**프로그래밍 원리와 실습 보고서**

**Python을 활용하여 Diary를 작성해 보자.**

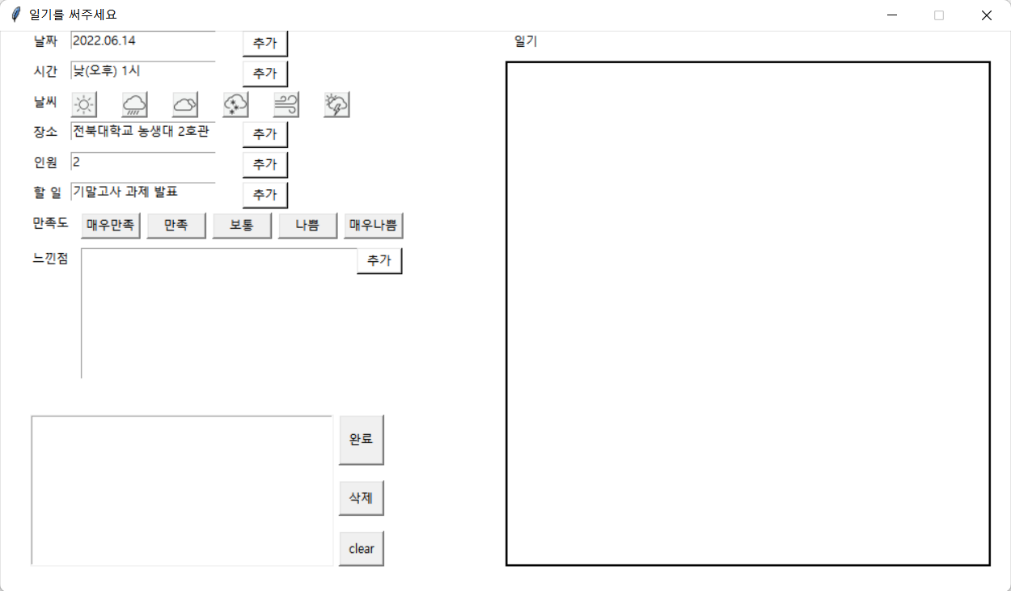
## 202110075 석승원 / 202110076 송주연

**프로젝트 배경 및 목표**

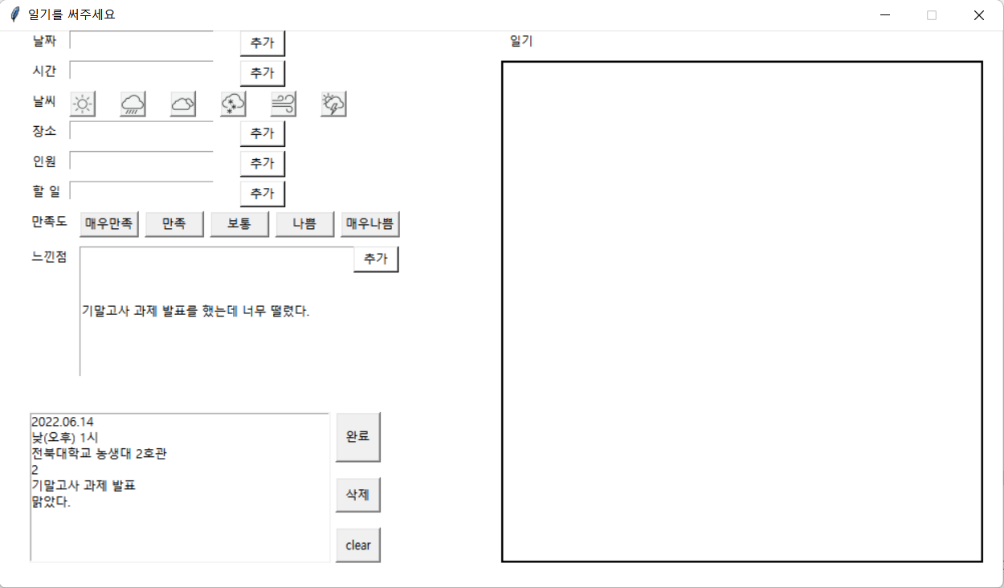
이 프로젝트는 바쁜 일상에 지쳐 일기를 쓸 겨를도 없는 현대인들이 하루의 일상을 놓치지 않고 기록할 수 있으면 좋겠다는 생각에 진행하게 되었습니다.  
하루동안 기억에 남는 하나의 일과에 대한 간단한 정보만 입력하면 자동으로 일기를 구현해 줍니다.  
일기는 느낀점이 중요한거 아니냐고요? --> 맞습니다! 그래서 느낀점 또한 추가하여 일기의 질을 높일 수 있도록 구현하였습니다.  
  
