

예시 링크

이 문서는 프로그래밍을 공부하고 개발하는 과정을 쓰는 일지다. 여기에는 날마다 나에게 주어진 과제를 풀고, 프로그래밍 관련 리서치를 하고, 코딩을 한 것을 기록한다. 왜 이런 생각을 하게 되었는지, 어떻게 하게 되었는지를 기록해서 사고방식을 개발한다.

1. 까먹을 것 같은 내용은 공부하면서 미리 기록해 둔다.
 2. 프로그래밍 공부한 내용을 기억나는대로 쓴다.
 3. 프로그래밍 실습한 내용을 스크린샷으로 캡처해서 기록한다.
 4. 웹 검색, 리서치한 내용을 쓴다.
 5. 왜 이런 생각을 하게 되었는지, 어떻게 하게 되었는지를 기록한다.
-

[양식] 아래 양식을 복사해서 사용

코딩 문제 풀이

내용 간단 요약

예제 실행 결과

문제 풀이

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

회고

잘한 것은 무엇인가?

잘못한 것은 무엇인가?

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

코딩 문제 풀이

내용 간단 요약

예제 실행 결과

문제 풀이

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

회고

잘한 것은 무엇인가?

잘못한 것은 무엇인가?

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

내용 간단 요약

예제 실행 결과

```
from random import * # 랜덤 모듈 import
```

```
words = ["apple", "banana", "orange"] # 리스트에 영어 단어 후보를 나열
```

```
word = choice(words) # 랜덤으로 단어 중 1개를 선택
```

```
print("answer : " + word) # 참고용으로 정답 출력 (실제 게임에서는 지우기)
```

```
letters = "" # 플레이어가 지금까지 입력한 알파벳들 저장
```

```
# 정답을 맞힐 때까지 무한 반복
```

```

while True:
    succeed = True # 성공 여부 확인 변수
    print()
    for w in word: # 제시 단어를 알파벳별로 한 글자씩 비교
        if w in letters: # 현재 알파벳이 플레이어가 입력한 값들 중에 있으면
            print(w, end=" ") # 그 알파벳을 표시
        else: # 입력한 값들 중에 없으면
            print("_", end=" ") # 밑줄을 표시
    succeed = False # 밑줄이 있다는 것은 아직 다 풀지 못했음을 의미!
    print()
    if succeed: # 만약 성공했다면 게임 종료
        print("Success")
        break
    letter = input("Input letter > ") # 플레이어로부터 한 글자씩 입력
    if letter not in letters: # 입력값 중에 포함되어 있지 않다면
        letters += letter # 새로 입력받은 글자를 입력값에 추가
    if letter in word: # 입력한 글자가 제시 단어에 포함되었다면
        print("Correct")
    else: # 포함되어 있지 않다면
        print("Wrong")

```

문제 풀이

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

회고

잘한 것은 무엇인가?

잘못한 것은 무엇인가?

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

내용 간단 요약

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

회고

잘한 것은 무엇인가?

잘못한 것은 무엇인가?

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

내용 간단 요약 II

예제 실행 결과

문제 풀이

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

회고

잘한 것은 무엇인가?

잘못한 것은 무엇인가?

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

내용 간단 요약

문제 풀이

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

회고

잘한 것은 무엇인가?

잘못한 것은 무엇인가?

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

내용 간단 요약

문제 풀이

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

회고

잘한 것은 무엇인가?

잘못한 것은 무엇인가?

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

내용 간단 요약

문제 풀이

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

회고

잘한 것은 무엇인가?

잘못한 것은 무엇인가?

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

내용 간단 요약

문제 풀이

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

잘한 것은 무엇인가?

잘못한 것은 무엇인가?

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

책 12장

내용 간단 요약

문제 풀이

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

[양식] 아래 양식을 복사해서 사용

코딩 문제 풀이

내용 간단 요약

예제 실행 결과

문제 풀이

```
year = int(input())
month = int(input())
day = int(input())
```

```
a = year - month + day
if a % 10 == 0:
    print('대박')
else:
    print('그럭저럭')
```

1902

2

10


```
if r == 6: print(1)
elif r == 5:
    if bonus == 0: print(3)
    else: print(2)
elif r == 4: print(4)
elif r == 3: print(5)
else: print(0)
```

0

잘 이해되지 않는 것들

검색해보니까 입력값을 split 함수를 써서 공백으로 분리시켰는데 내가 짠 코드는 그걸 안하고 그냥 값을 따로 입력했는데 split을 꼭 써야하는건지 모르겠다.

3번째 문제 if 다음에 왜 elif를 써야하는지 모르겠다. if 계속쓰면 안되나??

4번째 문제는 if 랑 elif를 번갈아가며 쓰니까 잘 모르겠다.

책 외부에서 알게 된 내용

*map, split을 사용하는 입력 방식

실수나 정수인 여러 값들이 입력값일 때,

split 메서드를 사용해 공백이나 어떤 문자를 기준으로 입력값을 여러 값들로 분리하고,

map 내장 함수로 각각의 값들을 int형이나 float형으로

정수, 실수 자료형으로 변환하는 하는 기법이 많이 쓰인다.

map내장 함수:

여러 개의 데이터를 한 번에 다른 형태로 변환하기 위해서 사용됩니다. 따라서, 여러 개의 데이터를 담고 있는 list나 tuple을 대상으로 주로 사용하는 함수다.

map(변환 함수, 순회 가능한 데이터)

함수는 두번째 인자로 넘어온 데이터가 담고 있는 모든 데이터에 변환 함수를 적용하여 다른 형태의 데이터를 반환한다.

예) year, month, day = map(int, input().split(" "))

회고

잘한 것은 무엇인가?

간단한 코드를 작성해봤다.

잘못한 것은 무엇인가?

어떻게 시작해야하는지 모르겠어서 답을 먼저 봤다.

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

답에서 안배운 것 같은 내용들로 코드를 짜서 잘 모르겠다. map 내장함수?
잘 이해되지 않는 것들에 있는 내용들 잘 모르겠다.

향후 어떻게 다르게 할 것 인가?

파이썬 공부를 열심히 한다.

