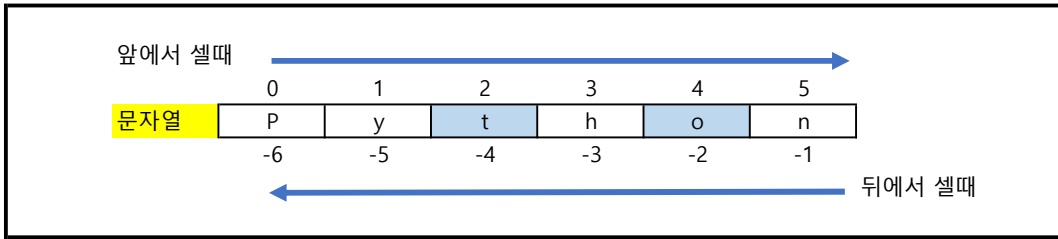


# 9. 문자열 가공

## 열거된 문자, 문자열

- 문자들을 일렬로 나열시켜 순서에 맞게 index 를 부여한 형태
- . 문자열을 구성하는 개별 문자를 읽을때는 [ ] 괄호 와 문자 의 위치 index 를 적는다. (리스트 와 같은 형식)
- > 문자열의 index



- > 문자열 index 로 문자열 을 출력하는 예시
- \* 각각 t, o 를 지칭하여 나타낸다

```
Chap01_intro > 08.String[문자열 분리].py > ...
1 sValue = 'Python'
2 print(sValue[2])
3 print(sValue[-2])
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Microsoft Windows [Version 10.0.22621.2134]  
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/Python.exe -i

t  
o

- > 반복문을 이용한 문자열 의 문자 추출
- \* Python 문자열의 수 개수 만큼 반복하여 각각 문자를 추출 후 출력 한다.

```
103 sValue = 'Python'
104 for char in sValue :
105     print(char,end=' ')
106
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

erD\.vscode\extensions\ms-python.python-2023.16.0\pythonFiles\0.MyFirstPython\0.Source\Python\09.String[문자열 관리\_1].py" Python

- > 문자열의 길이 를 지정하여 문자 를 추출 하는 방법
- \* 문자열의 길이를 구할 수 있는 len() 내장 함수를 이용하여 구한다.

```
114 sValue = 'Python'
115 for index in range(2,len(sValue)) :
116     print(sValue[index],end=' ')
117
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

thon

- > 115행 : 0 부터 sValue 의 최종 길이 5 까지 를 반복
- \* index : 0 ~ 5 의 값을 가지며 sValue 의 index 가 된다.
- > 116행 : 추출 한 index 의 문자열 의 문자를 하나씩 출력한다.

> 문자열의 index 로 문자를 변경 할 수 없다.

\* 문자열은 변경할수 없는 데이터 형식 (Immutable) 로 되어있어 한번 초기화 된 문자열의 데이터 를 수정 하는것은 불가능 하다.

```
Chap01_intro > 08.String[문자열 분리].py > ...
1 sValue = 'Python'
2 sValue[1] = 'p'
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe "d:
Traceback (most recent call last):
  File "d:\Python\Chap01_intro\08.String[문자열 분리].py", line 2, in <module>
    sValue[1] = 'p'
    ~~~~~^^^
TypeError: 'str' object does not support item assignment
```

> Immutable 형식의 데이터 할당 프로세스

\* 문자열의 데이터를 갱신하여 새로이 배치하는 작업을 하는것 보다 새로운 객체를 만들어 처리하는것이 더 효율적인 프로세스 처리 라고 판단.  
예 ) 한번 누르면 일정시간동안 자동으로 나오는 수도꼭지. 수동으로 잠그는 방식보다 낭비되는 물이 더 손실 비용이 절감 되는 케이스