파이썬 모듈인 **tkinter**을 활용하여 자동 일기쓰기를 **GUI**로 구현하였습니다.

**구현 방법**

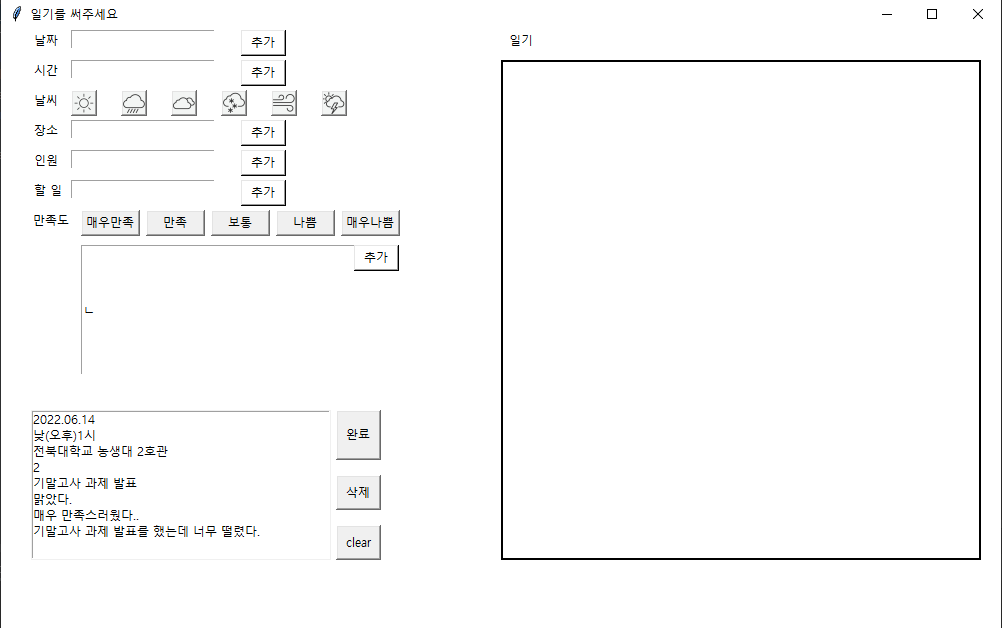
tkinter 모듈을 이용하여 gui로 구연했기 때문에 평소에 보던 검정 화면에서 벗어나 클릭할 수 있는 버튼과 직접 타이핑을 할 수 있는 텍스트 박스를 활용하여 보다 쉽게 활용할 수 있습니다.



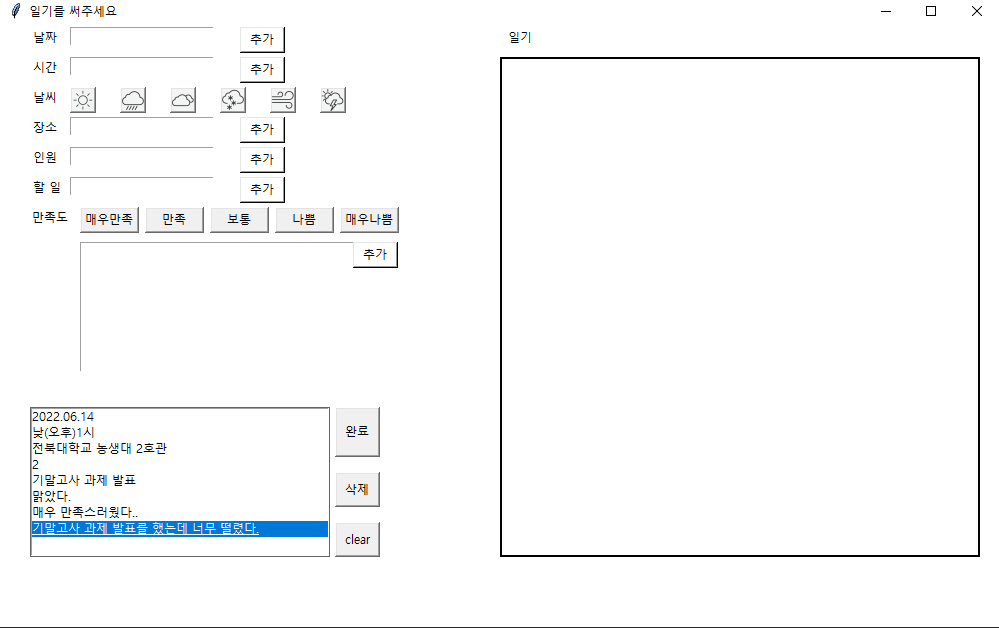
다음 사진과 같은 양식으로 텍스트 박스 안에 그날 한 일에 대한 간단한 정보를 입력합니다.