코딩 문제 풀이

내용 간단 요약

<https://codeup.kr/problem.php?id=1675>시저의암호
<https://codeup.kr/problem.php?id=1380>두 주사위의 합
<https://codeup.kr/problem.php?id=14203>등 찾기
<https://codeup.kr/problem.php?id=1511>테두리의 합
<https://codeup.kr/problem.php?id=1620>자릿수의 합

예제 실행 결과

문제 풀이

시저의 암호

```
code_list = ['x','y','z','a','b','c','d','e','f','g','h','i','j','\n',\n             'k','l','m','n','o','p','q','r','s','t','u','v','w']
alpha_list = ['a','b','c','d','e','f','g','h','i','j','\n',\n              'k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']
text = input()
for i in range(len(text)):
    if text[i] >= 'a' and text[i] <= 'z':
        print(code_list[alpha_list.index(text[i])],end="")
    else:
        print(text[i], end = "")
qhyhu wuxvw euxwxv
```

never trust brutus

1. 원문 알파벳을 코드리스트에 저장한다.
2. 암호문 알파벳을 알파리스트에 저장한다.
3. text길이 만큼 for문을 실행한다.
4. 만약 text[i]가 'a'보다 크거나 같거나 'z'보다 작거나 같으면 3만큼 이동한 알파벳으로 치환한 값을 출력한다.
5. 그렇지 않으면 검사했던 그 글자를 그대로 출력한다.

두 주사위의 합

```
from sys import stdin
```

```
k = int(stdin.readline())
```

```
for dice1 in range(1,7):
```

```
    for dice2 in range(1,7):
```

```
        if dice1+dice2 == k:
```

```
            print(dice1,dice2)
```

1. readline을 사용하기 위해 import한다.
2. 두 주사위를 굴러 나오는 합 k를 설정한다.(int로 변환)
3. 첫번째 주사위의 눈이 작은 순으로 출력하기 위해 첫번째 주사위의 숫자를 1,6까지 range문을 이용해 반복한다.
4. 두번째 주사위의 숫자도 1,6까지 반복한다.
5. 첫번째 주사위와 두번째 주사위 숫자의 합이 k와 같다면 첫번째 주사위,두번째 주사위의 숫자를 출력한다.

3등 찾기

```
import sys
```

```
def sort_score(n):
```

```
    student = []
```

```
    for i in range(n):
```

```
        name,score = input.split()
```

```
        score = int(score)
```

```
        student.append([name,score])
```

```
    result = sorted(student, key = lambda x : x[1], reverse = True)
```

```
    return result[2][0]
```

```
n= int(sys.stdin.readline())
```

```
print(sort_score(n))
```

1.def을 사용해

2. student를 리스트로 설정한다.

3. n번 반복한다.

4.이름과 점수를 입력값을 split함수로 나눠줘서 나타낸다.

5. 점수를 정수형 int로 설정한다.

6.student 리스트에 이름과 점수를 append함수를 써서 나타낸다.

7.sorted로 정렬하는데 내림차순으로 정렬한다.

8.세 번째로 점수가 높은 학생의 이름을 출력한다.

테두리의 합

```
n = int(input())
```

```
m = []
```

```
b = 1
```

```
for i in range(n):
```

```
    a=[]
```

```
    for j in range(n):
```

```
        a.append(b)
```

```
        b+=1
```

```
    m.append(a)
```

```
b = 0
```

```
for i in range(n):
```

```
    b += m[0][1]
```

```
for i in range(n):
```

```
    b+=m[n-1][i]
```

```
for i in range(1,n-1):
```

```
    b+=m[i][0]
```

1.n값을 입력받는다.

2.빈 리스트를 m이라는 변수로 설정한다.

3.for문을 n번 반복한다.

4. 변수가 a인 리스트를 하나더 생성한다.

5.a리스트에 b를 추가한다.

6.b리스트를 for문이 실행될때마다 확장한다.

자릿수의 합

```
n = int(input())
```

```
div = 1
```

```
s = 0
```

```
while(s==0 or s>=10):
```

```
    s=0
```

```
    while(n>0):
```

```
        s+=n%10
```

```
        n = n/10
```

```
    n=s
```

```
print(s)
```

```
1234567
```

```
1
```

1.수를 입력받는다.

2.몫이 0이 아닌동안 while문이 반복된다.

3.몫을 구한다.

4.각 자리수(나머지)를 구한다.

5.자리수의 합을 구한다.

6.자리수의 합을 출력한다.

7.끝난다.

잘 이해되지 않는 것들

시저의 암호 문제에서 `print(' ', end='')`이런식으로 쓴게 모르겠다.

두 주사위의 합에서 `from sys import stdin`이나 3등찾기에서 `import sys`를 썼는데 뭔지 모르겠다.

자릿수의 합에서 `div=1`부분이 꼭 있어야하는 건지 모르겠다. 없어도 출력값이 똑같았다.

테두리의 합 문제는 전체적으로 잘 모르겠다.

`cmd`창에 `hello world!` 무지개색으로 출력해오기 문제는 못 풀었다..

책 외부에서 알게 된 내용

`from sys import stdin, sys.stdin.readline()`

파이썬에서 입력값을 받을때 보통 `input()`을 이용한다. 하지만 알고리즘에서 `input()`을 이용할 때 종종 시간초과가 발생하기 때문에 `sys`모듈의 `sys.stdin`을 사용한다. `sys(system)/`

`sorted(정렬할 데이터, key파라미터, reverse파라미터)`

`리스트.sort()`는 본체의 리스트를 정렬해서 변환하는 것이고, `sorted(리스트)`는 본체 리스트는 내버려두고, 정렬한 새로운 리스트를 반환하는 것이다.

`interpreter`: 소스코드를 처음부터 한 라인씩 차례대로 해석하며 실행하는 프로그램을 뜻한다.

`cmd`는 `command`의 약자로 윈도우에서 도스 환경으로 연결한다.

실행은 윈도우키 + R을 누른다.

알람 시계 프로그램 짜는 법

1. UI(user interface)설계 => 보이는 거

2. 논리 구조 설계 => 하던대로

3. 구현 => 코드 짜기

UI설계 어떻게?

언제 알람 울린건지 시간 입력

시간 입력 어떻게?

분만 입력 받아서 입력받은 분 후에 알람 소리가 울리는 프로그램을 작성

시점: 프로그램 실행했을 때 알람 프로그램입니다.

몇 분 후에 알람을 울려 하는지 알려주세요(분 단위): [사용자 입력]

시점: 분 입력 받은 후

?분 후 알람이 울립니다.

시점: 알람 소리가 울린 후

알람이 울렸습니다.

논리 구조 설계

1. 프로그램 소개 글을 출력한다.