```
Chap01_intro > 08.String[문자열 분리].py > ...
1 sValue = '안녕하세요?'
2 sValue += ' 반갑습니다.'
3 sValue += ' Python 을 배워봅시다'
4 print(sValue)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe "d:
안녕하세요? 반갑습니다. Python 을 배워봅시다
```

- > 1행 : sValue 에 '안녕하세요?' 초기화
- > 2행 : 기존에 있던 sValue 의 데이터는 메모리에 두고  
새로운 sValue 가 메모리에 등록 되면서  
'안녕하세요 ? 반갑습니다.' 초기화
- > 3행 : 기존에 있던 sValue 는 두고  
새로운 sValue 가 메모리에 등록 되면서  
'안녕하세요 ? 반갑습니다. Python 을 배워봅시다.' 초기화

sValue	안녕하세요 ? 반갑습니다. Python 을 배워봅시다.
sValue(X)	안녕하세요? 반갑습니다. (가비지 컬렉터가 수거)
sValue(X)	안녕하세요? (가비지 컬렉터가 수거)

## 문자열 을 합하여 가공할때는 StringIO

- StringIO 모듈의 write() 기능을 이용하면 메모리의 손실 없이 특정 문자열에 문자를 효율적으로 추가 할 수 있다.

. 사용법은 다소 복잡하지만 대용량의 문자를 가공 할때는 중요한 역할을 한다.

```
119 from io import StringIO
120 buffer = StringIO()
121 buffer.write('안녕하세요?')
122 buffer.write(' 반갑습니다.')
123 buffer.write(' Python을 배워봅시다')
124 result = buffer.getvalue()
125 buffer.close()
126 print(result)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

/..../debugpy/launcher 64824 -- "D:\MyDoc\11.EDUCATION\1. Master Source\15.Python
안녕하세요? 반갑습니다. Python을 배워봅시다
```

## 문자열 슬라이스

- 문자열 index 를 기입 하는 ' [ ] ' 에 범위를 지정

	0	1	2	3	4	5	6	7	8
문자열	P	y	t	h	o	n	.	p	y
	-9	-8	-7	-6	-5	-4	-3	-2	-1

. 문자열에 범위 를 지정하여 나타내는 예시

```
Chap01_intro > 08.String[문자열 분리].py > ...
1 sValue = 'Python.py'
2 print(sValue[2:5])
3 print(sValue[3:])
4 print(sValue[:4])
5 print(sValue[2:-3])
6 print(sValue[-3:])
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Pyt
thon
hon.py
Pyth
thon
.py
```

- > 2행 : index 2 ~ index 5-1 의 문자열 출력 'tho'
- > 3행 : index 3 ~ 모든 문자열 출력 'hon.py'
- > 4행 : index 0 부터 index 4-1 의 문자열 출력 'Pyth'
- > 5행 : index 2 부터 index -3 -1 까지의 문자열 출력 'thon'
- > 6행 : index -3 위치로부터 마지막 까지의 문자열 출력 '.py'
- \* 파일의 확장자 등을 추출할때 유용하게 사용된다.

. 건너 뛰면서 문자를 추출하는 기능 ::

```
Chap01_intro > 08.String[문자열 분리].py > ...
1 sValue = 'Python.py'
2 print(sValue[::2])
3 print(sValue[::-2])
4 print(sValue[2::2])
5 print(sValue[-2::-1])
6
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python
Pto,y
y.otP
to.y
p.nohtyP
```

- > 2행 : 문자열의 처음부터 2칸씩 건너뛰면서 출력
- > 3행 : 문자열의 끝부터 2칸씩 건너 들어오면서 출력
- > 4행 : index 2 의 위치부터 2칸씩 건너뛰면서 출력
- > 5행 : index -2 의 위치 부터 1칸씩 들어오면서 출력
- \* :: -1 은 문자열 반전 기능으로 활용 할 수 있다.

. 범위와 Step 의 동시 지정

```
87 sValue = 'Python.py'
88 print(sValue[2:8:2]) # 2 ~ 7 까지 2칸씩 건너뛰며 출력 to.
89

