.CSS_SELECTOR, '#userId')
        fpw = driver.find_element(By.CSS_SELECTOR, '#userPwd')
        fbtn = driver.find_element(By.CSS_SELECTOR, '#btn_login')

        fid.clear()
        fid.send_keys(uid)

        fpw.clear()
        fpw.send_keys(upw)

        fbtn.click()

        driver.get(redirect)
    except:
        raise Exception('Error')

checkLogin(user['id'], user['pw'])
```

```
In [57]: urls['concert'] = 'http://ticket.interpark.com/TPGoodsList.asp?Ca=Liv&Sort={}'
```

```
In [58]: driver.get(urls['concert'].format(5))
```

```
In [68]: # for _ in driver.find_elements(By.CSS_SELECTOR,
#                                     '.stit table > tbody > tr > td:first-child a'):
#     print(re.search(r'(\d{8})$', _.get_attribute('href')).group(1))
#     print(_.find_element(By.CSS_SELECTOR, 'img').get_attribute('alt'))
```

```
# print(re.search(r'한여름', _.find_element(By.CSS_SELECTOR, 'img')).get_attribute('alt'))

cList = [(re.search(r'(\d+)$', a.get_attribute('href')).group(1),
             a.find_element(By.CSS_SELECTOR, 'img').get_attribute('alt'))
          for a in driver.find_elements(By.CSS_SELECTOR,
                                         '.stit table > tbody > tr > td:first-child a')]
```

```
In [73]: len(list(filter(lambda t:re.search('대구', t[1]), cList)))
```

```
Out[73]: 28
```

```
In [86]: def checkConcert(name, sort=5):
            if urlparse(urls['concert']).path != \
               urlparse(driver.current_url).path:
                driver.get(urls['concert'].format(sort))

            cList = [(re.search(r'(\d+)$', a.get_attribute('href')).group(1),
                             a.find_element(By.CSS_SELECTOR, 'img').get_attribute('alt'))
                      for a in driver.find_elements(By.CSS_SELECTOR,
                                                     '.stit table > tbody > tr > td:first-child a')]

            if name:
                return list(filter(lambda t:re.search(name, t[1]), cList))
            else:
                return cList

checkConcert('NELL')[0]
```

```
Out[86]: ('23009076', 'NELL CLUB CONCERT 2023 'Burn'')
```

```
In [87]: urls['info'] = 'https://tickets.interpark.com/goods/{'
```

```
In [88]: urls['info'].format(checkConcert('NELL')[0][0])
```

```
Out[88]: 'https://tickets.interpark.com/goods/23009076'
```

```
In [89]: concert = checkConcert('NELL')[0]
driver.get(urls['info'].format(concert[0]))
```

```
In [90]: # 팝업닫기
for _ in driver.find_elements(By.CSS_SELECTOR,
                              '.popup.is-visible .popupCloseBtn'):
    _.click()
```

```
In [94]: # for _ in driver.find_elements(By.CSS_SELECTOR,
#     '''ul[data-view="days"] > li[class=""],
#     ul[data-view="days"] > li[class="picked"]'''):
#     print(_.text)
day = 3
driver.find_elements(By.CSS_SELECTOR,
                    '''ul[data-view="days"] > li[class=""],
                    ul[data-view="days"] > li[class="picked"]''')[day-1].click()
```

```
In [103]... time = 1
tList = driver.find_elements(By.CSS_SELECTOR,
                             'ul.timeTableList > li.timeTableItem > a')
if time > len(tList):
```

```
time = len(tList)
tList[time-1].click()
```

```
In [250]... def checkDateTime(cid, day=1, time=1):
            if urlparse(urls['info'].format(cid)).path != \
                urlparse(driver.current_url).path:
                driver.get(urls['info'].format(cid))

            for popup in driver.find_elements(By.CSS_SELECTOR,
                                                '.popup.is-visible .popupCloseBtn'):
                popup.click()

            wait = WebDriverWait(driver, 3, .1)
            wait.until(
                EC.visibility_of_all_elements_located(
                    (By.CSS_SELECTOR, '''
                        ul[data-view="days"] > li[class=""],
                        ul[data-view="days"] > li.picked
                    ''')
                )
            )

            dList = driver.find_elements(By.CSS_SELECTOR,
                '''ul[data-view="days"] > li[class=""],
                ul[data-view="days"] > li.picked''')
            dList[(day if day <= len(dList) else 1)-1].click()

            tList = driver.find_elements(By.CSS_SELECTOR,
                'ul.timeTableList > li.timeTableItem > a')
            tList[(time if time <= len(tList) else 1)-1].click()

            a = driver.find_element(By.CSS_SELECTOR, '.sideBtn.is-primary')
            if a.is_enabled():
                a.click()
            else:
                driver.execute_script('alert("예매불가");')

            # checkDateTime('23009076', 4)
```

```
In [251]... dList = driver.find_elements(By.CSS_SELECTOR,
            '''ul[data-view="days"] > li[class=""],
            ul[data-view="days"] > li.picked''')
```

```
In [249]... dList[4].click()
```

```
In [252]... checkDateTime(NELL[-1][0], 2, 1)
```

```
In [230]... driver = Chrome()

            checkLogin(user['id'], user['pw'], driver.current_url)

            NELL = checkConcert('NELL')

            checkDateTime(NELL[-1][0], 2, 1)
```

```
-----
IndexError                                Traceback (most recent call last)
Cell In [230], line 7
```

