00_Python 복습

Python 복습

- 복습 테스트 문제

다음 문제를 작성하시오.

1) range()를 사용하여, 1~100의 수 중, 7의 배수를 리스트 a에 담으시오.
2) a에서 인덱스 4~7까지 조회하시오.
3) a의 끝에서 세번째 값을 조회하시오.
4) a에 데이터 갯수(길이)를 구하시오.
5) a 값들의 합과 평균을 구하시오.
6) [심화] 반복문을 이용하여 100이하의 소수(prime number)를 리스트에 담아봅시다.
7) [심화] 6번의 코드를 참조하여, 소수 추출기 함수를 만들어봅시다.
8) 아래 값으로 딕셔너리 d를 생성해봅시다.
9) d에서 [1,2,3]을 출력해봅시다.
10) d에서 23을 출력해봅시다.
11) d에서 5를 출력해봅시다.
12) d에 'newKey', value(1,2)를 추가하고 출력해봅시다.

Python 복습

- 기초 문제

1) range()를 사용하여, 1~100의 수 중, 7의 배수를 리스트 a에 담으시오.

```
a = list(range(7,101,7))
```

2) a에서 인덱스 4~7까지 조회하시오.

```
a[4:8]
```

3) a의 끝에서 세번째 값을 조회하시오.

```
a[-3]
```

Python 복습

- 기초 문제

4) a에 데이터 갯수(길이)를 구하시오.

```
len(a)
```

5) a 값들의 합과 평균을 구하시오.

```
print("a의 합 :",sum(a))
print("a의 평균 :",sum(a)/len(a))
```

Python 복습

- 기초 문제

6) [심화] 반복문을 이용하여 100이하의 소수(prime number)를 리스트에 담아봅시다.

- 소수(prime number) : 1과 자기자신으로만 나누어 지는 수

```
prime_number = []
for i in range(2,101):
    divide = 0
    for j in range(1, i+1):
        if i % j == 0:
            divide += 1
    if divide == 2:
        prime_number.append(i)

print(prime_number)
```

Python 복습

- 기초 문제

6) [심화] 반복문을 이용하여 100이하의 소수(prime number)를 리스트에 담아봅시다.

- 제곱급을 이용한 또 다른 방법

```
import math

prime_number = []
for i in range(2,101):
    divide = 0
    for j in range(2,int(math.sqrt(i))+1):
        if i % j == 0:
            divide += 1
    if divide == 0:
        prime_number.append(i)

print(prime_number)
```

Python 복습

- 기초 문제

7) [심화] 6번의 코드를 참조하여, 소수 추출기 함수를 만들어봅시다.

- 입력변수 : 정수 n
- 처리 : n 이하의 소수를 리스트에 저장
- 출력변수 : 소수 리스트
- [주의] 입력변수 n은 2이상의 정수여야 함.(입력값 체크 부분 포함)

```
def prime1(n):
    prime_number = []
    for i in range(2,n+1):
        divide = 0
        for j in range(1, i+1):
            if i % j == 0:
                divide += 1
        if divide == 2:
            prime_number.append(i)
    return prime_number
print(prime1(100))
```

Python 복습

- 기초 문제

제곱근을 이용한 함수 만들기

```
def prime2(n):
    prime_number = []
    for i in range(2,n+1):
        divide = 0
        for j in range(2,int(math.sqrt(i))+1):
            if i % j == 0:
                divide += 1
        if divide == 0:
            prime_number.append(i)
    return prime_number

print(prime2(100))
```

Python 복습

- 기초 문제

함수의 속도비교

```
import time

start = time.time()
prime1(10000)
print("prime1's time :",time.time() - start)

start = time.time()
prime2(10000)
print("prime2's time :",time.time() - start)
```

- 기초 문제

8) 아래 값으로 딕셔너리 d를 생성해봅시다.

key	value
v1	[1,2,3]
v2	{'a':23, 'b':[4,5]}

```
d = {'v1':[1,2,3],'v2':{'a':23,'b':[4,5]}}
```

9) d에서 [1,2,3]을 출력해봅시다.

```
d['v1']
```

Python 복습

- 기초 문제

10) d에서 23을 출력해봅시다.

```
d['v2']['a']
```

11) d에서 5를 출력해봅시다.

```
d['v2']['b'][1]
```

12) d에 'newKey', value(1,2)를 추가하고 출력해봅시다.

```
d['newKey'] = (1,2)
```

Python 복습

- 기초 문제

13) [심화] 반복문과 딕셔너리를 사용하여 주어진 단어에서 가장 많이 사용된 알파벳을 출력해봅시다.

- 주어진 단어 : "Big Data Analysis"
- 대문자, 소문자는 구분하지 않습니다.
- 가장 많이 사용된 알파벳을 2개 이상일 경우, 전부 출력해줍니다.

```
S = 'Big Data Analysis'
S = S.lower()
alphabet = {}
```

Python 복습

- 기초 문제

```
for i in S:
    if i not in alphabet:
        alphabet[i] = 1
    else:
        alphabet[i] += 1

alpha_list = []
for k,v in alphabet.items():
    if v == max(alphabet.values()):
        alpha_list.append(k)

print(alpha_list)
```