

봉석&세정이가 문제 났지롱

파이썬

내 꿈은 우주대스타

1. Text-handling, 주어진 파일 (1984.txt) 을 불러오고 다음과 같은 함수를 작성하세요!

함수	설명
get_file_contents	문자열값으로 filename을 입력받아 해당 파일에 존재하는 모든 text 데이터를 문자열 형태로 반환함
get_number_of_characters_with_blank	문자열값으로 filename을 입력받아 해당 파일에 존재하는 모든 글자의 갯수를 integer 값으로 반환함
get_number_of_characters_without_blank	문자열값으로 filename을 입력받아 해당 파일에 존재하는 모든 글자의 갯수를 공백을 제외하고 integer 값으로 반환함. 단 여기서 공백은 " ", "\t", "\n" 을 의미함
get_number_of_lines	문자열값으로 filename을 입력받아 해당 파일에 존재하는 모든 줄(line)수를 integer 값으로 반환함. 이때 마지막 줄은 count에서 제외함
get_number_of_target_words	문자열값으로 filename과 찾고자하는 target_words를 입력받아 해당 파일에 존재하는 target_words와 같은 글자의 수를 대소문자와 상관없이 integer 값으로 반환함

참고하세요 !

https://github.com/TEAMLAB-Lecture/python-101/blob/master/2019-gahcon-ime/assignment/lab_9/README.md

<함수 작성 양식 참고>

```
def get_file_contents(filename):
    # '''
    # Input:
    # - filename : 문자열값으로 처리할 파일의 이름
    # Output:
    # - 파일에 나오는 모든 Text 데이터를 문자열로 반환
    # Examples:
    # >>> import file_io_example as fie
    # >>> fie.get_file_contents("1984.txt").split("\n")[0]
    # GEORGE ORWELL
    # ==Modify codes below=====
    contents = None

    # =====
    return contents
```

```
def get_number_of_characters_with_blank(filename):
    # '''
    # Input:
    # - filename : 문자열값으로 처리할 파일의 이름
    # Output:
    # - 빈칸을 포함하여 해당 파일에 나오는 글자 수의 총합
    # Examples:
    # >>> import file_io_example as fie
    # >>> fie.get_number_of_characters_with_blank("1984.txt")
    # 558841
    # ==Modify codes below=====
    result = None

    # =====
    return result
```

```
def get_number_of_characters_without_blank(filename):
    # '''
    # Input:
    # - filename : 문자열값으로 처리할 파일의 이름
    # Output:
    # - 빈칸을 포함하지 않는 해당 파일에 나오는 글자 수의 총합
    #   여기서 빈칸이라고 함은 " ", "\t", "\n" 을 모두 포함함
    # Examples:
    # >>> import file_io_example as fie
    # >>> fie.get_number_of_characters_without_blank("1984.txt")
    # 459038
    # ==Modify codes below=====
    result = None

    # =====
    return result
```

```
def get_number_of_lines(filename):
    # '''
    # Input:
    # - filename : 문자열값으로 처리할 파일의 이름
    # Output:
    # - 해당 파일의 모든 라인수의 총합
    #   단 마지막 줄 수는 제외함
    # Examples:
    # >>> import file_io_example as fie
    # >>> fie.get_number_of_lines("1984.txt")
    # 1414
    # ==Modify codes below=====
    result = None

    # =====
    return None
```

```
def get_number_of_target_words(filename, target_words):
    # '''
    # Input:
    # - filename : 문자열값으로 처리할 파일의 이름
    # - target_words : 문자열값으로 처리할 파일의 이름
    # Output:
    # - 대소문자 구분없이 해당 파일에 target_words가 포함된 횟수
    # Examples:
    # >>> import file_io_example as fie
    # >>> fie.get_number_of_words("1984.txt", "Hi")
    # 3938
    # >>> fie.get_number_of_words("1984.txt", "had")
    # 1327
    # >>> fie.get_number_of_words("1984.txt", "and")
    # 2750
    # ==Modify codes below=====
    result = None
```

결과확인

```
Type "help", "copyright", "credits" or "license" for more information.
>>> import file_io_example as fie
>>> fie.get_file_contents("1984.txt").split("\n")[0]
'GEORGE ORWELL'
>>> fie.get_number_of_characters_with_blank("1984.txt")
558840
>>> fie.get_number_of_characters_without_blank("1984.txt")
459038
>>> fie.get_number_of_lines("1984.txt")
1414
>>> fie.get_number_of_target_words("1984.txt", "Hi")
3938
>>> fie.get_number_of_target_words("1984.txt", "had")
1327
```