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

D:\MyDoc\11.EDUCATION\1. Master Source\15.Python\0.MyFirstPython\0.Source\Python> d: && cd "d:\MyDoc\11.EDUCATION\1. Mast
cmd /C "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe c:\Users\MasterD\.vscode\extensions\ms-python
/./.../debugpy\launcher 55162 -- "D:\MyDoc\11.EDUCATION\1. Master Source\15.Python\0.MyFirstPython\0.Source\Python\09.Str
to.
```

## 문자열 메서드

- 메서드는 클래스에 소속되어 있는 함수이며 객체에 대해 특화된 작업을 수행한다.
  - . 클래스 : 프로그램 로직을 수행하는 기본 단위이며 변수 또는 클래스 함수(메서드) 등을 구현해두고 필요시 사용 할 수 있도록 만든 모듈(수행 단위) 별 프로그램의 모음, 계획서, 설계서
  - . 객체 : 클래스를 사용 하기 위하여 선언(인스턴스) 한 메모리 상에 등록 되어있는 상태의 클래스
- 문자열 메서드
  - . 문자열을 다룰 수 있도록 만들어둔 클래스에서 제공하는 문자열 클래스의 함수

index	0	1	2	3	4	5	6						
문자열	P	y	t	h	o	n							
index	7	8	9	10	11	12	13	14	15	16	17		
문자열	P	r	o	g	r	a	m	m	i	n	g		

```

Chap01_intro > 08.String[문자열 분리].py > ...
1 sValue = 'Python Programming'
2 print(len(sValue))
3 print(sValue.find('o'))
4 print(sValue.rfind('og'))
5 print(sValue.index('m'))
6 print(sValue.count('n'))
7
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python31
18
4
9
13
2

```

- >1행 : 문자열을 초기화 한 sValue 변수는 문자열 클래스의 객체가 되고 sValue 객체에 'Python Programming' 을 초기화 한다.
- >2행 : len() 은 내장 함수로서 문자열의 길이 를 나타낸다
- >3행 : 문자열에 'o' 가 포함 되어 있는지 확인후 index 를 반환한다
  - \* index 0 부터 가장 첫 'o' 를 찾았을 경우 index 반환
  - 없을 경우 -1 을 반환한다.
- >4행 : index 17 부터 들어오면서 첫 'og' 의 위치를 반환
- >5행 : index 0 부터 'm' 의 문자열이 존재하는 위치를 반환
  - \* 없을 경우 오류를 발생한다.
- >6행 : 'n' 문자의 개수 반환

- . 문자열을 인수로 받는 클래스 함수(메서드) 로서 int (index) 를 반환한다.
  - > 대소문자도 구별하여 체크한다.

```

145 indexValue = sValue.count('P')
146 print(indexValue, type(indexValue))
147 indexValue = sValue.count('p')
148 print(indexValue, type(indexValue))
149
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

D:\MyDoc\11.EDUCATION\1. Master Source\15.Python\0.MyFirstPython\0.Source\Python\
scode\extensions\ms-python.python-2023.16.0\pythonFiles\lib\python\debugpy\adapte
hon\0.Source\Python\09.String[문자열 관리_1].py" "
2 <class 'int'>
0 <class 'int'>

```



## 조사

- 문자열이 존재 하는지 확인 할수 있는 기능  
. In 또는 not in 으로 확인 할 수 있다.

```
Chap01_intro > 08.String[문자열 분리].py > ...
1 sValue = 'Python Programming'
2
3 if 'P' in sValue :
4     print('대문자 P 문자는 존재 합니다.')
5
6 if 'p' not in sValue :
7     print('소문자 p 문자는 존재 합니다.')
8
9 if 'prox' not in sValue :
10    print('prox 문자열은 존재 하지 않습니다.')
11
12
13 print('m' in sValue)
14 print('python' not in sValue)
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe
대문자 P 문자는 존재 합니다.
소문자 p 문자는 존재 합니다.
prox 문자열은 존재 하지 않습니다.
True
True
```

\* 문자열 찾기 기능은 대 / 소문자 를 구분한다.

>3행 : 대문자 P 가 있는지 확인하는 분기문

>6행 : 소문자 p 가 있는지 확인하는 분기문

>9행 : 'prox' 문자열 이 존재 하지 않는지 확인하는 분기문

>13행 : 'm' 문자 가 있으면 True

>14행 : 'python' 문자열이 없으면 True

\* 대 소문자 구분

- 해당 문자열로 시작하는지 , 종료 하는지 확인하는 기능  
. startswith() , endswith()