```

3 checkLogin(user['id'], user['pw'], driver.current_url)
5 NELL = checkConcert('NELL')
----> 7 checkDateTime(NELL[-1][0], 2, 1)

```

Cell In [225], line 13, in checkDateTime(cid, day, time)

```

8     popup.click()
10 dList = driver.find_elements(By.CSS_SELECTOR,
11 ''ul[data-view="days"] > li[class=""],
12 ul[data-view="days"] > li[class="picked"'])
----> 13 dList[(day if day <= len(dList) else 1)-1].click()
15 tList = driver.find_elements(By.CSS_SELECTOR,
16 'ul.timeTableList > li.timeTableItem > a')
17 tList[(time if time <= len(tList) else 1)-1].click()

```

**IndexError: list index out of range**

```
In [265]... driver.switch_to.window(driver.window_handles[-1])
driver.close()
```

```
In [118]... driver.switch_to.window(driver.window_handles[0])
```

```
In [121]... checkDateTime(NELL[0][0])
```

```
In [254]... len(driver.window_handles)
driver.switch_to.window(driver.window_handles[-1])
```

```
In [255]... iframe = driver.find_element(By.CSS_SELECTOR, '#ifrmSeat')
driver.switch_to.frame(iframe)
```

```
In [142]... captcha = driver.find_element(By.CSS_SELECTOR, '#divRecaptcha')
captcha.is_displayed()
```

Out[142]: False

```
In [138]... text = input()
```

DDCTNZ

```
In [139]... driver.find_element(By.CSS_SELECTOR, '.validationTxt').click()
```

```
In [140]... driver.find_element(By.CSS_SELECTOR, '#txtCaptcha').send_keys(text)
```

```
In [141]... driver.execute_script('fnCheck();')
```

```
In [256]... captcha = driver.find_element(By.CSS_SELECTOR, '#divRecaptcha')
```

```

while True:
    if not captcha.is_displayed():
        break

    text = input()
    driver.find_element(By.CSS_SELECTOR, '.validationTxt').click()
    driver.find_element(By.CSS_SELECTOR, '#txtCaptcha').send_keys(text)
    driver.execute_script('fnCheck();')

```

PLZPXX

```
In [160]... driver.refresh()
```

```
In [257]... seat = driver.find_element(By.CSS_SELECTOR, '#ifrmSeatDetail')
```

```
In [258]... seat = driver.find_element(By.CSS_SELECTOR, '#ifrmSeatDetail')
driver.switch_to.frame(seat)
```

```
In [259]... n = 2
# driver.switch_to.default_content()
# driver.switch_to.frame(
#     driver.find_element(By.CSS_SELECTOR, '#ifrmSeat'))
# driver.execute_script('fnRefresh();')
# seat = driver.find_element(By.CSS_SELECTOR, '#ifrmSeatDetail')
# driver.switch_to.frame(seat)

for s in driver.find_elements(By.CSS_SELECTOR,
                              'img[src$="1_90.gif"]')[n:]:
    s.click()
```

```
In [261]... driver.switch_to.parent_frame()
```

```
In [262]... driver.execute_script('fnSelect();')
```

```
In [263]... driver.switch_to.default_content()
```

```
In [197]... driver.switch_to.frame(iframe)
```

```
In [320]... driver.switch_to.window(driver.window_handles[0])
driver.close()
```

```
In [321]... # 다시(여기부터)
driver = Chrome()
```

```
In [322]... checkLogin(user['id'], user['pw'])
```

```
In [323]... NELL = checkConcert('NELL')
```

```
In [325]... checkDateTime(NELL[-1][0], 1, 1)
```

```
In [326]... if len(driver.window_handles) == 1:
    raise Exception('Error')

driver.switch_to.window(driver.window_handles[-1])

iframe = driver.find_element(By.CSS_SELECTOR, '#ifrmSeat')
driver.switch_to.frame(iframe)

captcha = driver.find_element(By.CSS_SELECTOR, '#divRecaptcha')

while True:
    if not captcha.is_displayed():
        break

    text = input()
    driver.find_element(By.CSS_SELECTOR, '.validationTxt').click()
    driver.find_element(By.CSS_SELECTOR, '#txtCaptcha').send_keys(text)
    driver.execute_script('fnCheck();')
```

```

seat = driver.find_element(By.CSS_SELECTOR, '#ifrmSeatDetail')
driver.switch_to.frame(seat)

n = 2
for s in driver.find_elements(By.CSS_SELECTOR,
                              'img[src$="S_f"]'):
    driver.execute_script(s.get_attribute('onclick'))

i = 0
for s in driver.find_elements(By.CSS_SELECTOR,
                              'img[src$="1_90.gif"]'):
    driver.execute_script(s.get_attribute('onclick'))
    if EC.alert_is_present()(driver):
        driver.switch_to.alert.accept()
    else:
        i += 1

    if i > (n-1):
        break

driver.switch_to.parent_frame()
driver.execute_script('fnSelect();')

```

SMXBMC

In [328]... `driver.switch_to.default_content()`

In [329]... `driver.switch_to.frame(driver.find_element(By.CSS_SELECTOR, '#ifrmBookStep'))`  
`driver.find_element(By.CSS_SELECTOR, '[name="SeatCount"] > option:last-child').click()`

In [330]... `driver.switch_to.default_content()`  
`driver.execute_script('fnNextStep("P");')`