2. 입력 안내 글을 출력하고, 몇 분 후에 알람을 울린 것인지 입력받는다.

3. 몇 분 후에 알람이 울리는지 안내 메시지를 출력한다.
4. 입력 받은 기간 동안 대기한다.
5. 알람 소리를 스피커로 출력한다.
6. 알람이 울렸다는 안내 메시지를 출력한다.

회고

잘한 것은 무엇인가?

간단한 코드를 작성해봤다.

잘못한 것은 무엇인가?

어떻게 시작해야하는지 모르겠어서 답을 먼저 봤다.

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

파이썬 공부를 열심히 한다.

내용 간단 요약

행맨 게임

예제 실행 결과

```
from random import * # 랜덤 모듈 import

words = ["apple", "banana", "orange"] # 리스트에 영어 단어 후보를 나열

word = choice(words) # 랜덤으로 단어 중 1개를 선택

print("answer : " + word) # 참고용으로 정답 출력 (실제 게임에서는
지우기)

letters = "" # 플레이어가 지금까지 입력한 알파벳들 저장


# 정답을 맞힐 때까지 무한 반복

while True:

    succeed = True # 성공 여부 확인 변수

    print()

    for w in word: # 제시 단어를 알파벳별로 한 글자씩 비교

        if w in letters: # 현재 알파벳이 플레이어가 입력한 값들 중에 있으면

            print(w, end=" ") # 그 알파벳을 표시

        else: # 입력한 값들 중에 없으면

            print("_", end=" ") # 밑줄을 표시

        succeed = False # 밑줄이 있다는 것은 아직 다 풀지 못했음을
의미 !

    print()


    if succeed: # 만약 성공했다면 게임 종료
```

```
print("Success")
```

```
break
```

```
letter = input("Input letter > ") # 플레이어로부터 한 글자씩 입력
```

```
if letter not in letters: # 입력값 중에 포함되어 있지 않다면
```

```
    letters += letter # 새로 입력받은 글자를 입력값에 추가
```

```
if letter in word: # 입력한 글자가 제시 단어에 포함되었다면
```

```
    print("Correct")
```

```
else: # 포함되어있지 않다면
```

```
    print("Wrong")
```

문제 풀이

주제 간단 소개: 행맨 게임: 영어 단어에 알파벳을 맞추는 게임이다.

UI:

답을 어떤 영어단어로 할지 설정

영어 단어 어떻게 설정?

변수에 영어 단어 리스트로 저장

알파벳을 하나씩 맞추기를 시도

시점: 프로그램 실행했을 때 영어 단어에 해당 알파벳이 있는지 판별하는 프로그램이다.

해당 알파벳이 미리 설정한 영어단어에 존재하면 정답이다.

하지 않으면 오답이다.

설정한 영어 단어의 알파벳을 모두 맞히면 게임이 끝난다.

논리구조:

영어 단어 답을 출력한다.

_밑줄을 출력한다.

답이라고 생각하는 알파벳을 입력하라는 문구를 출력한다.

입력한 알파벳이 영어 단어에 있는지 판단한다.

영어 단어에 해당 알파벳이 존재하면 correct를 존재하지않는다면 wrong을 출력한다.

단어의 알파벳을 모두 맞추면 success를 출력한다.

코드 간단 분석:

1. choice함수를 사용하기 위해 랜덤 모듈을 import해준다.
 2. 영어 단어 후보를 나열한 리스트를 words라는 변수에 설정한다.
 3. choice함수를 이용해서 랜덤으로 words의 단어 중 한개를 word라는 변수에 저장한다.
 4. 'answer': word라는 문구를 출력한다.
 5. letters에 지금까지 입력한 알파벳들을 저장한다.
 6. while문이 True이면 정답 맞힐때까지 while문을 반복한다.
 7. w라는 변수가 word안에 있다.
 8. w라는 현재 알파벳이 플레이어가 입력한 알파벳 letters값 중에 있다면 해당 알파벳을 표시하여 출력한다.
 9. 없다면 밑줄을 표시하여
- 출력한다.
10. 알파벳을 모두 맞췄다면 success를 출력하고 게임을 종료한다.
 11. 성공하지 못했다면 새로 입력받은 글자를 letters라는 입력값에 추가한다.
 12. 입력한 글자가 답의 단어에 포함되어있다면 correct, 포함되어있지 않다면 wrong을 출력한다.

잘 이해되지 않는 것들

print() 출력 안에 아무것도 없는건 뭘 뜻하는 거,,? -> 줄바꿈용

succeed = True

succeed = False

letters = " 문자열에 저장? letters = []면 빈 리스트에 letters라는 변수로 저장하는거랑 비슷한건가? -> o o

책 외부에서 알게 된 내용

from random import 랜덤함수 사용하기 위해 설정한다.

choice함수: 이미 정해놓은 목록이 있고 그 중에서 하나를 뽑을 때 사용한다.

"_", end = " " 는 플레이어가 입력한 값중에 일치한 값이 없으면 _를 출력

succeed = True 를 해주지만 밑에 if 문에서 밑줄(_)이 남아있으면(플레이어가 아직 문자를 다 맞추지 못했다면) succeed = False로 바꿔주는 코드이다.

회고

잘한 것은 무엇인가?

간단한 게임원리의 코드를 작성해봤다.

잘못한 것은 무엇인가?

논리 설계를 써보고 단어 설정까지는 했는데 어떤식으로 해야할지 모르겠어서 답을 봤다.

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

구글링 더 열심히 한다.

내용 간단 요약

개인별 만들고 싶었던 프로그램 만들어보기 중간 발표

플젝 소개 + 실행 가능한 코드 + UI + 논리구조

플젝 소개: 통학버스에 자리가 남았는지 확인하기

UI: 남은 버스 자리수, 남은 자리가 없습니다!!

