

2주차

2020.11.2 – 11. 8

CodeUp1286

K개의 줄에 입력 받은 값을 하나의 리스트로 만들기

```
N = int(input())  
Nums = [int(input()) for _ in range(N)]
```

N = 5개 = 5

CodeUp1281

문자열 이어붙이기 : '+' 기호 사용

```
a,b = map(int, input().split())
sum = 0

s = ''

for i in range(a,b+1):
    if i%2 != 0:
        if i==a:
            sum = sum+i
            s = s+'{}'.format(i)
        else :
            sum = sum+i
            s = s+'+{}'.format(i)
    else :
        sum = sum-i
        s = s+'-{}'.format(i)

result = '{}={}'.format(s,sum)
print(result)
```

Global

```
A = 0
```

```
[함수] def func() :  
    global A(A값 공유함)  
    A += 1
```

```
for i in range(10) :  
    func()
```

```
print(a)
```

속도 빠른 입력

1줄에 입력받기(int형 리스트)

```
import sys
Data = map(int, sys.stdin.readline().split())
```

1줄에 입력받기(문자열, 띄어쓰기 구분)

```
Import sys
Data = sys.stdin.readline().split()
-> ['a', 'c\\n']
Data = sys.stdin.readline().rstrip('\\n').split('구분문자')
-> ['a','c']
```

K개의 줄에 입력 받은 값을 하나의 리스트로 만들기(속도빠른방법)

```
Import sys
```

```
Data = [int(sys.stdin.readline()) for i in range(n)]
```

CodeUp 1295

[대문자, 소문자]

name = 'abC'

- 1) 모두대문자변환 : name.upper()
- 2) 모두소문자변환 : name.lower()
- 3) 대문자확인 : name.isupper()
- 4) 소문자확인 : name.islower()
- 5) 문자열 바꾸기 : name.replace('바꿀문자','새문자')
 - 1) 바꿀 문자가 여러 개 있어도 다 같이 바꿈
- 6) 문자열 왼쪽에 0 채우기 : name.zfill(0개수)
- 7) 문자열 위치 찾기 : name.find('찾을 문자열')
 - 1) 찾을 문자열이 여러 개인 경우 왼쪽에서부터 가장 빨리 나온 위치 반환
- 8) 문자열 개수 세기 : name.count('찾을 문자열')

- 대문자, 소문자 아니면 Pass함

CodeUp 1418

특정 문자(target)가 몇 번째에 나오는지 모두 출력

```
import sys
a = sys.stdin.readline()

target = 't'
index = -1

while True:
    # 입력이 0번부터 시작하도록 1을 더해줌
    index = a.find(target, index+1)
    if index == -1:
        break
    # 위치가 1부터 시작하므로 결과에 1을 더해줌
    print(index+1, end=' ')
```

CodeUp 2721

```
names = [input().split() for _ in range(3)]
```

```
print(names[0])  
print(names[1])  
print(names[2])
```

```
print(names[0][0])  
print(names[1][0])  
print(names[2][0])
```

```
print(names[0][0][0])  
print(names[1][0][0])  
print(names[2][0][0])
```

```
if names[0][0][0] == names[2][0][len(names[2][0])-1]:  
    print('good')  
else:  
    print('bad')
```

```
turtle  
error  
robot  
['turtle']  
['error']  
['robot']  
turtle  
error  
robot  
t  
e  
r
```

입력 여러 줄 받
은 문자열 배열
요소 접근 방법

그리디 알고리즘1 - 거스름돈

현재 상황에서 지금 당장 좋은 것만 고르는 방법
거슬러야 할 동전의 최소개수?(500,100,50,10 동전)
거스름돈 : N, N은 항상 10의 배수

```
N = int(input())

coins = [500, 100, 50, 10]
count = 0

for coin in coins :
    # 500부터 나눈 몫
    count = count + N//coin
    # 500부터 나눈 나머지
    N = N%coin

print(count)
```

[아이디어]
가장 큰 수부터 나누면서
작은 수 방향으로 진행,
입력 값 계속 변화시켜줌
(화폐 단위 무작위인 경우
다른 방법 적용필요)

그리디 알고리즘2 - 큰수의법칙

CodeUp 1675

[문자열 접근방법]

Name = 'abc'

for i in name :

 print(i) -> i는 문자 하나씩 접근함

*참고

ord('a') = 97, ord('z') = 122

ord('A') = 65, ord('Z') = 90

```
import sys
name = sys.stdin.readline()

s = ''
for i in name :
    if ord(i)>99 :
        s = s+chr(ord(i)-3)
    elif ord(i)>=97 and ord(i)<=99 :
        s = s+chr(ord(i)+23)
    else :
        s = s+' '
print(s)
```