OneNote scheduler

**학과** : 전자융합공학과

**학년 :** 1학년

**학번** **:** 2022742021

**이름** **:** 김주호

* 과제 개요 : (어떤 내용을 할 것인지를 간략하게 요약합니다)

Google Calendar에 등록한 일정을 간편하게 볼 수 있도록 OneNote에 일정을 자동 작성하는 프로그램을 구현했습니다.

**OAuth인증** 후 **Google Calendar API**를 통해 가져온 유저의 일정 정보가 **달력 알고리즘**에 의해 **키보드& 마우스 자동화**로 OneNote에 일정을 작성합니다.

* 문제 정의와 과제의 필요성 (왜 이 과제를 하려고 하는지, 어떤 불편한 점을 개선하기 위한 것인지, 이것을 이용하면 어떻게 좋은지를 개괄적으로 설명합니다)

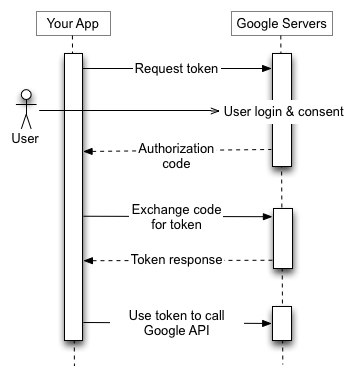
저는 매일 OneNote에 하루일과를 적고 바쁘게 살아갑니다. 하지만 너무 정신없이 살다 보면 가끔 달력에 적어 놓은 중요한 할 일을 종종 놓치곤 합니다. 그렇다고 달력에 작성된 모든 일정을 일일이 확인하여 OneNote에 적는 것이 번거롭다고 생각했습니다. 그래서 이 프로그램을 만들게 되었습니다.

예시를 하나 들어보겠습니다. 작년에 제 google calendar에 있는 일정은 100개가 넘었습니다. 제가 작성하다 실수한 점도 고려하면 작년 일정을 작성하는데 걸린 시간은 약 2시간이었습니다. 만약 그때 제가 프로그램을 사용했다면 얼만큼 걸렸을까요? 현재 제 프로그램으론 6분 이내 작성 가능합니다. 게다가 프로그램이 실행되는 동안 다른 일을 할 수 있었을 겁니다. 즉 매년 일정을 작성할 때 2시간을 소모하지 않아도 된다는 것입니다.

결론적으로 제 과제는 생산성을 높여주는 프로그램입니다. 자신이 세운 목표를 달성하면서 중요한 일정을 놓치는 것을 방지하는, 즉 체계적인 일정 설계를 하는 프로그램입니다.

* 개발 개요 (과제의 구체적인 내용을 서술합니다. 필요한 기술과 처리 절차 등을 가능하면 구체적으로 기술합니다.)

# 프로그램의 원리



(출처 : https://developers.google.com/identity/protocols/oauth2)

프로그램 폴더에 저장되어 있는 유저 정보, 즉 credentials.json 파일을 이용하여 OAuth인증을 합니다. OAuth 인증할 땐 google\_apis.py, Google.py, OAuth.py가 사용됩니다. 즉 본인 인증을 합니다. 본인 인증 후 발급된 유저 고유의 토큰을 이용하여 google calendar API를 통해 유저의 일정 정보를 불러옵니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

(출처 : OAuth.py 코드 일부 캡쳐)

그 이후 GetEvents 함수를 통해 모든 일정을 사전형으로 만듭니다. 사전의 키는 일정 날짜, 키의 값은 일정 이름으로 저장됩니다.

예) {'2021-08-01': [‘집 청소하기’], '2021-08-26': [‘요리하기’,’빨래하기’,’춤추기’]}

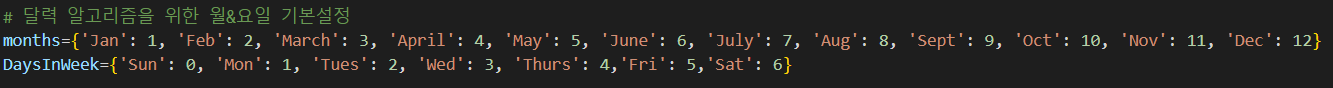
\*모든 일정의 사전형은 **OAuth.py 또는 main.py**에서 **print(GetEvents(“primary”))**를 실행하면 보실 수 있습니다.

최종적으로 main.py, 즉 메인코드에서 달력 알고리즘을 통해 날짜에 맞게 일정을 분류한 후, OneNote에 일정을 자동 작성하게 됩니다. 메인 코드에 대해 자세히 살펴보겠습니다.