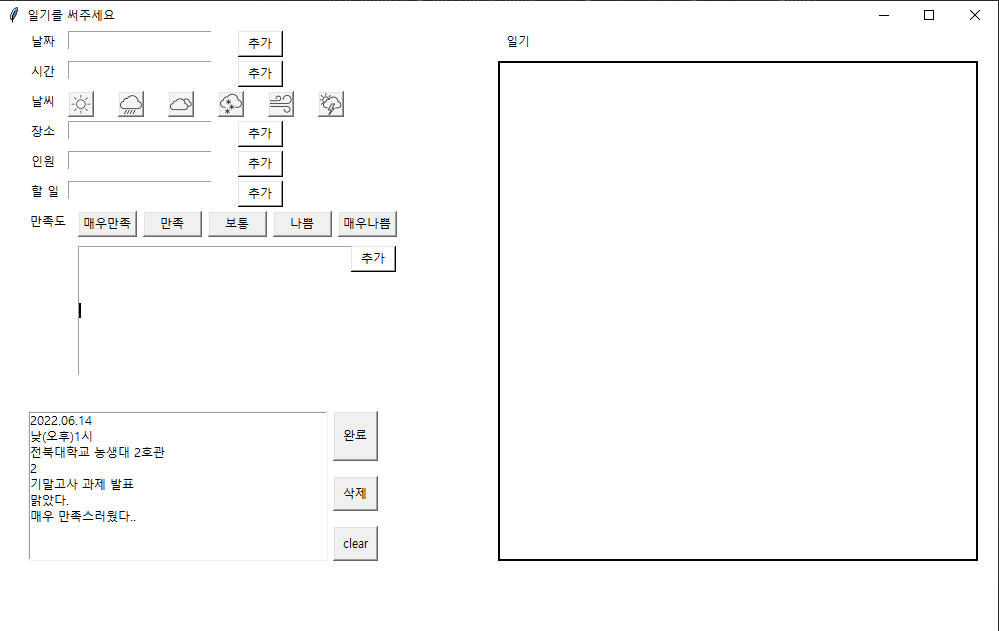
텍스트를 입력한 후 텍스트 박스 옆에 있는 '추가' 버튼을 클릭하고 확인 리스트 박스에 입력 값이 잘 들어갔는지 확인합니다.  
  
  
날씨의 경우 모든 선택지가 버튼으로 구현되어 있습니다. 날씨는 차례대로 '맑음', '비', '구름', '눈', '바람', '번개'입니다. 당시 날씨에 맞는 버튼을 클릭하면 리스트 박스 안에 해당 날씨가 자동으로 입력됩니다.  
만족도도 날씨와 마찬가지로 모든 선택지가 버튼으로 구현되어 있기 때문에 당일 한 일에 대한 만족도를 스스로 평가하여 입력하면 됩니다.



다음과 같이 모든 항목을 채웠으나 마음에 들지 않아 수정하고 싶을 경우에는

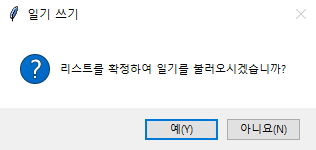


지우고 싶은 항목을 클릭한 뒤에 삭제 버튼을 누르면 다음과 같이 삭제됩니다.

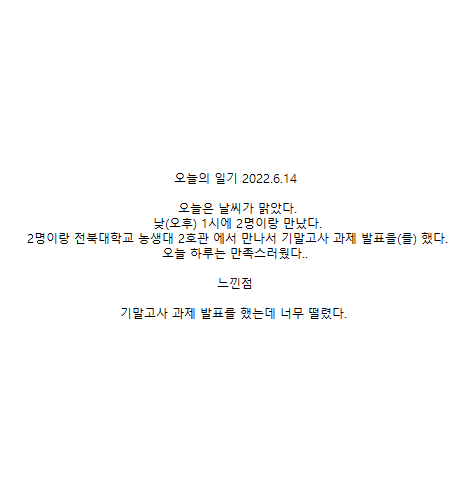


다음과 같이 항목이 삭제된 것을 확인할 수 있습니다.

