**AI · 빅데이터 · IoT 인재 양성 오프라인 심화과정**

**파이썬 프로그래밍 기초 실습 – 3일차 (1, 4, 5번)**

2017년 9월 6일(수)

이름: ( 박소선 (python3 기준) ) 이메일: ( parksosun1103@gmail.com )

1. ***반복문***

문제13) - **PPT 문제 (2일차)**

“done“을 입력할 때까지 사용자로부터 숫자를 입력 받아 리스트에 저장하고, “done“을 입력하면, 리스트의 평균, 최대값과 최소값을 출력하는 프로그램을 작성하시오.

(힌트) sum(), max(), min() 함수를 사용

Enter a number: 6

Enter a number: 2

Enter a number: 9

Enter a number: 3

Enter a number: 5

Enter a number: done

[6.0, 2.0, 9.0, 3.0, 5.0]

Average: 5.0

Maximum: 9.0

Minimum: 2.0

|  |
| --- |
| **result = []**  **while True**:  untilDone = input(**"Enter a number: "**)  **if** untilDone != **"done"**:  result.append(float(untilDone))  **else**:  **break**  average = sum(result) / len(result)  print(**"Average: "**, average)  print(**"Maximum: "**, max(result))  **print("Minimum: ", min(result))** |

1. ***Pythonic Code I***

- 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()**

* 동일한 원소를 가진 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 |

1. ***리스트 관련***
2. **스택, 큐**

**: 여러 데이터 항목을 일정한 순서로 나열한 자료구조**

**# 스택(Stack)**

* LIFO(Last Input First Output)

나중에 삽입(push)된 데이터가 처음(pop)으로 삭제

* 파이썬은 리스트를 사용하여 스택 구조를 활용

**push 기능: append(), pop 기능: pop()**

|  |
| --- |
| >>> stack = []  >>> stack.append(10)  >>> stack.append(20)  >>> print stack  [10, 20]  >>> stack.pop()  20  >>> stack.pop()  10  >>> print stack  [] |

실습) 입력된 글자를 역순으로 출력(스택으로 구현할 것)

|  |
| --- |
|  |
| Input a string: Hello  o l l e H |

**# 큐(Queue)**

* FIFO (First Input First Output)
* 삽입한 순서대로 먼저 삽입(enqueue)된 데이터가 가장 먼저 삭제(dequeue)
* 파이썬은 리스트를 사용하여 큐 구조를 활용

**enqueue기능: append(), dequeue기능:pop(0)**

|  |
| --- |
| >>> q = []  >>> q.append(10)  >>> q.append(20)  >>> print q  [10, 20]  >>> q.pop(0)  10  >>> q.pop(0)  20  >>> print stack  [] |

1. ***딕셔너리(dict)***

중괄호{ }로 묶여 있으며, key와 value의 쌍으로 이루어짐. d = {key1:value1, key2:value2}

실습1) for 문을 사용하여 딕셔너리 타입의 d의 모든 value를 출력해 보시오.

|  |
| --- |
| d = {'youn': 1, 'park': 2, 'kim': 10}  **for v in d.values():**  print(v, end=**" "**) |
| **1 2 10** |

실습2) 아래와 같이 주어진 2개의 리스트로 딕셔너리를 만들어 출력하시오. 단, 순서는 다를 수 있음

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

힌트) zip()

실습3) 아래와 같은 딕셔너리가 있다.

|  |
| --- |
| {'one':1, 'two':2, 'three':3, 'four':4, 'five':5} |

key의 알파벳을 기준으로 순서대로 아래와 같이 (key, value)로 출력하시오.

|  |
| --- |
| **origin\_d = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5}**  list\_d = list(origin\_d)  alpha\_d = sorted(list\_d)  *#for loop - dictionary[key] = value*  **for** ele **in** alpha\_d:  print(ele, origin\_d[ele]) |
| **five 5**  **four 4**  **one 1**  **three 3**  **two 2** |

실습5) – **PPT 문제**

어떤 문장을 입력 받으면 해당 문장에서 각 알파벳이 몇 개씩 나오는지 저장하는 딕셔너리를 만든 후, 아래와 같이 출력하시오.

|  |
| --- |
| **sentence = input("Enter a sentence: ")**  letter\_dict = {}  **for** w **in** sentence:  **if** (w **in** letter\_dict):  letter\_dict[w] += 1  **else**:  letter\_dict[w] = 1 print(letter\_dict) |
| Enter a sentence: Python is fun!  {'!': 1, ' ': 2, 'f': 1, 'i': 1, 'h': 1, 'o': 1, 'n': 2, 'P': 1, 's': 1, 'u': 1, 't': 1, 'y': 1} |

1. ***파일***

실습1) **– PPT 자료**

파일에 있는 각각의 단어 수 구하기

<test.txt>

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

|  |
| --- |
| **f = open("test.txt", "r")**  *#word: frequency*  data = f.read() *#data is string type* data\_ls = data.split() print(data\_ls) diction = {}  **for** word **in** data\_ls:  **if**(word **in** diction):  diction[word] += 1  **else**:  diction[word] = 1 print(diction)  **for** key **in** diction:  print(key, diction[key])  f.close() |
| 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 작성하세요.

|  |
| --- |
| **import os**  filename = input(**"Enter a file name: "**)  **if** os.path.exists(filename):  f2 = open(filename, **"r"**)  data2 = f2.read()  print(data2.upper())  **else**: print(**"That file does not exist."**)  f2.close() |
| Enter a file name: test.txt  FIRST LINE  SECOND LINE  THIRD LINE |

실습3) **– PPT 자료**

파일의 라인 수, 단어 수를 구하여 출력하는 프로그램을 작성하시오. (단, 단어는 공백으로 구분된 문자열을 의미한다고 가정)

<test.txt>

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

|  |
| --- |
| **import sys**  **args = sys.argv[1:]** f = open(args[0],**"r"**)  data = f.read() num\_words = len(data.split())  #go back to first line of file f.seek(0) num\_lines = 0 **for** line **in** f:  num\_lines += 1  print(num\_lines, num\_words)  f.close() |
| $ python ex3.py test.txt  **3 6** |

실습4) 아래의 실행예시처럼 리눅스 쉘에서 원본파일명(src.txt)과 사본파일(dst.txt)을 입력 받아, 복사하는 프로그램을 작성하시오.

|  |
| --- |
| **import sys**  **inputs = sys.argv[1:]**  *#argv[0] = ex4.py #argv[1] = inputs[0] = src.txt #argv[2] = inputs[1] = dst.txt*  src = open(inputs[0], **"r"**) copy = src.read()  dst = open(inputs[1], **"w"**) dst.write(copy)  src.close() dst.close() |
| $ python ex4.py src.txt dst.txt |

실습5) 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.

|  |
| --- |
| 201701 89 78  201702 76 84  201703 67 42  201704 66 88  201705 98 99 |

<report.txt> 예시

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

|  |
| --- |
| 201701 89 78 82.4(B)  201702 76 84 80.8(B)  201703 67 42 52.0(F)  201704 66 88 79.2(C)  201705 98 99 98.6(A) |

<주의사항>

학생수를 예시의 학생수처럼 고정해서 구현하지 말 것

<제출 파일명>

Day3\_홍길동.py