```
Chap01_intro > 08.String[문자열 분리].py > ...
1 sValue = 'Python_Programming.jpg'
2
3 if sValue.startswith('Python'):
4     print('Python 관련 데이터 입니다.')
5 if sValue.endswith('.jpg') :
6     print('jpg 형식 그림파일 입니다.')
7
8 print(sValue.startswith('python'))
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe "d:
Python 관련 데이터 입니다.
jpg 형식 그림파일 입니다.
False
```

>3행 : Python 으로 시작 하는지 확인

>5행 : .jpg 로 끝나는지 확인

>8행 : python 으로 시작시 True

\* 대소문자 구별

. 파이선 문자열 메서드

함수	설명
isalpha	모든 문자 가 알파벳인지 조사한다
islower	모든 문자 가 소문자인지 조사한다
isupper	모든 문자 가 대문자인지 조사한다
isspace	모든 문자 가 공백인지 조사한다
isalnum	모든 문자 가 알파벳 또는 숫자 인지 조사한다
isdecimal	모든 문자 가 숫자인지 조사한다
isdigit	모든 문자 가 숫자인지 조사한다
isnumeric	모든 문자 가 숫자인지 조사한다
isidentifier	명칭으로 쓸 수 있는 문자로만 구성되어있는지 조사한다
isprintable	인쇄 가능한 문자로만 구성되어 있는지 조사한다

> isidentifier  
유효한 식별자 : 영문(A~z), 숫자(0~9), \_ 로만 구성.  
유효하지 않은 식별자 : 숫자로 시작, 공백이나 특수문자 포함.

변경

- 문자열의 특정 문자들을 변경 하는 기능
- . 문자열 변경 기능을 사용한 예제.

```

Chap01_intro > 08.string[문자열변경].py > ...
1 sValue = 'Visual Code 를 이용한 Python course'
2 print(sValue.lower())
3 print(sValue.upper())
4 print(sValue)
5
6 print(sValue.swapcase())
7 print(sValue.capitalize())
8 print(sValue.title())
9
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python> cmd /C "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python,
thonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher 50964 -- d:\Python\Chap
visual code 를 이용한 python course
VISUAL CODE 를 이용한 PYTHON COURSE
Visual Code 를 이용한 Python course
vISUAL cODE 를 이용한 pYTHON COURSE
Visual code 를 이용한 python course
Visual Code 를 이용한 Python Course

```

>2행 : 모든 알파벳을 소문자로 변경  
>3행 : 모든 알파벳을 대문자로 변경  
>4행 : sValue 값 출력  
\* 문자열은 Immuteable 형식 이므로 문자열 자체를 변경하지는 않는다.  
>6행 : 대문자 와 소문자를 변경하여 출력  
>7행 : 문장의 가장 첫 문자를 대문자로 변경  
>8행 : 문단의 가장 첫 문자를 대문자로 변경

- . 대 소문자 구별 없이 일괄 비교 할 수 있도록 하는 로직
- > 모든 영단어를 대문자 / 소문자 로 변경하여 비교

```

99 ~while True :
100     userid = input('ID 를 입력하세요 : ')
101 ~     if userid.lower() == 'python' :
102         print(userid + ' 님 반갑습니다.')
103         break
104 ~     if userid.upper() == 'ADMIN' :
105         print(userid + ' 님 반갑습니다.')
106         break
107
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ID 를 입력하세요 : python
python 님 반갑습니다.