이후 리스트 박스 안에 원하는 값이 모두 들어간 것을 확인했다면, 이후 '입력' 버튼을 클릭하면 일기가 자동으로 써서 출력됩니다.



다음 안내창이 뜨면 일기를 불러오려면 '예' 다시 수정하려면 '아니요'를 클릭합니다.

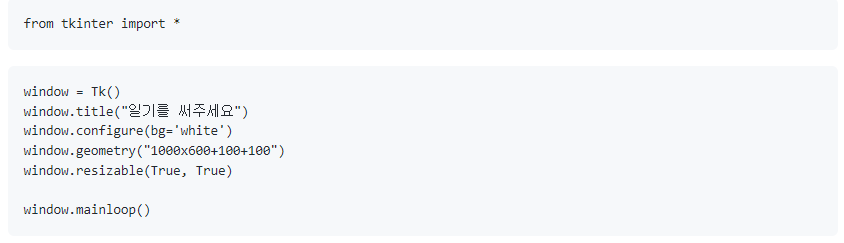


다음과 같이 일기가 자동으로 출력됩니다.

## 사용한 기술

**GUI** 에 대한 표준 **Python 인터페이스**인 **tkinter** 모듈을 사용하여 window 창을 활용하여 프로그램을 구현.

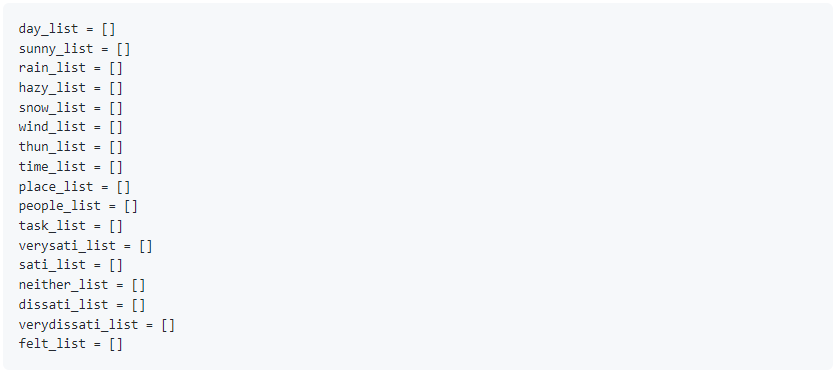
**tkinter 사용하기**



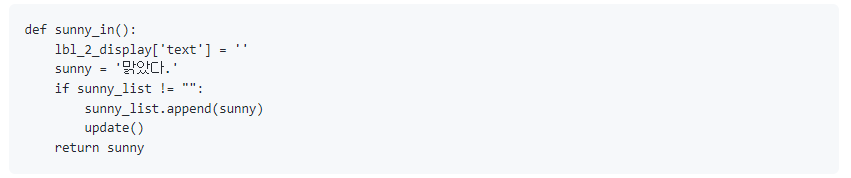
기본적인 window 틀을 만들어 줍니다. 순서대로 배경색, 사이즈, 창사이즈 조절 가능 여부를 의미합니다.  
  
마지막 **window.mainloop()** 는 창을 임의로 종료할때 까지 실행되도록 합니다.

**프로그램 설계**

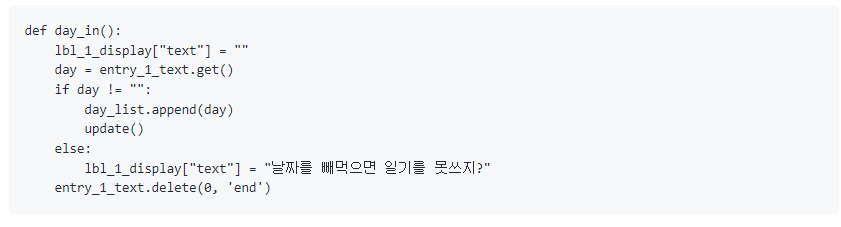
자동으로 일기쓰기 프로그램은 날짜, 시간, 날씨, 장소, 인원, 할 일, 만족도, 느낀점을 각각 따로 입력 받아 이미 짜여진 일기 스크립트에 적용되어 자동으로 일기가 작성되는 방법을 베이스로 합니다.  
  
