

프로그래밍특론

국민대학교
이정미 교수

kook906@kookmin.ac.kr



1. 문자열 함수

문자열

문자열의 길이 반환: 함수 len()

- 슬라이싱(slicing)
 - 전체에서 일부분을 참조하는 방법
- str[start:end]
 - 문자열 str에서 start 첨자에서 end-1 첨자까지의 문자열을 반환

입력

```
str="Hello python!"  
n=len(str)  
print('문자열',str, '길이', n)
```

출력결과

문자열 Hello python! 길이 13

내장 함수 min() max()

- 인자의 최댓값과 최솟값을 반환하는 함수

- 인자가 문자열 1개
 - 문자열을 구성하는 문자에서 코드 값으로 최대와 최소인 문자를 반환
- 인자가 문자열이 2개 이상
 - 문자열 중 최대와 최소인 문자열을 반환
- 2개 이상의 숫자
 - 최대와 최소 수를 반환

- 내장 함수(built - in function)

- 함수 len(), print(), int(), float(), str(), type(), min(), max() 등
- 파이썬 라이브러리로 인터프리터에서 아무런 설정 없이(나중에 배울 부가적인 설치와 임포트 필요 없이) 바로 사용할 수 있는 함수
- 파이썬 내장 함수 설명
 - docs.python.org/ko/3/library/functions.html

```
>>> min('ipython')
'h'
>>> max('ipython')
'y'
>>> min('3259')
'2'
>>> max('3259')
'9'
>>> min(3, 96.4, 13)
3
>>> max(3, 96.4, 13)
96.4
>>> min('ipython', 'java')
'ipython'
>>> max('ipython', 'java')
'java'
```

replace()

- 메소드 **str.replace(a, b)**
 - 문자열 str에서 a가 나타나는 모든 부분을 b로 모두 바꾼 문자열을 반환
- 메소드 **str.replace(old, new, count)**
 - 문자열 old를 new로 대체하는데, 옵션인 count는 대체 횟수를 지정
 - 옵션인 count가 없으면 모두 바꾸고, 있으면 앞에서부터 지정한 횟수만큼 바꿈

```
>>> str = '파이썬 파이썬 파이썬'
>>> str.replace('파이썬', 'Python!')
'Python! Python! Python!'
>>> str.replace('파이썬', 'Python!', 1)
'Python! 파이썬 파이썬'
>>> str.replace('파이썬', 'Python!', 2)
'Python! Python! 파이썬'
```

replace(), count(), join()

- 메소드 `replace()` 사용
 - 표준 입력으로 받은 실수 형태의 문자열에서 소수점을 제거
- 각 자릿수를 참조해 모두 더하기
- 메소드 `count()`: 부분 문자열 출현 횟수를 반환
- 메소드 `join()`: 문자열의 문자와 문자 사이에 원하는 문자열을 삽입
 - `'->'.join('12345')`
 - 문자열 '12345' 사이에 '->'을 삽입한 문자열을 반환

```
>>> str = '단순한 것이 복잡한 것보다 낫다.'
>>> str.count('복잡')
1
>>> str.count('것')
2
```

문자열을 찾는 메소드 find()와 index()

- 클래스 str에서 부분 문자열 sub가 맨 처음에 위치한 첨자를 반환

- 메소드 str.find(sub): 없으면 -1을 반환
- 메소드 str.index(sub): 없으면 ValueError를 발생
- 역순으로 검색: rfind(), rindex() 지정 문자열 마지막 위치 찾기

- rfind(찾을 문자, 찾기 시작할 위치, 찾기를 끝맺을 위치)

- ```
>> english = "adcjdklcmabc"
```

- ```
>> english.find('c')
```

- 2

- ```
>> english.rfind('c')
```

- 7

# 문자열을 여러 문자열로 나누는 split( ) 메소드

- 문자열 str에서 공백을 기준으로 문자열을 분리
  - 리스트 [항목1, 항목2, ...] 반환
  - 공백은 whitespace라고도 부르며, 공백, 탭, 엔터키(뉴라인) 등
- str.split(',')
  - 괄호 안에 특정한 문자열 값이 있을 경우
    - 이 부분 문자열 값을 구분자를 이용해 문자열을 나눠 줌
- 표준 입력 input( )에서 한 번에 여러 값을 입력
- 최댓값과 최솟값은 함수 max( )와 min( )을 사용



# 메소드 center()와 strip()

- 폭을 지정하고 중앙에 문자열 배치하는 메소드 center( )
- 문자열 앞뒤의 특정 문자들을 제거하는 strip( ) 메소드
  - lstrip()
  - rstrip()
  - strip(' ')

## input() 예제

```
'파이썬 강의'.center(30, "=")
'파이썬 강의'.center(30, "*")
#strip() 제거
' python '.strip()
' pyth on '.lstrip()
'***python--'.strip("* -")
```

# 문자열의 format( ) 메소드를 이용해 간결한 출력 처리

- **str.format(인자들)**

- 문자열 str 중간 중간에 변수나 상수를 함께 출력

```
>>> '3 + 4 = 7'
```

```
'3 + 4 = 7'
```

```
>>> str = '{} + {} = {}'.format(3, 4, 3 + 4)
```

```
>>> print(str)
```

```
3 + 4 = 7
```

수고 하였습니다.