

Chapter. 03

파이썬 프로그래밍 언어

| 문자열 다루기

FASTCAMPUS
ONLINE

금융공학/퀀트 I

강사. 서찬웅

I 문자열

강의내용

1. 문자열
2. 이스케이프문자
3. 논리 연산자
4. 연산자 우선순위

I 문자열이란?

문자열(string)이란 큰따옴표 혹은 작은따옴표안에 표현된 자료형(str) 입니다.

```
text1 = "안녕하세요."
text2 = '안녕하세요'
text3 = """안녕하세요.
수강생 여러분 반갑습니다.
파이팅!!"""
```

- text1은 큰따옴표로 만들어진 문자열입니다.
- text2는 작은 따옴표로 만들어진 문자열입니다.
- text3은 여러 줄을 입력한 문자열입니다. 여러 줄을 입력하고 싶다면 큰따옴표 3개로 시작해서 3개로 끝나면 됩니다.
 - 혹은 작은 따옴표 3개로 시작해서 작은 따옴표 3개로 끝나면 됩니다.
- 한가지 주의사항은 큰따옴표로 시작했다면 큰 따옴표, 작은 따옴표로 시작했다면 작은 따옴표로 끝나야합니다.

```
text4 = '작은 따옴표와 큰 따옴표를 혼합해서 사용하면...'
```

```
File "<ipython-input-4-2c048bcaca4d>", line 1
text4 = '작은 따옴표와 큰 따옴표를 혼합해서 사용하면...'
      ^
```

SyntaxError: EOL while scanning string literal

I 이스케이프문자란?

- 이스케이프 문자는 [이스케이프 시퀀스](#)를 따르는 문자들로서 다음 문자가 특수 문자임을 알리는 백슬래시(\)를 사용한다. 일부 제어 시퀀스인 이스케이프 문자들은 미리 예약 되어있다. – Wikipedia
- 문자열 안에 작은 따옴표나 큰 따옴표를 쓰고 싶다면 이스케이프 문자를 사용하면 됩니다.

| 이스케이프 문자 | 의미 | 예제 |
|----------|------------------|-----------------------------------------------------------------------------|
| \' | 문자열 안에 작은 따옴표 표현 | <pre>print ('Chanwoong\'' fastcampu')</pre> <div>Chanwoong' fastcampu</div> |
| \" | 문자열 안에 큰 따옴표 표현 | <pre>print ('Chanwoong say \'"Hi\'"')</pre> <div>Chanwoong say "Hi"</div> |
| \n | 줄바꿈 | <pre>print ('Chanwoong \nseo')</pre> <div>Chanwoong seo</div> |
| \t | 문자 사이의 tab 간택 | <pre>print ('Chanwoong \tseo')</pre> <div>Chanwoong seo</div> |

윈도우 언어 설정에 따라서 역슬래시 표현이 원화 (₩)로 출력
표현은 원화 표시라도 실제 파이썬에서 역슬래시로 동작

I 시퀀스 자료형이란?

- 여러 객체를 저장하고 있습니다.
- 각 객체들은 순서를 가지고 있습니다.
- 각 객체들은 첨자를 이용하여 접근이 가능합니다.

✓ 종류

- 리스트, 튜플, 문자열

✓ 공통연산

- 인덱싱(indexing)
- 슬라이싱(slicing)
- 연결하기
- 반복하기
- 멤버십 테스트
- 길이정보

I 인덱싱(indexing)

- 인덱싱은 인덱스 번호를 사용하여 특정 위치에 있는 데이터에 접근하는 것을 의미합니다.
- 문자열 안의 각 문자는 위치가 정해져 있습니다. 이를 인덱스 번호로 접근할 수 있습니다.
- 문자열 색인은 0부터 시작하며, 첫 번째 문자의 색인은 0이고 두번째 문자는 1 입니다.

| | | | | | | | | | | | |
|----------|-----|-----|----|----|----|----|----|----|----|----|----|
| 문자열 | f | a | s | t | | c | a | m | p | u | s |
| 양수 Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 음수 Index | -11 | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 |

✓ len() 함수는 시퀀스형 자료안의 전체 요소의 개수를 알려주는 함수입니다.

```
len("fast campus")
```

11

I 슬라이싱(Slicing)

- 시퀀스 자료형의 일정 영역에서 새로운 객체를 반환하며, 결과의 자료형은 원래의 자료형과 동일합니다.
- 슬라이싱은 세가지 값을 가진다.