이를 위해서는 각각 입력 받은 요소를 각각의 리스트에 집어 넣어 필요할때 꺼낼 수 있도록 설계했습니다.



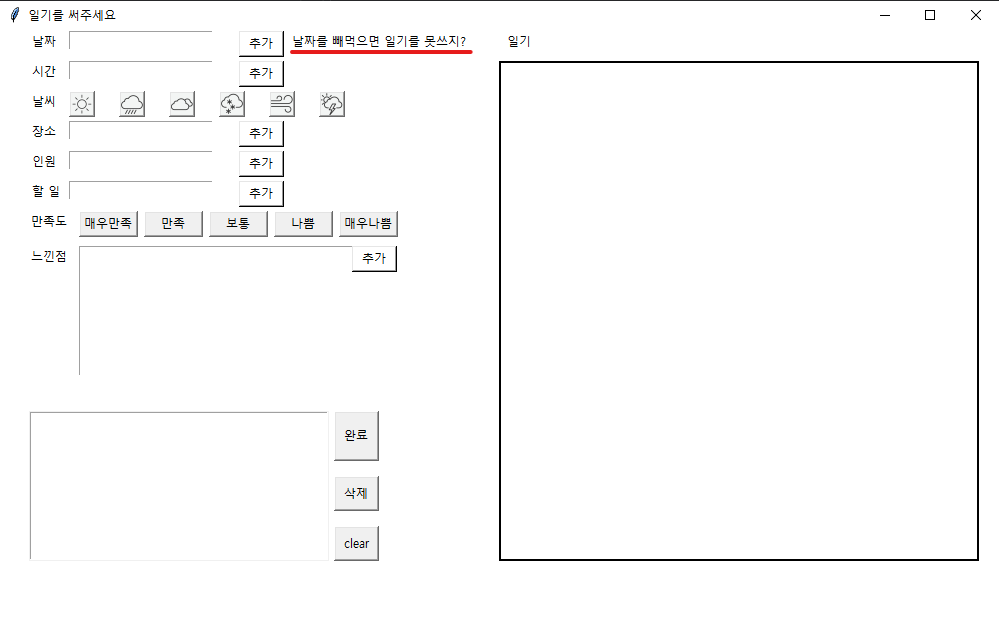
다음과 같이 각각을 리스트에 저장합니다. 날씨와 만족도의 경우 버튼 클릭으로 구현하고 싶었습니다.  
고민 끝에 각각의 요소를 모두 리스트로 만들어 리스트 안에 텍스트가 들어가도록 설계했습니다.

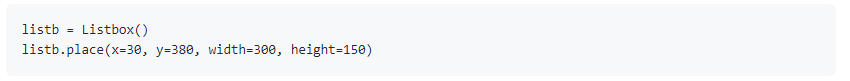


다음 날씨 예시 코드와 같이 '맑았다'를 sunny\_list 리스트 안에 고정으로 넣어 버튼을 클릭시 '맑았다'가 출력되도록 설계했습니다.

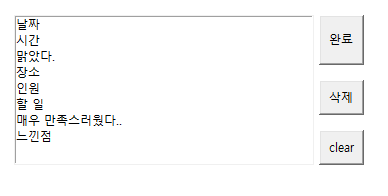


다음 날짜 예시 코드는 사용자가 날씨 예시 코드와는 다르게 정해진 텍스트가 아닌 사용자가 입력한 텍스트가 리스트 안에 들어가도록 설계했습니다. 만약에 사용자가 아무것도 입력을 하지 않고 '추가'를 클릭했을 시 "날짜를 빼먹으면 일기를 못쓰지?"와 같은 반응이 나오도록 설계했습니다.

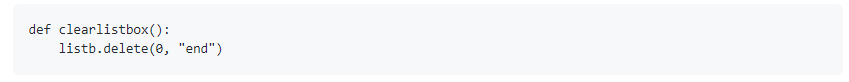




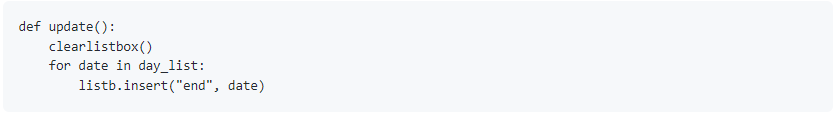
사용자가 어떠한 값을 입력했는지 확인하기 위해서 리스트 박스를 만들어 입력한 값을 확인하고 수정할 수 있도록 설계했습니다.