논리구조 :

1. 총 버스 자리수를 설정한다.
2. 남은 자리수가 1씩 줄어들도록 한다.
3. 남은 자리수가 0이 되면 남은 자리가 없음을 출력한다.

```
total = 20 #통학 버스 안의 총 자리수
```

```
present_seat = 0
```

```
for present_seat in range(20,0,-1):
```

```
    print('남은 자리 수:', present_seat)
```

```
    if present_seat==0:
```

```
        print('남은 자리가 없습니다!!')
```

```
total = 20 #통학 버스 안의 총 자리수
```

```
present_seat = 0
```

```
for present_seat in reversed(range(20)):
```

```
    print('남은 자리 수:', present_seat)
```

```
    if present_seat==0:
```

```
        print('남은 자리가 없습니다!!')
```

예제 실행 결과

남은 자리 수: 19

남은 자리 수: 18

남은 자리 수: 17

남은 자리 수: 16

남은 자리 수: 15

남은 자리 수: 14

남은 자리 수: 13

남은 자리 수: 12

남은 자리 수: 11

남은 자리 수: 10

남은 자리 수: 9

남은 자리 수: 8

남은 자리 수: 7

남은 자리 수: 6

남은 자리 수: 5

남은 자리 수: 4
남은 자리 수: 3
남은 자리 수: 2
남은 자리 수: 1
남은 자리 수: 0
남은 자리가 없습니다!!

문제 풀이

잘 이해되지 않는 것들

`range(20)`앞에 `reversed` - 리스트 내림차순으로 정렬할때 `reverse`는 배웠는데 `reversed`는 잘 몰랐다.

책 외부에서 알게 된 내용

`reversed()`: 파이썬 내장함수로 리스트나 튜플등의 원소를 뒤집을 때 사용한다.
리스트 자료형은 `reverse()`메소드가 존재하기 때문에 `reversed()`함수 없이 역순환을 수행할 수 있지만 튜플은 `reverse()`메소드가 존재하지 않기 때문에 `reversed()`함수를 유용하게 사용할 수 있다.

회고

잘한 것은 무엇인가?

내가 만들어보고 싶은 프로그램에 대한 소스코드를 짜본것

잘못한 것은 무엇인가?

간단한 코드인데 1씩 줄어들고 0까지 포함시키는 것을 못하겠어서 해맸다.

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

나중에 최종으로 만들때 단순히 남은 자리 수만 아는 것이 아니라 좌석 번호를 부여해서 해당 번호가 찻는지 알 수 있도록 소스코드를 작성해보고 싶다.
한번에 여러명 탈때도 추가

내용 간단 요약

UI설계: 남은자리수, 탑승할 인원 입력, 남은자리, 남은 자리가 없습니다!!

논리구조 :

1. 통학 버스 안의 총 자리수 설정
2. 남은 자리 출력
3. 탑승할 인원수 입력
4. 원래 남은 자리에서 탑승인원 뺀 새로운 남은 자리 출력
5. 남은 자리가 양수인 범위에서 while문 반복
6. 남은 자리가 음수가 되면 '다시 시도해주세요'출력
7. 남은 자리가 0이되면 '남은 자리가 없습니다!!'출력

#통학버스 남은 자리 수 구하기

seat_count = 20 #통학 버스 안의 총 자리수

```
while seat_count > 0:
```

```
    print('남은 자리: ',seat_count)
```

```
    come_people = int(input("탑승할 인원수를 입력해주세요 :"))
```

```
    seat_count = seat_count - come_people
```

```
if seat_count < 0:
```

```
    print('남은자리가 음수입니다. 다시 시도 해주세요')
```

```
else: ## 음수가 아니라면 남은 자리가 없다고 출력
```

```
    print('남은 자리가 없습니다!!')
```

예제 실행 결과

남은 자리: 20

탑승할 인원수를 입력해주세요 :2

남은 자리: 18

탑승할 인원수를 입력해주세요 :3

남은 자리: 15

탑승할 인원수를 입력해주세요 :2

남은 자리: 13

탑승할 인원수를 입력해주세요 :1

남은 자리: 12

탑승할 인원수를 입력해주세요 :5

남은 자리: 7

탑승할 인원수를 입력해주세요 :2

남은 자리: 5

탑승할 인원수를 입력해주세요 :2
남은 자리: 3

탑승할 인원수를 입력해주세요 :3
남은 자리가 없습니다!!

문제 풀이

잘 이해되지 않는 것들

탑승할 인원수를 input으로 사용자가 입력하는데 어차피 정수값 n명으로 입력하는데 왜 int 써야하는지 모르겠다.

책 외부에서 알게 된 내용

for문, while문은 서로 변환이 가능하기 때문에 반복문을 작성할 때 어느 쪽을 선택해도 좋지만, for문은 반복 횟수를 알고 있을 때 주로 사용하고 while문은 조건에 따라 반복할 때 주로 사용한다.

while 문은 조건식이 true 일 경우에 계속해서 반복한다.

조건식에는 비교 또는 논리 연산식이 주로오는데, 조건식이 false가 되면 반복 행위를 멈추고 while문을 종료한다.

회고

잘한 것은 무엇인가?

여러명이 탈때도 고려해서 코드를 새로 짜봤다.

잘못한 것은 무엇인가?

자리번호 매겨서 하는건 못했다.

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

구글링을 더 하고 다양하게 시도해볼것이다.

내용 간단 요약

UI설계: Enter를 누르면 타이머를 시작합니다.

Enter를 누르면 측정을 종료합니다.

시작 후 gap초의 시간이 흘렀습니다.

논리구조

1. time모듈을 import해준다.
2. Enter를 누르면 타이머를 시작합니다.문구 출력
3. time.time()함수를 이용해 현재 시간을 입력받는다.
4. Enter를 누르면 측정을 종료합니다.문구 출력
5. time.time()함수를 이용해 종료 시간을 입력받는다.
6. 시작시간과 종료시간의 차이를 gap이라는 변수로 설정한다.
7. gap값을 round함수를 이용해서 소수점 셋째 자리까지 출력하게 한다.

```
import time
```

```
start = input("Enter를 누르면 측정을 시작합니다.")
```

```
begin = time.time()
```

```
print(begin)
```

```
stop = input("Enter를 누르면 측정을 종료합니다.")
```

```
end = time.time()
```

```
print(end)
```

```
gap = end - begin
```

```
gap = round(gap,3)
```

```
print("시작 후", gap, "초의 시간이 흘렀습니다.")
```

Enter를 누르면 측정을 시작합니다.

1653192976.552657

Enter를 누르면 측정을 종료합니다.

1653192977.398122

시작 후 0.845 초의 시간이 흘렀습니다.

문제 풀이

잘 이해되지 않는 것들

input 값 입력해주는게 enter치는건가?

