**파이썬 프로그래밍 기초 실습4**

2019년 12월 27일(금)

이름: ( ) 이메일: ( )

1. ***파일***

실습1) **– PPT 자료**

파일에 있는 각각의 단어 수 구하기

<test.txt>

|  |
| --- |
| first line  second line  third line |

|  |
| --- |
|  |
| line 3  second 1  third 1  first 1 |

(힌트) 딕셔너리 이용

실습2) – **PPT 자료**

파일명을 입력 받아, 해당 파일을 한 줄씩 읽어 파일의 내용을 모두 대문자로 출력하는 프로그램을 작성하시오.

|  |
| --- |
|  |
| Enter a file name: test.txt  FIRST LINE  SECOND LINE  THIRD LINE |

실습3) 아래의 실행예시처럼 리눅스 쉘에서 원본파일명(src.txt)과 사본파일(dst.txt)을 입력 받아, 복사하는 프로그램을 작성하시오.

|  |
| --- |
|  |
| $ python ex3.py src.txt dst.txt |

실습4) 아래의 score.txt를 읽어서 학생들의 성적을 처리하여 그 결과를 report.txt로 출력하는 프로그램을 작성하시오.

(총점) 중간고사 40%, 기말고사 60%

(학점기준)

총점이 90이상이면 A,

80이상~90미만이면 B

70이상~80미만이면 C

60이상~70미만이면 D

60미만이면 F

<score.txt> 예시

- 각 줄(line) 단위로 한 학생의 학번, 중간고사 점수, 기말고사 점수가 순서대로 기록되어 있다.

|  |
| --- |
| 201901 89 78  201902 76 84  201903 67 42  201904 66 88  201905 98 99 |

<report.txt> 예시

**학번, 중간고사, 기말고사, 평균(학점) 순으로 아래와 같이 출력해 주세요,**

|  |
| --- |
| 201901 89 78 82.4(B)  201902 76 84 80.8(B)  201903 67 42 52.0(F)  201904 66 88 79.2(C)  201905 98 99 98.6(A) |

<주의사항>

학생수를 예시의 학생수처럼 고정해서 구현하지 말 것

<제출 파일명>

ex4.py

1. ***Pythonic Code***

- 파이썬 특유의 문법을 활용하여 효율적으로 코드를 표현(파이썬 스타일의 코딩 기법)

예) 여러 문자열을 하나로 이을 때

|  |
| --- |
| >>> colors = ['red', 'green', 'blue']  >>> result = ''  >>> for word in colors: # 일반적으로  result += word  >>> result = ''.join(colors) |

1. **split(), join()**

**1일차 자료 참고**

1. **List Comprehensions**

* 리스트에 for 문을 사용하여 반복적으로 표현식을 실행해서 리스트 요소들을 정의하는 용법
* 기존 리스트를 사용하여 간단히 다른 리스트를 만드는 기법

|  |
| --- |
| **[표현식 for 요소 in 시퀀스자료형 [if 조건식]]** |

|  |
| --- |
| >>> ls = [] # 일반적으로~  >>> for i in range(1,6):  ls.append(i)  >>> ls  [1, 2, 3, 4, 5]  **>>> ls = [i for i in range(1,6)]**  >>> ls  [1, 2, 3, 4, 5]  >>> [ i \*\* 2 for i in range(1,6)]  >>> [ i for i in range(1,6) if i % 2 == 0]  **# 리스트가 2개**  >>> L1 = [1, 2, 3]  >>> L2 = [3, 4, 5]  >>> L1  [1, 2, 3]  >>> L2  [3, 4, 5]  >>> [x \* y for x in L1 for y in L2] |

1. **enumerate(), zip()**

**# enumerate()**

- 리스트의 모든 요소를 인덱스와 쌍으로 추출

|  |
| --- |
| >>> colors = ['red', 'green', 'blue']  >>> result = list(enumerate(colors))  >>> result  [(0, 'red'), (1, 'green'), (2, 'blue')]  >>> for idx, color in enumerate(colors):  print(idx, color) |

**# zip()**

* 2개 이상의 리스트를, 각 리스트의 같은 인덱스 원소끼리 묶은 튜플을 요소로 하는 리스트를 만들어 줌

|  |
| --- |
| >>> zip([1,2,3], [4,5,6]) # 리스트  [(1, 4), (2, 5), (3, 6)]  >>> zip([1,2,3], [4,5,6], ['a','b','c'])  [(1, 4, 'a'), (2, 5, 'b'), (3, 6, 'c')]  >>> zip((1,2,3), (4,5,6)) # 튜플  [(1, 4), (2, 5), (3, 6)]  >>> [sum(x) for x in zip((1,2,3), (4,5,6))] |

# enumerate(), zip()

|  |
| --- |
| for i, (a,b) in enumerate(zip(['a','b','c'], ['d','e','f'])):  print( i, a, b) |

실습1)

아래와 같이 주어진 2개의 리스트로 딕셔너리를 만들어 출력하시오. 단, 순서는 다를 수 있음

|  |
| --- |
| L1 = ['one', 'two', 'three', 'four']  L2 = [1, 2, 3, 4] |
| **{'one': 1, 'two': 2, 'three': 3, 'four': 4}** |

힌트) zip()