[start : stop : step]

- start 번호 ~ stop 전 번호까지 선택된다. 예를 들면 [1:5]라고 하면 인덱스 1,2,3,4까지 선택된다.

- 세 번째 step은 데이터를 취하는 간격이고 이를 확장 슬라이싱이라 합니다.
- 세상에서 가장 긴 영어 단어 : Pneumonoultramicroscopicsilicovolcanoconiosis
- 위의 단어를 문자열로 표현하면 아래와 같은 인덱스를 가집니다.

| | | | | | | | | | |
|-----|-----|-----|-----|-------|-----|----|----|----|----|
| P | n | e | u | | ... | o | s | i | s |
| 0 | 1 | 2 | 3 | | ... | 41 | 42 | 43 | 44 |
| -45 | -44 | -43 | -42 | | ... | -4 | -3 | -2 | -1 |

I 슬라이싱(Slicing) 연습

```
tmp = "Pneumonoultramicroscopicsilicovolcanoconiosis"
```

```
tmp[0:5]
'Pneum'
```

```
tmp[10:20]
'tramicrosc'
```

```
tmp[10:]
'tramicroscopicsilicovolcanoconiosis'
```

```
tmp[:-1]
'Pneumonoultramicroscopicsilicovolcanoconiosi'
```

```
tmp[::-2]
'Pemnutairsoislcvlaooiss'
```

```
tmp[::-1]
'sisoinoconaclovociliscipocsrcimartluonomuenP'
```


I 연결 및 반복하기

- 시퀀스형 자료형의 특징은 각각의 시퀀스 변수들을 + 연산자를 사용하여 연결할 수 있습니다.
- * 연산자를 사용하여 데이터를 반복해서 출력할 수 있습니다.

```
"Hello" + "fastcampus"
```

```
'Hellofastcampus'
```

```
"Hello"*6
```

```
'HelloHelloHelloHelloHelloHello'
```

- 위의 예제는 + 연산자를 사용해서 두 문자열을 하나로 합치고, * 연산자를 사용하여 문자열을 반복한 예제입니다.
- 아래는 모모랜드의 뽀뽀 가사의 일부입니다. 아래 코드가 이해가 된다면....

Give it to you, my 눈눈눈눈눈 눈빛
 쏟아지는 my t-t-t-t-t-touch
 하나뿐인 my lo-lo-lo-lo-love, my lover

```
"Give it to you, my " + "눈"*6 + " 눈빛"
```

```
'Give it to you, my 눈눈눈눈눈 눈빛'
```

```
"쏟아지는 my " + "t-t-t-t-t-touch"
```

```
'쏟아지는 my t-t-t-t-t-touch'
```

```
"하나뿐인 my " + ("lo-lo-lo-lo-love")[:1] + "ve" + ", my lover"
```

```
'하나뿐인 my lo-lo-lo-lo-love, my lover'
```

I 멤버십 테스트

✓ in 연산자를 사용하여 대상이 되는 변수 안에 사용자가 질의하는 텍스트가 포함되어 있는지 확인할 수 있습니다..

```
text = "fastcampus"  
'cam' in text
```

True

```
'test' in text
```

False

```
'test' not in text
```

True

I 문자열 메소드

- 문자열은 정확히 말하면 문자열 클래스입니다.
- 클래스안에 구현된 함수들을 메소드라고 부릅니다.
- 문자열 메소드는 텍스트 데이터를 전처리할 때 유용하게 사용할 수 있습니다. 아래는 자주 사용하는 메소드만 적었습니다.
- 특정 문자열 개수 세기 - count()
- 대소문자 변환 - upper(), lower() - 대소문자 변환, capitalize() - 첫 문자를 대문자로
- 검색 관련 - count(s) - 문자열 s가 몇 번 발생한지 카운트, find(s), rfind(s) - s위치, 없으면 -1
index(s) - s위치, 없으면 예외 ValueError 발생
- 편집 및 치환
strip(), lstrip(), rstrip() - 좌우 공백 없앰
replace(a, b) - a를 b로 바꾼다
expandtabs() - 탭을 공백 문자로 바꾼다
- 분리와 결합
split() - 문자열 분리
join() - 문자열 결합

I 정리

- 문자열 정의
- 이스케이프 문자
- 시퀀스형 자료형
- 인덱스, 인덱싱, 슬라이싱
- 문자열의 연결과 반복
- 멤버십 테스트
- 문자열 메소드

감사합니다