책 외부에서 알게 된 내용

time 모듈은 운영 체제가 제공하는 다양한 시간 기능을 다루는 모듈이다. 운영 체제마다 시간을 다루는 방식이 다르기 때문에 이 모듈의 함수들은 어떤 운영 체제에서 실행하느냐에 따라 결과가 다르다. time.time() 함수는 컴퓨터의 현재 시각을 구하는 함수이며, 반환 값 기준 시각(운영 체제마다 다름)에서 몇 초가 지났는지를 나타내는 실수다.

round함수는 숫자를 지정한 자리수로 반올림한다.

회고

잘한 것은 무엇인가?

새로운 모듈을 써봤다.

잘못한 것은 무엇인가?

책에서 안찾아본것

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

책에 있는 것도 해보겠다.

내용 간단 요약

UI설계: [1]번호 is a phone number:

[2]True

[3]번호(숫자)가 아닌 것 is a phone number:

[4]False

논리구조

1. def함수를 사용해서 함수이름을 isPhoneNumber로 설정한다.
2. if문을 써서 text라는 매개변수에 길이가 12가 아니면 False를 반환한다.
3. i라는 변수가 0~2의 범위에서, 숫자가 아니면 False를 반환한다.
4. 4번째 자리가 문자열 '-'가 아니면 False를 반환한다.
5. i라는 변수가 4~6의 범위에서, 숫자가 아니면 False를 반환한다.
6. 8번째 자리가 문자열 '-'가 아니면 False를 반환한다.
7. i라는 변수가 8~11의 범위에서, 숫자가 아니면 False를 반환한다.

```
def isPhoneNumber(text):
    if len(text) != 12:
        return False
    for i in range(0, 3):
        if not text[i].isdigit():
            return False
    if text[3] != '-':
        return False
    for i in range(4, 7):
        if not text[i].isdigit():
            return False
    if text[7] != '-':
        return False
    for i in range(8, 12):
        if not text[i].isdigit():
            return False
    return True
print('415-555-4242 is a phone number:')
print(isPhoneNumber('415-555-4242'))
print('Moshi moshi is a phone number:')
print(isPhoneNumber('Moshi moshi'))
```

```
415-555-4242 is a phone number:
True
Moshi moshi is a phone number:
False
```

문제 풀이

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

isdecimal(): 문자열로 존재하는 숫자들을 골라내거나 정수화 시켜주기 위해서 사용한다.

주어진 문자열이 int형으로 변환이 가능한지 알아내는 함수이다.

-isdigit() , isnumeric()

이 두개의 함수와 isdecimal()은 분명한 차이점이 있다.

isdigit(), isnumeric() 은 3²에서 ²는 특수문자이지만 이 두함수는 True 반환을 한다.

isdigit() -> '숫자' 모양이라면 무조건 True를 반환

isnumeric() -> 숫자값 표현에 해당하는 문자열까지 인정. (제곱근 및 분수, 거듭제곱 특수문자)

회고

잘한 것은 무엇인가?

새로운 함수를 써봤다.

잘못한 것은 무엇인가?

배웠던 함수랑 비슷한 기능인데 바로 안떠올랐다.

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

수업에서 배운것과 연관짓는 활동을 해보고 싶다.

내용 간단 요약

UI설계: 비밀번호를 입력하세요:

n번째 자리 맞혔습니다!!

비밀번호를 맞혔습니다!!

비밀번호:

논리구조

1. 비밀번호 설정
2. 사용자한테 비밀번호 입력받기
3. 사용자가 입력한 비밀번호와 설정된 비밀번호가 같으면 비밀번호 맞혔습니다 출력
4. 같지 않으면 맞은 자리숫자 맞혔다고 출력
5. 비밀번호가 틀렸습니다 출력

```
password = "
```

```
user_input = input('비밀번호를 입력하세요:')
```

```
if user_input == password:
```

```
    print('비밀번호를 맞혔습니다!!')
```

```
    print('tmi : ')
```

```
else:
```

```
    if password[0] == user_input[0]:  
        print('첫번째 자리 맞혔습니다!!')
```

```
    if password[1] == user_input[1]:  
        print('두번째 자리 맞혔습니다!!')
```

```
    if password[2] == user_input[2]:  
        print('세번째 자리 맞혔습니다!!')
```

```
    if password[3] == user_input[3]:  
        print('네번째 자리 맞혔습니다!!')
```

```
print('비밀번호가 틀렸습니다!! 다시 입력해주세요')
```

비밀번호를 입력하세요:

첫번째 자리 맞혔습니다!!

두번째 자리 맞혔습니다!!

비밀번호가 틀렸습니다!! 다시 입력해주세요

비밀번호를 입력하세요:

비밀번호를 맞혔습니다!!

tmi :

문제 풀이

잘 이해되지 않는 것들

책 외부에서 알게 된 내용

EXE는 일반적인 파일 확장자로 컴퓨터 프로그램의 실행 파일을 가리킨다. 오픈VMS, 도스, 마이크로소프트 윈도우, 리엑트오에스, OS/2 운영 체제에서 사용할 수 있다.

회고

잘한 것은 무엇인가?

첨에 이상하게 했다가 잘 돌아감

잘못한 것은 무엇인가?

첨에 이상하게 함

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

향후 어떻게 다르게 할 것 인가?

열심히한다.

내용 간단 요약

코드를 깨보려고 했는데 내 코드를 주아가 깨려고 하니까 종료되었다.
문제를 파악하고 해결하려고 주아랑 머리를 맞대고 생각해봤다.
코드가 반복을 안하고 기회를 한번만 줘서 맞출 수 가 없게 되어있었다.
그래서 반복을 하도록 코드를 다시 짜보았다.

UI설계: 비밀번호를 입력하세요:

n번째 자리 맞혔습니다!!

비밀번호를 맞혔습니다!!

비밀번호:

논리구조

tool사용하는 법

책 외부에서 알게 된 내용

가상 머신(Virtual Machine, VM)은 물리적 하드웨어 시스템(오프프레미스 또는 온프레미스에 위치)에 구축되어 자체 CPU, 메모리, 네트워크 인터페이스 및 스토리지를 갖추고 가상 컴퓨터 시스템으로 작동하는 가상 환경이다. 하이퍼바이저라 불리는 소프트웨어는 하드웨어에서 가상 머신의 리소스를 분리하고 적절히 프로비저닝하여 VM에서 사용할 수 있도록 한다.

- 존 더 리퍼(John the Ripper)

Jo사전 공격을 통해 비밀번호 강도를 테스트하고

암호화된 비밀번호 크래킹 도구이다. 비밀번호를 무차별 대입 공격해 비밀번호를 크래킹 하기위해 만들어졌다.

john the Ripper는 Solar Designer가 개발한 Unix계열 password crack tool이다.

