

# Python 데이터 분석과 이미지 처리

나동빈

# 중급 Captcha Hacking 5 – 해킹 자동화

## 해킹 자동화: 수식 정제 (utils.py)

```
import re

def remove_first_0(string):
    temp = []
    for i in string:
        if i == '+' or i == '-' or i == '*':
            temp.append(i)
    split = re.split('\*|\+|-', string)
    i = 0
    temp_count = 0
    result = ""
    for a in split:
        a = a.lstrip('0')
        if a == '':
            a = '0'
        result += a
        if i < len(split) - 1:
            result += temp[temp_count]
            temp_count = temp_count + 1
        i = i + 1
    return result
```

# 중급 Captcha Hacking 5 – 해킹 자동화

## 해킹 자동화: 구현 완료 (run.py)

```
import requests
import shutil
import time

host = "http://localhost:10000"
url = '/start'

# target_images 라는 폴더 생성
with requests.Session() as s:
    answer = ''
    for i in range(0, 100):
        start_time = time.time()
        params = {'ans': answer}

        # 정답을 파라미터에 달아서 전송하여, 이미지 경로를 받아옵니다.
        response = s.post(host + url, params)
        print('Server Return: ' + response.text)
        if i == 0:
            returned = response.text
            image_url = host + returned
            url = '/check'
        else:
            returned = response.json()
            image_url = host + returned['url']
```

# 중급 Captcha Hacking 5 - 해킹 자동화

## 해킹 자동화: 구현 완료 (run.py)

```
print('Problem ' + str(i) + ': ' + image_url)

# 특정한 폴더에 이미지 파일을 다운로드 받습니다.
response = s.get(image_url, stream=True)
target_image = './target_images/' + str(i) + '.png'
with open(target_image, 'wb') as out_file:
    shutil.copyfileobj(response.raw, out_file)
del response

# 다운로드 받은 이미지 파일을 분석하여 답을 도출합니다.
answer_string = get_result(target_image)
print('String: ' + answer_string)
answer_string = utils.remove_first_0(answer_string)
answer = str(eval(answer_string))
print('Answer: ' + answer)
print("--- %s seconds ---" % (time.time() - start_time))
```