

# HomeWork3

- 기간 : 11월 22일부터 28일까지
- 파이썬 파일(.py)파일을 자신의 학번\_hw3\_1.py형식으로 저장하여 담당TA메일로 제출하기
- 각 프로그램 문장에 comment를 넣기 (프로그램 내에도 자신의 이름\_학번을 기술함)
  - Comment는 첫 줄에 #자신의 학번, 이름을 적으시고
  - 각 문장당 주석처리를 해주세요.

# 과제 1 : 단어의 출현 횟수를 count하기

- 주어진 example과 print문 내의 문장을 단어형태로 나뉘어 몇 번 나오는 지를 반환하는 함수를 작성하기.
- 인자값은 문자열을 입력받으며, 반환값은 튜플들의 리스트로 반환한다.
- 함수 : def dict(words):
- 호출과 결과값

```
if __name__ == '__main__':  
    example = """  
    you are learning a programming class  
    i hope you will learn funny class  
    """  
    print(dict(example))  
    print(dict('have a nice nice day day day'))
```

```
[('a', 1), ('are', 1), ('class', 2), ('funny', 1), ('hope', 1), ('i', 1), ('learn', 1), ('learning', 1), ('programming', 1), ('will', 1), ('you', 2)]  
[('a', 1), ('day', 3), ('have', 1), ('nice', 2)]
```

## 과제 2: pixel

- 2진수의 숫자표현으로 글꼴을 표시할 수 있다. 점이 있는 곳은 1(black), 없는 곳은 0(white)로 표시된다.

	■	■	■	
				■
	■	■	■	■
■				■
■				■
	■	■	■	■

0, 1, 1, 1, 0  
0, 0, 0, 0, 1  
0, 1, 1, 1, 1  
1, 0, 0, 0, 1  
1, 0, 0, 0, 1  
0, 1, 1, 1, 1



	■	■	■	
				■
	■	■	■	■
■				■
■				■
	■	■	■	■

1, 3, 1  
4, 1  
1, 4  
0, 1, 3, 1  
0, 1, 3, 1  
1, 4

- 점을 \*으로 표시하고, 점이 없는 곳은 빈칸 ' '으로 표시한 문자열을 숫자로 표현하거나 그 반대의 과정을 하는 함수를 작성하기. 또한 압축과 압축해제를 하는 함수를 작성하기

함수명	인자	반환값
def string_to_image(pic_str):	pic_str:문자열의 리스트. '*' 혹은 ' '으로 이루어진 문자열	리스트의 리스트: '*'은 1, ' '은 0으로 바뀌어 저장된 리스트의 리스트형
def image_to_str(umcompressed_img):	umcompressed_img:숫자로 이미지를 표현한 리스트	문자열의 리스트 : 인자로 들어온 리스트의 원소로 주어진 리스트는 한줄씩의 그림을 표현함
def compress_img(umcompressed_image):	umcompressed_img:숫자로 이미지를 표현한 리스트	리스트의 리스트: 압축된 리스트
def decompress_img(compressed_img):	compressed_img:리스트의 각 원소를 0과 1의 숫자의 반복으로 나타낸 리스트	리스트의 리스트: '*'은 1, ' '은 0으로 바뀌어 저장된 리스트의 리스트형
def print_lines(sequence): 위의 함수들의 호출 결과를 한 줄단위로 맞춰서 출력하는 함수	sequence : 이미지 혹은 문자열의 리스트	

◦ 함수 호출 부분

```
if __name__ == '__main__':  
    image1=[[' ', '*', '*', '*', ' '],  
            [' ', ' ', ' ', ' ', '*'],  
            [' ', '*', '*', '*', '*'],  
            ['*', ' ', ' ', ' ', '*'],  
            ['*', ' ', ' ', ' ', '*'],  
            [' ', '*', '*', '*', '*']]  
  
    image2=[[0,1,1,1,0],  
            [0,0,0,0,1],  
            [0,1,1,1,1],  
            [1,0,0,0,1],  
            [1,0,0,0,1],  
            [0,1,1,1,1]]  
  
    result1=str_to_img(image1)  
    print_lines(result1)  
    result2=img_to_str(image2)  
    print_lines(result2)  
    result3=compress_img(image2)  
    print_lines(result3)  
    result4=decompressed_img(result3)  
    print_lines(result4)
```

```
[0, 1, 1, 1, 0]  
[0, 0, 0, 0, 1]  
[0, 1, 1, 1, 1]  
[1, 0, 0, 0, 1]  
[1, 0, 0, 0, 1]  
[0, 1, 1, 1, 1]
```

\*\*\*

\*

\*\*\*\*

\* \*

\* \*

\*\*\*\*

[1, 3, 1]

[4, 1]

[1, 4]

[0, 1, 3, 1]

[0, 1, 3, 1]

[1, 4]

[0, 1, 1, 1, 0]

[0, 0, 0, 0, 1]

[0, 1, 1, 1, 1]

[1, 0, 0, 0, 1]

[1, 0, 0, 0, 1]

[0, 1, 1, 1, 1]