- 사전 공격: 이런 유형의 공격에서 도구는 사전에 입력된 많은 수의 단어, 구문, 이전에 유출된 데이터에서 얻은 가능성 있는 비밀번호 목록이 제공하는 비밀번호로 로그인을 시도한다. 이 도구는 올바른 비밀번호를 찾기 위해 애플리케이션에 목록의 모든 단어를 입력한다.

- 무차별 대입 공격: 이 유형의 공격에서 도구는 사용자에게 올바른 비밀번호의 최소 및 최대 길이와 구성될 가능성이 있는 글자 유형(문자, 문자+숫자, 특수 문자 등) 및 위치(예를 들어, 생성하는 모든 비밀번호는 처음 4글자가 알파벳이고 두 자리 숫자와 두 개의 특수 문자가 뒤에 추가됨) 등 몇 가지를 구성하도록 요청한다.

이상적인 대입 구성을 찾기 위해 약간의 추측과 전문 지식이 필요하다. 그리고 도구는 이 범위 안에서 기준에 따라 가능한 모든 비밀번호 조합을 추측한다.

올바른 비밀번호를 찾아내면 사용자에게 알린다. 이 과정이 효과적이지만 매우 느릴 수 있다. 예를 들어, 대소문자와 함께 숫자와 특수 문자가 혼합된 9자리 비밀번호는 컴퓨터가 추측하는 데 9년 이상이 소요되기 때문에 실제로 크래킹이 불가능하다. 그래서 보안 전문가들은 항상 다양한 문자 유형의 조합으로 구성된 길고 복잡한 비밀번호를 선택하라고 제안하는 것이다.

- 레인보우 테이블(Rainbow tables): 업무에 필수적이고 보안 지향적인 애플리케이션은 비밀번호를 평문으로 저장하는 경우가 거의 없고 길이가 고정된 해시를 저장한다. 때문에 유출된 데이터 등으로부터 얻은 해시된 비밀번호 목록이 있는 경우에 레인보우 테이블이 효율적일 수 있다.

이 경우, 사전에 연산된 비밀번호 해시 목록을 기존의 데이터와 비교해 평문 형식의 올바른 비밀번호를 찾는다. 해시된 데이터가 미리 계산되기 때문에 레인보우 테이블을 사용하는 것이 무차별 대입보다 빠르다.

레인보우 테이블은 비밀번호 해시가 솔팅(Salting)되어 있고 솔트 값이 너무 커 전체적인 복잡성이 증가할 때 효과적이지 못하다. 그래서 해시된 사용자 비밀번호를 데이터베이스에 저장하는 것 외에 솔팅을 보안 방어책으로 사용하는 것이다. 솔팅을 제대로 적용하면 비밀번호 데이터베이스가 유출되더라도 실제로 해커가 사용자 비밀번호를 본래의 평문 형식으로 되돌리는 것이 불가능하다.

회고

잘한 것은 무엇인가?

전에는 실행까지 해봤는데도 찾지 못했던 코드의 문제를 다시한번 보면서 알아내서 해결한 점.

잘못한 것은 무엇인가?

과제로 내주신 tool을 이용해서 하는 것을 이해하지 못해서 과제를 제대로 못한 점.

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

모르는게 너무 많다...

향후 어떻게 다르게 할 것 인가?

tool에 대해서 공부한다.

책 12장

내용 간단 요약

openpyxl 모듈은 파이썬 프로그램을 읽고 엑셀 파일을 수정할 수 있게 해준다. Windows, OS X 및 Linux에서 실행되는 무료 대안이 있다. LibreOffice Calc와 OpenOffice Calc는 모두 스프레드시트용 Excel의 .xlsx 파일 형식과 함께 작동한다.

이는 openpyxl 모듈이 이러한 응용 프로그램의 스프레드시트에서도 작동할 수 있음을 의미한다.

소프트웨어는 각각 <https://www.libreoffice.org/> 및 <http://www.openoffice.org/>에서 다운로드할 수 있다.

Excel spreadsheet document는 workbook이라고 불린다.

단일 통합 문서는 확장명이 .xlsx인 파일에 저장된다. 각 통합 문서에는 여러 시트(워크시트라고도 함)가 포함될 수 있다. 사용자가 현재 보고 있는(Excel을 닫기 전에 마지막으로 본) 시트를 활성 시트라고 한다.

각 시트에는 열(A에서 시작하는 문자로 주소 지정)과 행이 있다.

(1부터 시작하는 숫자로 주소 지정). 특정 열과 행에 있는 상자를 셀이라고 한다. 각 셀에는 숫자 또는 텍스트 값이 포함될 수 있다. 데이터가 있는 셀 그리드가 시트를 구성한다.

Installing the openpyxl Module

모듈의 이름은 openpyxl이다. 올바르게 설치되었는지 테스트하려면 import openpyxl을 입력해야 한다.

모듈이 올바르게 설치되면 오류 메시지가 생성되지 않는다. 이 장의 대화형 셀 예제를 실행하기 전에 openpyxl 모듈을 가져오는 와야 한다. 그렇지 않으면 NameError: name 'openpyxl' is not defined 오류가 발생한다.

최신 버전이 있고 어떤 추가 기능을 사용할 수 있는지 확인하려면

<http://openpyxl.readthedocs.org/>에서 확인할 수 있다.

Reading Excel Documents

이 장의 예제에서는 루트 폴더에 저장된 example.xlsx라는 스프레드시트를 사용한다.

spreadsheet를 직접 만들거나 <http://nostarch.com/automatestuff/>에서 다운로드할 수 있다.

그림 12-1은 Excel이 새 통합 문서에 대해 자동으로 제공하는 Sheet1, Sheet2 및 Sheet3이라는 세 개의 기본 시트에 대한 탭을 보여준다. (생성되는 기본 시트의 수는 운영 체제와 스프레드시트 프로그램에 따라 다를 수 있다.)

예제 파일의 시트 1은 표 12-1과 같아야 합니다. (웹사이트에서 example.xlsx를 다운로드하지 않았다면 이 데이터를 시트에 직접 입력해야 한다.)