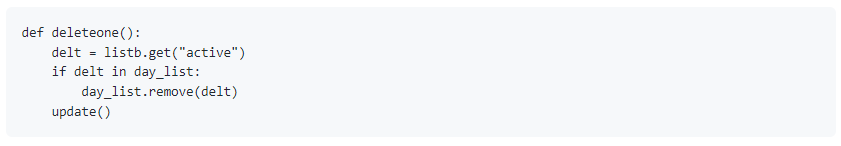
다음과 같이 입력한 값이 리스트 박스 안에 들어가고, 수정 작업을 할 수 있습니다.



다음 코드는 리스트 박스 안에 입력된 값을 지우는 코드입니다.



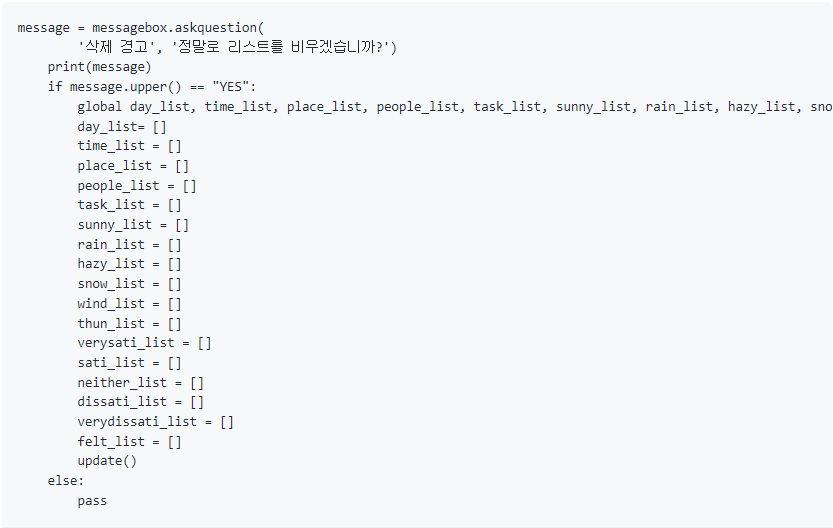
다음 날짜 예시 코드는 리스트 박스 안에 입력 받은 값을 넣어주는 코드입니다. 위에서 만든 clearlistbox()를 이용하여 선행 입력 값을 삭제하여 같은 값이 중복되어 들어가는 것을 방지합니다. for 반복문을 사용하여 날짜뿐만 아니라 다른 값들 또한 입력 받을 수 있도록 설계했습니다.



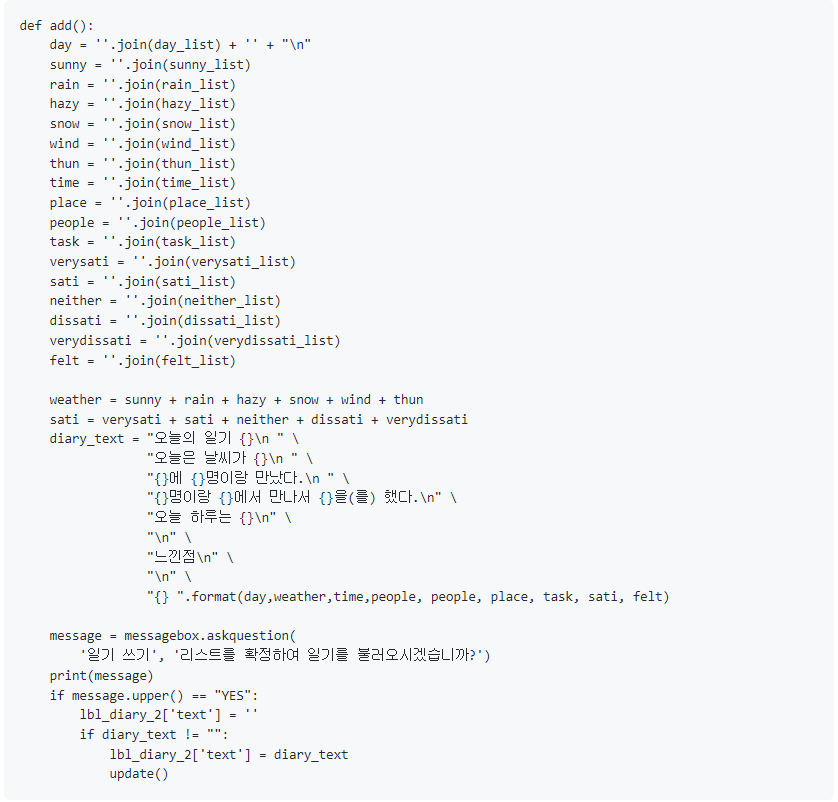
다음 날짜 예시 코드는 리스트 박스 안에 입력 받은 값을 제거하는 코드입니다. if 문을 사용하여 날짜 리스트 안에 들어 있는 입력 받은 값을 제거하고, update()을 다시 실행하여 리스트 박스 안에 들어간 값을 제거합니다.