D:\MyDoc\11.EDUCATION\1. Master Source\15.Python\0.MyFirstPython\0.Source\Python> d: && cd "
cmd /C "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe c:\Users\MasterD
/../../debugpy\launcher 53573 -- "D:\MyDoc\11.EDUCATION\1. Master Source\15.Python\0.MyFirst
ID 를 입력하세요 : admin
admin 님 반갑습니다.

```

> 사용자가 입력한 단어 를 대/소문자 로 변경하여 비교하면  
사용자는 대/소 문자를 신경쓰지 않고 입력 할 수 있다.

- 문자열의 공백 제거

. 문자열에 불필요하게 포함 된 공백을 제거하는 기능

```
Chap01_intro > 08.string[문자열변경].py > ...
1 sValue = ' Python '
2 print(sValue , '좋아염')
3 print(sValue.lstrip() , '좋아염')
4 print(sValue.rstrip() , '좋아염')
5 print(sValue.strip() , '좋아염')
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]  
(c) Microsoft Corporation. All rights reserved.

D:\Python> cmd /C "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:\Python\Chap01\_intro\08.string[문자열변경].py"

Python 좋아염  
Python 좋아염  
Python 좋아염  
Python 좋아염

- > 3행 : 문장 왼쪽의 모든 공백을 제거한다.
- > 3행 : 문장 오른쪽의 모든 공백을 제거 한다.
- > 3행 :문장 양끝의 모든 공백 을 제거 한다.

분할

- 문자열을 특정 문자 기준으로 나눌때 사용하는 기능

. split([나누고싶은 문자])

> split() 을 할 경우 공백을 기준으로 문자열을 분할한다.

```
Chap01_intro > 08.string[문자열변경].py > ...
1 sProgramming = 'Python Java C# C JavaScript'
2 print(sProgramming.split())
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]  
(c) Microsoft Corporation. All rights reserved.

D:\Python> cmd /C "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:\Python\Chap01\_intro\08.string[문자열변경].py"

['Python', 'Java', 'C#', 'C', 'JavaScript']

> 특정 문자 'w' 를 기준으로 분할하여 새로운 문자열 로 구성후 표현

```
Chap01_intro > 08.string[문자열변경].py > ...
1 Databases = 'MySQL\\MSSql\\Oracle\\Maria\\Mongo'
2 Database = Databases.split('\\')
3 print(Database)
4 for DB in Database :
5     print(DB,end=' database! ')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]  
(c) Microsoft Corporation. All rights reserved.

D:\Python> cmd /C "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:\Python\Chap01\_intro\08.string[문자열변경].py"

['MySQL', 'MSSql', 'Oracle', 'Maria', 'Mongo']  
MySQL database! MSSql database! Oracle database! Maria database! Mongo database!

- 문자열을 행을 기준으로 나누고자 할때 사용

. splitlines()

> 개행 지점을 기준으로 문자를 나누어 관리 할 수 있다.

```
Chap01_intro > 08.string[문자열변경].py > ...
1 sMessage = '안녕하세요\n반갑습니다\n쉽고 재미있는\nPython 입니다.'
2 sValue = sMessage.splitlines()
3 for line in sValue : print(line)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]  
(c) Microsoft Corporation. All rights reserved.

D:\Python> cmd /C "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:\Python\Chap01\_intro\08.string[문자열변경].py"

안녕하세요  
반갑습니다  
쉽고 재미있는  
Python 입니다.



- 문자열의 조합 join
  - . 특정 문자열 을 삽입하면서 나누어져 있는 문자열을 조합 한다.

```
Chap01_intro > 08.string[문자열변경].py > ...
1 Programmings = 'Python Java C# C JavaScript'
2 Programming = Programmings.split()
3 print(' Language '.join(Programming))
4
```

Microsoft Windows [Version 10.0.22621.2134]  
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/P  
Python Language Java Language C# Language C Language JavaScript

## 대체

- 특정 문자열을 원하는 문자열 로 변경
  - . replace()