-Opening Excel Documents with OpenPyXL(openpyxl 모듈을 사용하여 이를 조작하는 방법)

openpyxl 모듈을 가져오면 openpyxl.load_workbook() 함수를 사용할 수 있고 이를 입력한다. openpyxl.load_workbook() 함수는 파일 이름을 가져와서 통합 문서 데이터 유형의 값을 반환한다. 이 Workbook 개체는 Excel 파일을 나타내며, File 개체가 열린 텍스트 파일을 나타내는 것과 비슷하다.

example.xlsx로 작업하려면 현재 작업 디렉토리에 있어야하므로 os를 가져오고 os.getcwd()를 사용하여 현재 작업 디렉토리 확인가능, 다음을 수행가능. os.chdir()을 사용하여 현재 작업 디렉토리를 변경.

Getting Sheets from the Workbook

get_sheet_names() 메서드를 호출하여 통합 문서의 모든 시트 이름 목록을 가져올 수 있다. 각 시트는 시트 이름 문자열을 get_sheet_by_name() 통합 문서 메서드에 전달하여 얻을 수 있는 Worksheet 객체로 표시된다. 마지막으로 Workbook 객체의 get_active_sheet() 메서드를 호출하여 통합 문서의 활성 시트를 가져올 수 있다. Worksheet 객체가 있으면 제목 속성에서 이름을 가져올 수 있다.

Getting Cells from the Sheets

Cell 객체에는 당연히 해당 셀에 저장된 값이 포함된 값 속성이 있다. 셀 객체에는 셀의 위치 정보를 제공하는 행, 열 및 좌표 속성이 있다.

'Apples' 문자열을 사용. 행 속성은 정수 1을, 열 속성은 'B'를, 좌표 속성은 'B1'을 제공 OpenPyXL은 A 열의 날짜를 자동으로 해석하고 문자열이 아닌 날짜/시간 값으로 반환 Z 열 다음에 열이 AA, AB, AC 등의 두 문자를 사용하여 시작해서 까다로울수도. 대안으로 시트의 cell() 메서드를 사용하고 행 및 열 키워드 인수에 정수를 전달하여 셀을 가져올 수도 있다. 첫 번째 행 또는 열 정수는 0이 아니라 1. 시트의 cell() 메서드를 사용하여 row=1을 전달합니다.

그리고 column=2는 sheet['B1']을 지정하는 것과 마찬가지로 셀 B1에 대한 Cell 객체를 가져옴. 그런 다음 cell() 메서드와 키워드 인수를 사용하여 일련의 셀 값을 인쇄하는 for 루프를 작성 o.

for 루프의 i 변수는 행 키워드 인수에 대해 cell() 메서드에 전달되는 반면 2는 항상 열 키워드 인수에 대해 전달됩니다. 문자열 'B'가 아닌 정수 2가 전달된다는 점에 유의.

Worksheet 객체의 get_highest_row() 및 get_highest_column() 메서드를 사용하여 시트의 크기를 결정 o.

get_highest_column() 메서드는 Excel에 나타나는 문자가 아닌 정수를 반환.

Converting Between Column Letters and Numbers

openpyxl.cell 모듈에서 이 두 함수를 가져온 후 get_column_letter()를 호출하고 27과 같은 정수를 전달하여 27번째 열의 문자 이름이 무엇인지 알아낼 수 o. column_index_string() 함수는 반대로 열의 문자 이름을 전달하면 해당 열이 몇 번째인지 알려줌. 통합 문서를 로드하고 Worksheet 개체를 가져오고 get_highest_column()과 같은 Worksheet 개체 메서드를 호출하여 정수를 얻음. 그런 다음 해당 정수를 get_column_letter()에 전달.

Getting Rows and Columns from the Sheets

워크시트 객체를 슬라이스하여 스프레드시트의 행, 열 또는 직사각형 영역에 있는 모든 셀 개체를 가져올 수 있다. 그런 다음 슬라이스의 모든 셀을 반복할 수 있다. 대화형 셀에 다음을 입력한다.

여기에서 A1에서 C3까지의 직사각형 영역에 Cell 객체를 원한다고 지정하고 해당 영역에 Cell 객체를 포함하는 Generator 객체를 얻다. 이 Generator 객체를 시각화하는 데 도움이 되도록 tuple()을 사용하여 튜플에 Cell 객체를 표시할 수 있다.

이 튜플은 3개의 튜플을 포함한다. 원하는 영역의 맨 위에서 맨 아래까지 각 행마다 하나씩. 이 세 개의 내부 튜플 각각은 가장 왼쪽 셀에서 오른쪽으로 원하는 영역의 한 행에 Cell 개체를 포함.

따라서 전반적으로 시트 조각에는 왼쪽 상단 셀에서 시작하여 오른쪽 하단 셀로 끝나는 A1에서 C3까지의 영역에 있는 모든 Cell 개체가 포함.

영역의 각 셀 값을 인쇄하려면 두 개의 for 루프를 사용. 바깥쪽 for 루프는 슬라이스의 각 행을 지나갑니다. 그런 다음 각 행에 대해 중첩 for 루프가 해당 행의 각 셀을 통과.

특정 행이나 열의 셀 값에 액세스하려면 Worksheet 객체의 rows and columns 속성을 사용할 수도 있다. 대화형 셀에 다음을 입력.

Workbooks, Sheets, Cells

모든 기능, 방법 및 데이터에 대한 요약.

스프레드시트 파일에서 셀 읽기와 관련된 유형:

1. openpyxl 모듈을 import
2. openpyxl.load_workbook() 함수를 호출
3. 통합 문서 개체 get
4. get_active_sheet() 또는 get_sheet_by_name() 통합 문서 메서드를 호출
5. Worksheet 개체 get
6. 행 및 열 키워드인수와 함께 인덱싱 또는 cell() 시트 방법 사용
7. Cell 개체를 get
8. Cell 개체의 값 속성을 읽기

Worksheet 개체에서 rows 속성을 사용하면 튜플을 얻을 수 있다. 이러한 내부 튜플은 각각 행을 나타내며 해당 행에 Cell 객체를 포함. columns 속성은 또한 특정 열의 Cell 객체를 포함하는 내부 튜플 각각과 함께 튜플을 제공. 예를 들어.xlsx에는 7개의 행과 3개의 열이 있으므로 행은 7개의 튜플(각각 3개의 Cell 객체 포함)의 튜플을 제공하고 열은 3개의 튜플(각각 7개의 Cell 객체 포함)을 제공.