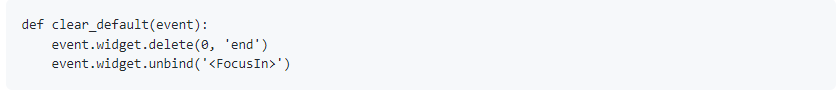
messagebox 모듈을 이용하여 새로운 윈도우 창이 실행되어 메세지를 전달할 수 있도록 설계했습니다.



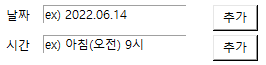
다음 코드는 리스트 박스 안에 입력 받은 값을 모두 지우는 코드입니다. messagebox 모듈을 이용하여 메세지 창을 띄우고, 사용자가 '예'를 클릭하면 모두 삭제되도록 설계했습니다. global을 통해 전역 변수인 리스트 들을 불러와 공백으로 만들고, update()를 실행하여 리스트 박스 내에 입력 받은 값들을 제거합니다.



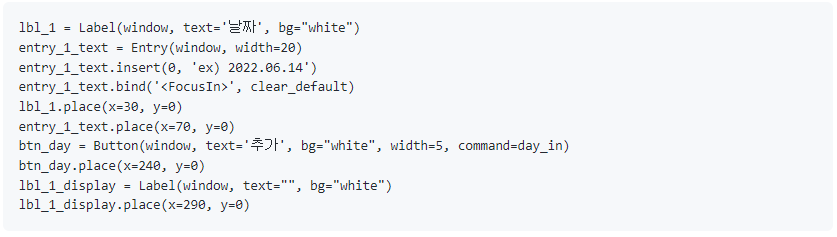
다음 코드는 사용자로 부토 입력 받아 리스트안에 들어간 정보를 기반으로 일기를 쓰는 코드입니다. ''.join()을 이용하여 리스트 내의 텍스트를 추출하여 미리 작성한 일기 스크립트 내에 지정한 위치에 대입합니다. 이 또한 마찬가지로 messagebox룰 이용하여 최종적으로 일기를 작성할 것인지 물어봅니다.



다음 코드는 입력하는 텍스트 박스 내에 클릭시 사라지는 디폴트 텍스트를 삽입합니다.



다음과 같이 디폴트 텍스트가 생성되고 클릭시 사라집니다.



마지막으로 다음 날짜 예제 코드는 tkinter의 GUI 위젯을 세팅하는 코드입니다.  
각각 Label(라벨), Entry(입력 칸), Button(버튼) 위젯의 크기를 조절하고 place()를 이용하여 위젯의 위치를 지정했습니다. command를 통해 위에서 본 day\_in 코드를 버튼을 누르면 실행되도록 설계했습니다.

## 참고 문헌

Python tkinter 강좌 - <https://076923.github.io/posts/Python-tkinter-1/>  
to do list - <https://pythonguides.com/python-tkinter-todo-list/>  
default box - <https://stackoverflow.com/questions/70406538/tkinter-entry-box-default-text>

**느낀점**

**202110075 석승원**

저는 이번 프로그래밍 원리와 실습을 수강하기 전에 나에게 썼던 편지에서 “코딩을 능숙하게 하지는 못하더라도 시간이 지나도 까먹지 않도록 원리를 정확히 이해하고 시간이 걸리더라도 문제를 해결할 수 있을 정도의 기초 프로그래밍 실력을 갖추는 것이 최종적인 목표” 라고 했습니다. 지금까지 교양 수업으로 프로그래밍 수업을 두 과목을 수강했으나, 저는 print(Hello, world!) 외에는 아무것도 못하는 수준이었기 때문입니다. 하지만 이번 프로그래밍 원리와 실습 수업을 통해서 저는 두가지의 변화가 있었습니다.