```
134 # # 특정 문자열을 원하는 문자열 로 변경 replace()
135
136 sValue = '안녕하세요 ㅇㅇㅇ 입니다 '
137 print(sValue)
138 print(sValue.replace('ㅇㅇㅇ','박효신'))
```

Microsoft Windows [Version 10.0.22621.2283]  
(c) Microsoft Corporation. All rights reserved.

D:\MyDoc\11. EDUCATION\1. Master Source\15. Python\0. MyFirstPython\0. Source\Python> cmd /c "C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/Python/08.string[문자열변경\_2].py"
안녕하세요 ㅇㅇㅇ 입니다
안녕하세요 박효신 입니다

- . 공백 추가를 이용한 정렬

```
135 # # 문자열에 공백을 추가하여 정렬 하기
136 message = 'Hello Python' # 12글자
137 message2 = 'VsCode' # 6글자
138 print(message.ljust(30),':') # 오른쪽 18 공백
139 print(message2.ljust(30),':') # 오른쪽 24 공백
140 print(message.rjust(30)) # 왼쪽 18 공백
141 print(message2.rjust(30)) # 왼쪽 24 공백
142 print(message.center(30)) # 좌우 9 공백
143 print(message2.center(30)) # 좌우 12 공백
```

```
Hello Python :
VsCode :
Hello Python
VsCode
Hello Python
VsCode
```

< 결과 >

## 실습

안녕하세요 2023 안동대학교 스마트팩토리 S/W 개발 교육과정을 이수하게 된 000 입니다. 즐겁고 보람찬 DIGITALTRANING 교육이 되었으면 합니다.

1. 000 (대문자 o) 본인이름 변경 후 출력
2. S/W 글자위치 찾아서 출력
3. 시작 단어와 마지막 단어 1자씩 출력
4. 문자의 앞 뒤에 '-KDT-' 입력
5. 'DIGITALTRANING' 글자만 소문자로 변경한 후 출력
6. 모든 위치의 공백 없앰후 출력
7. . (문단) 을 기준으로 행을 나눠서 출력
8. 7에서 나눈 문단을 좌우 공백을 일정하게 둔 가운데 정렬로 표현

```
안녕하세요 2023 안동대학교 스마트팩토리 S/W 개발 교육과정을 이수하게 된 000 입니다. 즐겁고 보람찬 DIGITALTRANING 교육이 되었으면 합니다.

1. 000 (대문자 o) 본인이름 변경 후 출력
안녕하세요 2023 안동대학교 스마트팩토리 S/W 개발 교육과정을 이수하게 된 김범수 입니다. 즐겁고 보람찬 DIGITALTRANING 교육이 되었으면 합니다.

2. S/W 글자위치 찾아서 출력
24

3. 시작 단어와 마지막 단어 1자씩 출력
안 .

4. 문자의 앞 뒤에 '-KDT-' 입력
-KDT- 안녕하세요 2023 안동대학교 스마트팩토리 S/W 개발 교육과정을 이수하게 된 000 입니다. 즐겁고 보람찬 DIGITALTRANING 교육이 되었으면 합니다. -KDT-

'5. DIGITALTRANING' 글자만 소문자로 변경한 후 출력
안녕하세요 2023 안동대학교 스마트팩토리 S/W 개발 교육과정을 이수하게 된 000 입니다. 즐겁고 보람찬 digitaltraning 교육이 되었으면 합니다.

6. 모든 위치의 공백 없앰후 출력
안녕하세요2023안동대학교스마트팩토리S/W개발교육과정을이수하게된000입니다.즐겁고보람찬DIGITALTRANING교육이되었으면합니다.

7. 문단을 기준으로 행을 나눠서 출력
안녕하세요 2023 안동대학교 스마트팩토리 S/W 개발 교육과정을 이수하게 된 000 입니다.
즐겁고 보람찬 DIGITALTRANING 교육이 되었으면 합니다.

8. 7에서 나눈 문단을 좌우 공백을 일정하게 둔 가운데 정렬로 표현
안녕하세요 2023 안동대학교 스마트팩토리 S/W 개발 교육과정을 이수하게 된 000 입니다.
즐겁고 보람찬 DIGITALTRANING 교육이 되었으면 합니다.
```