

## 22. 08 . 12 金 \_ 단어 맞추기 게임

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b26f2570-9c05-41f0-84d3-9492d4689dde/%EB%8B%A8%EC%96%B4\\_%EB%A7%9E%EC%B6%94%EA%B8%B0\\_%EA%B2%8C%EC%9E%84.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b26f2570-9c05-41f0-84d3-9492d4689dde/%EB%8B%A8%EC%96%B4_%EB%A7%9E%EC%B6%94%EA%B8%B0_%EA%B2%8C%EC%9E%84.pdf)

### 1. 단어 맞추기 게임 \_ 재일 작성

```
import random # 랜덤모듈 가져오기

#####

# F1. 함수 구현 리스트
def Output_Unit (Number) :
    Suffix = ["첫", "두", "세", "네", "다섯", "여섯", "일곱", "여덟", "아홉", "열"]
    Count = 0

    while len(Input_Words_List) < 3 :
        Word = input(str(Suffix[Count])+" 번째 단어를 입력 하세요\n")
        print()
        if 5 <= len(Word) <= 20 :
            Input_Words_List.append(Word)
            Count += 1
        else :
            print("5이상 20이하 글자로 구성된 단어를 입력 하세요.")
            print()

# F2. 올림 함수
def Raising (Word_Selected) :

    Blind_Count = 0 # 블라인드 처리 수

    Half_Nun= len(Word_Selected)/2

    # Half_Nun 짝수 일 때 반올림
    if type(Half_Nun) == float and len(Word_Selected)%2 == 0:
        Blind_Count = int(Half_Nun)

    # Half_Nun 홀수 일 때 반올림
    elif type(Half_Nun) == float and len(Word_Selected)%2 != 0:
        Blind_Count = int(Half_Nun)+1

    return Blind_Count

# F3. 출력문 반복 함수
def Trial_Count (Counting,BlankWord_List) :
    print(f'{Counting} 번째 시도, 아래 단어를 구성하는 알파벳 한 개를 입력하세요')
    # 현재 알고 있는 알파벳 출력
    for Element_BlankWord_List in BlankWord_List :
        print(Element_BlankWord_List,end=" ")
    print()
    print()

#####

Input_Words_List = [ ] # 키보드로 부터 받은 영단어 3 개 저장

# 1. 키보드로부터 영어 단어 3개를 입력 받아 리스트에 저장
Output_Unit (3)

# 2. 입력된 3개의 단어 중 한 개 단어를 임의 선택
Random_Index= random.randint(0,2) # 랜덤 인덱스

Word_Selected = Input_Words_List[Random_Index] # 랜덤 단어

# 3. 게임 시작을 알리는 문장 출력 => 단어 선택 완료 게임을 시작 합니다. 선택된 단어 :
if len(Input_Words_List) == 3 :
    print(f"단어 선택 완료 게임을 시작 합니다. 선택된 단어 : {Word_Selected}")
```

```

# 4. 선택된 단어의 글자 중 50%를 Blind 처리, Blind 처리 알파벳은 랜덤하게 선택

Blind_Num_Word = Raising(Word_Selected) # 올림처리 한 블라인드 갯수
Random_List = [] # 랜덤으로 얻은 리스트
BlankWord_List = [] # 블랭크 처리된 단어 리스트
BlankWord_Str = "" # 블랭크 처리된 단어 변수

# 4.1 반복문을 사용한 랜덤으로 단어 가리기
for Index in range (len(Word_Selected)) : # 반복 횟수는 단어 길이 만큼

    # 엘리먼트 추출
    Element = Word_Selected[Index]

    # 블라인드 처리
    while len(Random_List) < Blind_Num_Word :
        RANDOM_Num = random.randint(0, len(Word_Selected))

        # Random_List 요소 생성
        if not RANDOM_Num in Random_List :
            Random_List.append(RANDOM_Num)

    # Random_List 안에 있으면 "_" 처리
    if Index in Random_List :
        Element = "_"
        BlankWord_List.append(Element)
        BlankWord_Str += Element
    else :
        BlankWord_List.append(Element)
        BlankWord_Str += Element

print()
#####

# 5. 문제 맞추기

Duplicated_Num = 0 # 포함된 알파벳 알릴 수
Counting = 1 # 시도 횟수 측정

# 5.1 문제 맞출시 까지 반복
while True :

    #출력문 (함수사용 F3 )
    Trial_Count (Counting,BlankWord_List)
    # 입력값 받기
    InputNum_Trial = input()
    print()

    # 5.1.1 Blank 처리된 리스트 요소들과 "단어리스트" 비교
    for Index in range (len(BlankWord_List)):
        # 블랭크 리스트에 있는 요소들 하나씩 뽑기
        Element = BlankWord_List[Index]
        # 만약 "입력값"과 "단어리스트"의 값이 같다면
        if Word_Selected[Index] == InputNum_Trial :
            # 공백 부분을 "입력값"으로 교환
            BlankWord_List[Index] = InputNum_Trial
            Duplicated_Num+=1
        # 입력 값이 해당 단어에 없을 경우
        elif not InputNum_Trial in Word_Selected :
            print("단어 내 포함되지 않은 알파벳 입니다.")
            print()
            Counting += 1

    #출력문 (함수사용 F3 )
    Trial_Count (Counting,BlankWord_List)
    # 입력값 받기
    InputNum_Trial = input()
    print()
    continue

    # 5.1.2 알파벳을 하나씩 입력받아 단어안에 들어 있으면
    if not "_" in BlankWord_List :
        # 프로그램 종료전 출력문
        print("Clear - 선택된 단어 : ",Word_Selected," , 총 시도 횟수 : ",Counting,)
        break

    # 출력문
    print("입력한 알파벳 단어 내 포함 : ", Duplicated_Num,"글자")

    Counting += 1 # 반복문 횟수 세기
    Duplicated_Num = 0 # 포함된 단어 초기화

```