첫번째로 파이썬 작업 환경에 대한 겁이 없어졌습니다. 처음 파이참의 검정 화면을 보고 나중에는 프로그램을 직접 짜서 발표를 할 것이라는 말을 들었을 때 많이 무서웠습니다. 마치 아무것도 안보이는 터널을 라이트 하나에만 의존한채 걷는 느낌이었습니다. 하지만 강의 주차가 지나면 지날수록 이러한 생각이 많이 없어졌습니다. 하나의 문제를 해결하기 위해 친구들과 3일을 밤을 새고, 하염없이 구글에 찾아보고 매주 주말을 끙끙 앓으면서 보냈더니 어느덧 기말 프로젝트로 내가 생각한 간단한 프로그램 하나는 짤 정도의 수준이 되었습니다. 이 정도만 보더라도 저는 어느정도 생각했던 목표는 이루었다고 생각합니다.

두번째로 프로그램에 더욱 관심을 갖게 되었습니다. 불과 몇 달 전만 해도 print밖에 못썼던 제가 이제는 이를 활용하여 라즈베리파이도 건들이고, 더 나아가 딥러닝에도 발을 담그려고 하고 있으니까요. 또한 마지막 기말 프로젝트를 통해 다른 친구들의 발표를 보니 아직 제가 많이 부족하다는 것을 몸소 느끼게 되었습니다. 생각하지 못한 다양한 아이디어와 높은 완성도의 발표물들은 저 정도를 만들 수 있는 것이 저의 다음 목표로 삼아야겠다고 생각하게 되었습니다.

저는 같은 시간에 같은 것을 배워도 남들보다 항상 느렸습니다. 다른 사람들은 수업때 들은 내용을 바로바로 응용을 하는데 저는 그것이 힘들었습니다. 그래서 과제 하나를 하더라도 남들은 1시간에 끝낼 것을 저는 3시간이 걸리고 어쩔 때는 밤을 새기도 했습니다. 앞으로 프로그래밍 원리와 실습 이후로도 몇몇 프로그래밍 수업이 더 있을 텐데 얼마나 밤을 샐지 살짝 겁나기도 하고 설레기도 합니다.

**202110076 송주연**

이번 학기에 프로그래밍원리와실습 수업을 듣고나서 아주아주 조금 파이썬을 할 수 있게 된 것 같다. 이 강의를 들으면 종강쯤에는 혼자서 프로그램을 뚝딱 만들 수 있을 줄 알았는데 전혀 아니었다. 4개월 전 첫 과제로 작성한 나에게 쓴 편지를 보니까 4개월 후에 파이썬을 잘 다뤘으면 좋겠다고 했는데 그러기에는 시간이 턱없이 부족했던 것 같다. 사실 시간이 많아도 내가 잘할 수 있을지는 나도 모르겠다. 그래도 수업을 들으면서 1학년 때 배웠을 때 보다 많은 내용을 배우고 강의실에서 같이 실습해서 못 따라갈 때 옆 친구가 알려주고 바로 교수님께 질문할 수 있어서 비대면 보다 대면인 것이 훨씬 좋았다.

작년에는 if문과 while문을 배우면서부터 머리가 아프고 포기해서 가장 기초적인 덧셈, 뺄셈만 할 수 있는 실력이었다. 그런데 지금은 if문과 while문을 쓸 줄 알고 함수도 쓸 수 있을 정도까지, 딱 그 정도의 실력이 되었다. 더 많은 것을 배웠지만 응용할 수 없는 것이 화가 난다. 어려운 내용을 배운건지 아니면 내가 못하는 건지 모르지만 내가 생각한 것을 코드로 짤 수가 없어서 생각으로만 끝내야 한다는 것이 짜증났다. 더 배워서 혼자서 프로그램을 만들 수 있을 정도로 잘하고 싶은 마음이 든다.

기말 프로젝트를 하면서 처음으로 조별 과제를 해봤는데 덕분에 조원과 친해질 수 있게 된 것 같아서 좋았다. 사실 내가 한 게 많이 없어서 도움이 되었을지 모르지만 직접 코드를 짜서 프로그램을 만들었다는 것이 뿌듯했다. 그리고 과제 발표 날에 다른 조의 프로그램을 봤을 때는 좀 기분이 안 좋기도 하고 부럽고 대단해 보였다. 같은 수업을 들었는데 다른 사람들은 다 잘하는 것 같고 나만 못하는 것 같아서 마음이 복잡했다. 혼자서 한 분들도 둘이서 한 것만큼 너무 잘 만드셔서 너무 멋지다고 생각했다. 다음 학기에도 비슷한 강의가 있는 것 같은데 그 수업을 내가 따라갈 수 있을지 모르겠다.