



컴퓨팅 사고력을 키우는 SW 교육

# 파이썬

# Chapter 05

## 문자열 (String)

01 문자열 기본

02 문자열 함수



# Section 01 문자열 기본

# 문자열 기본 (1)

## ■ 리스트와 비슷한 문자열

```
aa = [10, 20, 30, 40, 50]  
aa[0]  
aa[1:3]  
aa[3:]
```

출력 결과

```
10  
[20, 30]  
[40, 50]
```

```
ss = "파이썬만세"  
ss[0]  
ss[1:3]  
ss[3:]
```

출력 결과

```
'파'  
'이썬'  
'만세'
```

## 문자열 기본 (2)

- 연결할 경우 더하기(+) 사용, 곱하기(\*)는 문자열을 반복

```
ss = '파이썬' + '만세'  
ss  
ss = '파이썬' * 3  
ss
```

출력 결과

```
'파이썬만세'  
'파이썬파이썬파이썬'
```

- 문자열의 길이 len( ) 함수

```
ss = '파이썬abcd'  
len(ss)
```

출력 결과

```
7
```

# 문자열 기본 (3)

- for문을 사용하여 모든 글자 뒤에 \$ 붙이기

## 소스코드 8-1

(파일명 : 08-01.py)

```
1 ss = '파이썬짱!'
2
3 sslen = len(ss)
4 for i in range(0, sslen) :
5     print(ss[i]+'$', end = '')
```

## 출력 결과

파\$I\$I\$썬\$짱\$!\$

# 문자열 기본 (4)

## ■ 문자열을 거꾸로 출력하는 프로그램 완성

### 소스코드 8-2

(파일명 : 08-02.py)

```
1  ## 변수 선언 부분
2  inStr, outStr = "", ""
3  count, i = 0, 0
4
5  ## 메인(main) 코드 부분
6  inStr = input("문자열을 입력하세요 : ")
7  count = len(inStr)
8
9  for i in range(0, count) :
10     outStr += inStr[count - (i+1)]
11
12  print("내용을 거꾸로 출력 --> %s" % outStr)
```



## Section 02 문자열 함수

# 문자열 함수 (1)

## ■ 문자열 함수 사용하기

- 대문자/소문자의 변환
- upper( ) 함수 : 소문자를 대문자로 변경
- lower( ) 함수 : 대문자를 소문자로 변경
- swapcase( ) 함수 : 대소문자를 상호 변환
- title( ) 함수 : 각 단어의 제일 앞 글자만 대문자로 변환

```
ss = 'Python is Easy. 그래서 programming이 재미있지 말입니다 ^^'  
ss.upper()  
ss.lower()  
ss.swapcase()  
ss.title()
```

### 출력 결과

```
'PYTHON IS EASY. 그래서 PROGRAMMING이 재미있지 말입니다 ^^'  
'python is easy. 그래서 programming이 재미있지 말입니다 ^^'  
'pYTHON IS eASY. 그래서 PROGRAMMING이 재미있지 말입니다 ^^'  
'Python Is Easy. 그래서 Programming이 재미있지 말입니다 ^^'
```



## 문자열 함수 (2)

- 문자열 찾기
- `count()` : 찾을 문자열이 몇 개 들었는지 개수를 셈
- `find()` : 찾을 문자열이 몇 번째 위치하는지 찾음  
`find('찾을 문자열', 시작위치)` 함수 : 시작위치부터 문자열을 찾음
- `rfind()` : 오른쪽부터 셈
- `index()` : `find()` 함수와 동일한 용도, 찾을 문자열이 없다면 오류가 발생
- `startswith()` : 문자열로 시작하면 `True`를, 그렇지 않으면 `False`를 반환
- `endswith()` : 문자열로 끝나면 `True`를 반환

## 문자열 함수 (3)

### 소스코드 8-3

(파일명 : 08-03.py)

```
1 ss=input("문자열 입력==>")
2 print("출력 문자열==>", end='')
3
4 if ss.startswith('(')==False :
5     print("(", end='')
6
7 print(ss, end='')
8
9 if ss.endswith(')')==False :
10    print(")", end='')
```

### 출력 결과

문자열 입력==>파이썬 열공중~~ ← 입력한 문자열  
출력 문자열==>(파이썬 열공중~~)

## 문자열 함수 (4)

- 문자열 공백 제거, 추가, 변경 : strip(), rstrip(), lstrip(), replace()

단 문자열中间的 공백은 제거되지 않음

```
ss = '   파   이   썬   '  
ss.strip()  
ss.rstrip()  
ss.lstrip()
```

출력 결과

```
'파   이   썬'  
'   파   이   썬'  
'파   이   썬   '
```

```
ss = '----파---이---썬----'  
print(ss.strip('-'))  
ss = '<<<파 << 이 >> 썬>>>'  
print(ss.strip('<>'))
```

출력 결과

```
파---이---썬  
파 << 이 >> 썬
```

# 문자열 함수 (5)

- 중간에 나온 공백까지 제거하는 코드

## 소스코드 8-4

(파일명 : 08-04.py)

```
1 inStr=" 한글 Python 프로그래밍 "  
2 outStr=""  
3  
4 for i in range(0, len(inStr)) :  
5     if inStr[i]!=' ' :  
6         outStr+=inStr[i]  
7  
8 print("원 문자열==>"+'['+inStr+']')  
9 print("공백 제거==>"+'['+outStr+']')
```