2. NumPy 활용, ex의 1번과 2번 축을 서로 바꿔보세요 ! (hint : ex.transpose())

<ex>

```
ex = np.array([ # B.shape == (2,2,3)
    [
        [ 0, 1, 2],
        [ 3, 4, 5]
    ],
    [
        [ 6, 7, 8],
        [ 9,10,11]
    ]
])
```

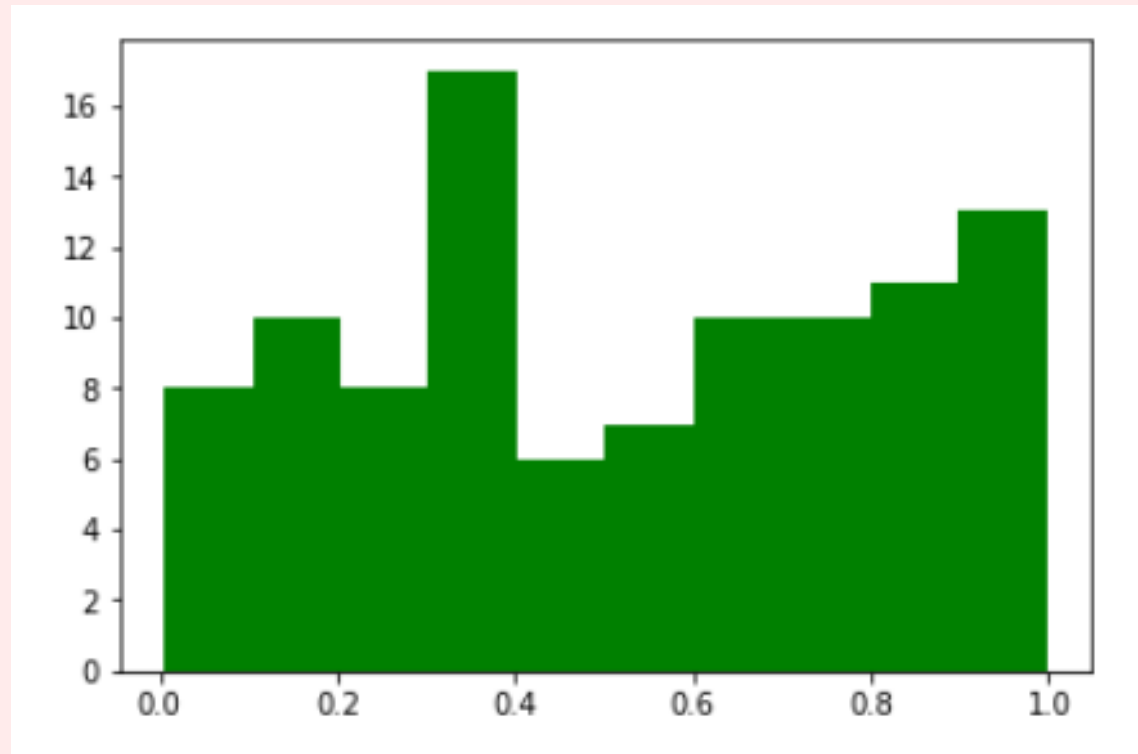
<작성 양식 참고>

```
# ex의 1번과 2번 축을 서로 바꿔보세요. (hint: ex.transpose())
# Note: shape 기준으로 (2,2,3) --> (2,3,2)
#
# desired output: [[[ 0  3]
#                  [ 1  4]
#                  [ 2  5]]
#
#                  [[ 6  9]
#                  [ 7 10]
#                  [ 8 11]]]
#
# ----- TODO ----- #
#
# 이부분을 지우고 쓰시면 됩니다. #
#
# ----- #
print(x)
```

3. NumPy + matplotlib 활용, 아래의 히스토그램을 그려보세요 !

- matplotlib 라이브러리를 불러올 것
- seed number를 826으로 설정해줄 것
- 표준정규분포 난수를 생성하는 `numpy.random.____(100)` 함수 사용 (100개)
- 히스토그램의 색깔은 'g'로 지정

결과확인



4. NumPy 활용, boolean indexing을 통해 array 추출 !

Array 생성 `a1 = np.arange(0, 30).reshape((5, 6))`

- even_arr라는 이름의 객체에 짝수를 나타내는 boolean 지정
- even_arr 이용하여 a1에서 짝수 indexing
- Indexing한 array를 3X5 형식으로 만들기

결과확인 `array([[0, 2, 4, 6, 8],
 [10, 12, 14, 16, 18],
 [20, 22, 24, 26, 28]])`

문제 끝났지롱

파이썬

내 꿈은 우주대스타