Main.py에서 아래와 같이 모듈을 불러왔습니다

|  |  |
| --- | --- |
|  | **OAuth 모듈**  GetEvents함수를 통해 본인인증 및 일정을 불러옴  **Subprocess 모듈**  OneNote를 자동으로 실행  **Pyautogui 모듈**  OneNote 새 섹션 생성 및 정보 입력  **Pyperclip 모듈**  날짜/요일/일정 정보를 복사 붙여넣기하여 빠르게 입력  **Time 모듈**  원활한 프로그램 실행을 위한 프로그램 지연 |

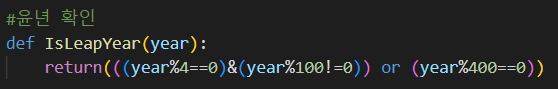
달력 알고리즘을 작성하기 위해 월과 각 요일에 해당하는 정수를 사전으로 만들었습니다.



데이터 형식을 맞추기 위해 아래와 같은 함수를 사용하였습니다.

|  |  |
| --- | --- |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | AddZero  1일을 1이 아닌 01로 저장할 수 있도록한다.  GetDates  결론적으로 AddZero를 통해 2022년 1월 1일을 2022-1-1이 아닌 2022-01-01로 저장이 되고, 변형된 데이터 형식이 반환된다.  Get\_Key  정수를 입력받아 해당하는 요일을 반환한다. |

년도를 입력받아 윤년 여부를 반환한다.



년도와 월을 입력 받아 그 달의 첫번째 요일을 반환한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

년도와 월을 입력받아 그 달의 총 날짜 수를 반환한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

OneNote에서 새로운 색션 생성후 그 달을 제목으로 입력한다. 예) Jan,Feb,March

텍스트이(가) 표시된 사진

자동 생성된 설명

해당 요일을 복사 붙여넣기 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

각 요일에 작성될 내용을 복사 붙여넣기 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

위 언급된 함수가 어울러져 날짜와 요일을 생성하는 함수 구성

텍스트, 모니터, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

모든 함수가 어울러져 해당 년도를 입력 받을 시 일정을 생성하는 함수가 개발되었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

맨 아래 줄에 있는 코드는 위 언급된 모든 것을 활용하여 아래와 같이 실행됩니다.

1. Event 변수에 사전형으로 된 모든 일정을 저장한다.
2. OneNote를 실행한다.
3. 원활한 프로그램 실행을 위해 2초간 시간 지연을 한다.
4. Event 변수에 저장된 일정을 2022년도 달력에 맞게 분류하여 궁극적으로 2022년도 일정을 OneNote에 생성한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* (계획이 변경되었다면) 중간 계획 변경 내용의 이유와 차이점 구체적 기술

Pyautogui와 pyperclip로 글자를 복사 붙여넣기하게 되어 keyboard 모듈을 삭제하게 되었습니다.



컴퓨터 스크린 사이즈와 onenote버전에 따라 프로그램 실행 환경이 바뀔수 있다는 점을 고려했습니다. 그래서 마우스 자동화를 이용한 연도 그룹 생성 코드를 삭제하게 되었습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

그에 따라 마우스 자동화를 제외한 일정생성 함수도 수정됐습니다

|  |  |
| --- | --- |
| **마우스 자동화를 통한 연도 생성** | **변경된 코드** |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | 텍스트이(가) 표시된 사진  자동 생성된 설명 |

* 개발 계획(개발 일정을 작성합니다)

|  |  |
| --- | --- |
| **달성** | **미달성** |
| OAuth 인증  Google Calendar에서 정보 불러오기  OneNote 자동으로 열기  OneNote에 자동 작성하기  모든 디바이스 호환 가능 | Google Calendar 정보 OneNote에 입력하기  모든 OneNote버전에 호환 가능하도록 코딩  프로그램 실행 시 달력에 추가적인 내용 추가 |

* 개발 내용

개발의 모든 과정이 어려웠습니다. OAuth와 달력 알고리즘은 교수님의 수업에서 다룬 개념이 아니라 독학을 해야 됬습니다. 유투브에서 Jie jenn님의 여러 영상을 통해 OAuth와 API의 기본적인 개념을 이해하고 google calendar api의 문서를 읽으며 유저의 정보를 불러오는 방법을 많이 배웠습니다.

* 결과

결과는 파일에 첨부된 동영상을 통해 확인할 수 있습니다.

* 결론

성공적으로 프로그램을 구현할 수 있습니다. 하지만 제가 아직 1학년이기에 부족한 점이 많습니다. 설명서에서 설명된 프로그램의 한계보다 부족한 점이 많지만 수많은 시행착오를 통해 혼자 많이 학습할 수 있었습니다.