## 출력 결과

```
원 문자열==>[ 한글 Python 프로그래밍 ]  
공백 제거==>[한글Python프로그래밍]
```

## 문자열 함수 (6)

- 문자열 변경 : `replace('기존 문자열', '새 문자열')`

```
ss = '열심히 파이썬을 공부중~~'  
ss.replace('파이썬', 'Python')
```

출력 결과

'열심히 Python을 공부중~~'

- 문자열을 입력 받아 'o'를 '\$'로 변경

**소스코드 8-5**

(파일명 : 08-05.py)

```
1 ss=input("문자열 입력 ==> ")  
2  
3 print("출력 문자열 ==>", end='')  
4 for i in range(0, len(ss)) :  
5     if ss[i]!='o' :  
6         print(ss[i], end='')  
7     else :  
8         print('$', end='')
```

출력 결과

문자열 입력 ==> IT CookBook for Python ← 입력한 문자열  
출력 문자열 ==> IT C\$\$kB\$\$k f\$r Pyth\$n

# 문자열 함수 (7)

- 문자열 분리, 결합
- `split()` : 문자열을 공백이나 다른 문자로 분리해서 리스트를 반환
- `splitlines()` : 행 단위로 분리
- `join()` : 문자열을 합해줌

```
ss = 'Python을 열심히 공부 중'
ss.split()
ss = '하나:둘:셋'
ss.split(':')
ss = '하나\n둘\n셋'
ss.splitlines()
ss = '%'
ss.join('파이썬')
```

## 출력 결과

```
['Python을', '열심히', '공부', '중']
['하나', '둘', '셋']
['하나', '둘', '셋']
'파%i%썬'
```

## 문자열 함수 (8)

### 소스코드 8-6

(파일명 : 08-06.py)

```
1 ss=input("날짜(연/월/일) 입력 ==> ")
2
3 ssList=ss.split('/')
4
5 print("입력한 날짜의 10년 후 ==> ", end='')
6 print(str(int(ssList[0])+10)+"년 " , end='')
7 print(ssList[1]+"월 ", end='')
8 print(ssList[2]+"일")
```

### 출력 결과

날짜(연/월/일) 입력 ==> 2019/12/31 ← 입력한 문자열  
입력한 날짜의 10년 후는 ==> 2029년12월31년

## 문자열 함수 (9)

- 'map(함수명, 리스트이름)' 함수 : 리스트 값 하나하나를 함수명에 대입

```
before=[ '2019' , '12' , '31' ]  
after=list (map (int , before))  
after
```

출력 결과

```
[2019, 12, 31]
```



# 문자열 함수 (10)

- 문자열 정렬 및 채우기
- center( ) : 숫자만큼 전체 자릿수를 잡은 후 문자열을 가운데 배치
- ljust( ) : 왼쪽에 붙여 출력
- rjust( ) : 오른쪽에 붙여 출력
- zfill( ) : 오른쪽으로 붙여 쓰고 왼쪽 빈 공간은 0으로 채움

```
ss = '파이썬'  
ss.center(10)  
ss.center(10, '-')  
ss.ljust(10)  
ss.rjust(10)  
ss.zfill(10)
```

출력 결과

```
'   파이썬   '  
'---파이썬---'  
'파이썬      '  
'      파이썬'  
'0000000파이썬'
```

# 문자열 함수 (11)

- 문자열 구성 파악 : True 또는 False를 반환
- `isdigit()` : 전체가 숫자로만 구성되어 있는가
- `isalpha()` : 전체가 글자(한글/영어)로만 구성되어 있는가
- `isalnum()` : 전체가 글자와 숫자가 섞여서 구성되어 있는가
- `islower()` : 전체가 소문자로만 구성되어 있는가
- `isupper()` : 전체가 대문자로만 구성되어 있는가
- `isspace()` : 전체가 공백문자로만 구성되어 있는가

```
'1234'.isdigit()  
'abcd'.isalpha()  
'abc123'.isalnum()  
'abcd'.islower()  
'ABCD'.isupper()  
'   '.isspace()
```

출력 결과

모두 True

# 문자열 함수 (12)

## ■ 대소문자 변환 프로그램 완성

### 소스코드 8-7

(파일명 : 08-07.py)

```
1  ## 변수 선언 부분
2  inStr, outStr = "", ""
3  ch = ""
4  count, i = 0, 0
5
6  ## 메인(main) 코드 부분
7  inStr = input("문자열을 입력하세요 : ")
8  count = len(inStr)
9
10 for i in range(0, count) :
11     ch = inStr[i]
12     if ( ord(ch) >= ord("A") and ord(ch) <= ord("Z")) :
13         newCh = ch.lower()
14     elif ( ord(ch) >= ord("a") and ord(ch) <= ord("z")) :
15         newCh = ch.upper()
16     else :
```

## 문자열 함수 (13)

```
17         newCh = ch
18
19         outStr += newCh
20
21     print("대소문자 변환 결과 -->%s" % outStr)
```

- 7행, 8행 : 문자열을 입력 받고 길이를 구함
- 10행~19행 : 문자열의 길이만큼 반복하고 11행에서는 문자 하나를 문자열에서 차례대로 추출. ord(문자) 함수는 문자를 Ascii code값에 해당하는 숫자로 변환. 문자가 A를 ascii code 숫자로 변환한 값(65)과 Z를 숫자로 변환한 값(90) 사이에 있다면 대문자로 인식해서 13행에서 소문자로 변환



Thank You

---