하나의 특정 튜플에 액세스하려면 더 큰 튜플의 인덱스로 참조할 수 있다. 예를 들어, 열 B를 나타내는 튜플을 얻으려면 sheet.columns[1]을 사용. 열 A의 Cell 객체를 포함하는 튜플을 얻으려면 sheet.columns[0]을 사용. 하나의 행이나 열을 나타내는 튜플이 있으면 해당 Cell 개체를 반복하고 해당 값을 인쇄할 수 있다.

예제 실행 결과

```
import openpyxl
```

```
>>> import openpyxl
>>> wb = openpyxl.load_workbook('example.xlsx')
>>> type(wb)
<class 'openpyxl.workbook.workbook.Workbook'>
```

```
>>> import openpyxl
>>> wb = openpyxl.load_workbook('example.xlsx')
>>> wb.get_sheet_names()
['Sheet1', 'Sheet2', 'Sheet3']
>>> sheet = wb.get_sheet_by_name('Sheet3')
>>> sheet
<Worksheet "Sheet3">
>>> type(sheet)
<class 'openpyxl.worksheet.worksheet.Worksheet'>
>>> sheet.title
'Sheet3'
>>> anotherSheet = wb.get_active_sheet()
>>> anotherSheet
<Worksheet "Sheet1">
```

```
>>> import openpyxl
>>> wb = openpyxl.load_workbook('example.xlsx')
>>> sheet = wb.get_sheet_by_name('Sheet1')
>>> sheet['A1']
<Cell Sheet1.A1>
>>> sheet['A1'].value
datetime.datetime(2015, 4, 5, 13, 34, 2)
>>> c = sheet['B1']
>>> c.value
'Apples'
>>> 'Row ' + str(c.row) + ', Column ' + c.column + ' is ' + c.value
'Row 1, Column B is Apples'
>>> 'Cell ' + c.coordinate + ' is ' + c.value
'Cell B1 is Apples'
>>> sheet['C1'].value
73
```

```
>>> sheet.cell(row=1, column=2)
<Cell Sheet1.B1>
>>> sheet.cell(row=1, column=2).value
'Apples'
>>> for i in range(1, 8, 2):
```



```
print(i, sheet.cell(row=i, column=2).value)
```

```
1 Apples
```

```
3 Pears
```

```
5 Apples
```

```
7 Strawberries
```

```
>>> import openpyxl
```

```
>>> wb = openpyxl.load_workbook('example.xlsx')
```

```
>>> sheet = wb.get_sheet_by_name('Sheet1')
```

```
>>> sheet.get_highest_row()
```

```
7
```

```
>>> sheet.get_highest_column()
```

```
3
```

```
>>> import openpyxl
```

```
>>> from openpyxl.cell import get_column_letter, column_index_from_string
```

```
>>> get_column_letter(1)
```

```
'A'
```

```
>>> get_column_letter(2)
```

```
'B'
```

```
>>> get_column_letter(27)
```

```
'AA'
```

```
>>> get_column_letter(900)
```

```
'AHP'
```

```
>>> wb = openpyxl.load_workbook('example.xlsx')
```

```
>>> sheet = wb.get_sheet_by_name('Sheet1')
```

```
>>> get_column_letter(sheet.get_highest_column())
```

```
'C'
```

```
>>> column_index_from_string('A')
```

```
1
```

```
>>> column_index_from_string('AA')
```

```
27
```

```
>>> import openpyxl
```

```
>>> wb = openpyxl.load_workbook('example.xlsx')
```

```
>>> sheet = wb.get_sheet_by_name('Sheet1')
```

```
>>> tuple(sheet['A1':'C3'])
```

```
((<Cell Sheet1.A1>, <Cell Sheet1.B1>, <Cell Sheet1.C1>), (<Cell Sheet1.A2>,  
<Cell Sheet1.B2>, <Cell Sheet1.C2>), (<Cell Sheet1.A3>, <Cell Sheet1.B3>,  
<Cell Sheet1.C3>))
```

```
>>> for rowOfCellObjects in sheet['A1':'C3']:
```

```
    for cellObj in rowOfCellObjects:
```

```
    print(cellObj.coordinate, cellObj.value)
```

```
print('--- END OF ROW ---')
```

```
A1 2015-04-05 13:34:02
```

```
B1 Apples
```

```
C1 73
```

```
--- END OF ROW ---
```

A2 2015-04-05 03:41:23
B2 Cherries
C2 85
--- END OF ROW ---
A3 2015-04-06 12:46:51
B3 Pears
C3 14
--- END OF ROW ---

```
>>> import openpyxl
>>> wb = openpyxl.load_workbook('example.xlsx')
>>> sheet = wb.get_active_sheet()
>>> sheet.columns[1]
(<Cell Sheet1.B1>, <Cell Sheet1.B2>, <Cell Sheet1.B3>, <Cell Sheet1.B4>,
<Cell Sheet1.B5>, <Cell Sheet1.B6>, <Cell Sheet1.B7>)
>>> for cellObj in sheet.columns[1]:
print(cellObj.value)
Apples
Cherries
Pears
Oranges
Apples
Bananas
Strawberries
```

문제 풀이

잘 이해되지 않는 것들

workbook이라는게 엑셀파일하나?
openpyxl모듈을 사용하려면 확장자가 .xlsx이어야한다?
271마지막- rows속성을 사용하면 튜플을 얻을수 있다고 했는데 그러면 다른 메서드를 사용하면 튜플말고 다른 형태로 얻어지나??

책 외부에서 알게 된 내용

워크시트: 스프레드 시트 프로그램에서 전산 입력용 데이터 시트와 마찬가지로 데이터를 일정한 양식으로 정리할 수 있도록 구성된 시트. 행과 열을 갖는 셀로 구성.
스프레드 시트: 데이터의 조작,분석,저장을 가능케하는 상호작용 컴퓨터 애플리케이션이다.
workbook개체: 파일열기,닫기,저장
range개체: 셀 선택하기,셀 값 속성
worksheet개체:워크시트 만들기, 삭제하기,선택하기,보호하기

회고

잘한 것은 무엇인가?

열심히 했다.

잘못한 것은 무엇인가?

모르는 함수가 많아서 기억에 남지가 않았다.

아직도 안 풀린 궁금증은 무엇인가? 왜 풀리지 않았는가?

생소한 함수들이라 전반적으로 잘 모르겠다.

향후 어떻게 다르게 할 것 인가?

엑셀쓸일 생기면 스프레드시트를 이용해봐야겠다.

