

Python 문자열 처리

키워드 : [Python string \(https://docs.python.org/3.7/tutorial/introduction.html#strings\)](https://docs.python.org/3.7/tutorial/introduction.html#strings)

사람이 사용하는 자연어(Natural language)는 프로그램 코드를 통해 여러 가지 연산과 조작이 가능합니다. 이러한 자연어를 우리는 프로그래밍 환경에서 문자열(String)이라고 부르며, 굉장히 다양한 방법을 통해 문자열을 조작할 수 있습니다. 이번 장에서는 파이썬 언어에서 문자열을 다루는 내용에 대해서 알아보도록 하겠습니다.

우선, 파이썬에서 문자열을 선언하는 방법은 작은따옴표(' ')를 이용해서 선언 하는 방법과 큰 따옴표(" ")를 이용해서 선언하는 두 가지 방법이 있습니다. 두 가지 방법은 모두 같은 결과를 반환합니다. 자바스크립트를 공부할 때 봤던 내용이죠^^

```
a = "Life is too short, You need python."
b = "Life is too short, You need python."
c = '''Life is too short, You need python.'''
d = """Life is too short, You need python."""
```

파이썬에서는 다중 라인 문자열을 표현하기 위해서 '''문자열''', """문자열""" 형식을 지원합니다. 또한, 문자열도 순서를 가지는 자료형입니다. 그렇기 때문에, 인덱싱과 슬라이싱이 가능합니다.

```
a = "Life is too short, You need python."
print(a[0]) # 'L'
print(a[:]) # 'Life is too short, You need python.'
```

문자열 안에 있는 공백도 문자열에 포함된다는 점, 함께 기억해주세요! :)

문자열 포매팅

모든 프로그래밍 언어는 문자열을 가공해서 사용하기 위해서 포매팅이라는 방식을 사용하게 됩니다. 포매팅이란, 포맷을 미리 정해놓고, 나중에 포맷에 해당하는 값을 대입하는 형식으로 작성하는 것을 말합니다.

문자열 포매팅 사용법은 다양하게 있습니다. 그중에서, 가장 기본이 되는 것은 예약된 키워드인 % 기호를 미리 문자열에 넣고, % 기호를 한 번 더 쓴 상태에서 원하는 문자열을 순서에 맞게 대입하는 형식입니다.

```
a = 'Life is %s %s' % ('too', 'short')
print(a) # Life is too short
```

미리 정해 놓은 포맷(%)이 하나일 경우 바로 대입해서 사용하고, 만약 % 뒤에 값이 복수 개이면, 튜플(소괄호)로 묶어주어야 합니다. 위 코드에선 '%s'를 포맷으로 작성하고 있습니다. 여기서 '%s'는 지시어이며, 포맷에 값을 할당할 때 어떤 자료형 (Type)으로 표현할지를 결정합니다.

아래는, 파이썬에서 사용되는 다양한 지시어와 의미입니다.

%s : 문자열

%c : 문자 1개

%d : 정수

%f : 실수 (%.소숫점f)

%o : 8진수

%x : 16진수

%% : Literal(문자 '%' 자체)

이 밖에 몇 가지 지시어와 포맷 형식이 더 있습니다. 이와 관련된 내용이 더 궁금하시다면, [여기](https://docs.python.org/3.7/library/stdtypes.html#old-string-formatting) (<https://docs.python.org/3.7/library/stdtypes.html#old-string-formatting>)를 참고해주세요.

또한, 문자열은 리스트를 확장할 때처럼, +, * 기호를 통해 여러 문자열을 하나로 만들거나, 동일한 문자열을 중복 생성할 수도 있습니다.

```
string1 = "문자열"  
string2 = "문자열"  
print(string1 + ' ' + string2) # 문자열 문자열  
print(string1 * 2) #문자열문자열
```

이스케이프 문자

다음은 이스케이프 문자입니다. 파이썬 언어뿐만 아니라, 다양한 언어에서 지원하는 이미 정해진 문자 구문입니다. 모두 역슬래시(\) 기호가 붙는 형태를 나타내고 있습니다. 이름에서 알 수 있듯이, 기존의 문자에서 어떤 의미를 이탈(부여)하여 원하는 형태로 문자를 나타낼 수 있습니다.

아래는, 이스케이프 문자 종류입니다.

\b : 백스페이스

\t : 탭

\n : 줄 바꿈

\f : 다음 페이지 출력

\r : 시작 라인으로 바꿈

**** : 역슬래시

\' : 작은 따옴표

\" : 큰 따옴표

위 이스케이프 문자는 코드 작성 시 유용하게 사용될 수 있는 부분들이 많으니, 자주 사용하는 이스케이프 문자(e.g., \n, \', \") 정도는 기억하고 있는 게 좋습니다. :)

문자열 메서드

파이썬에선 문자열과 관련된 굉장히 다양한 메서드를 지원합니다. 이번 절에서는 다양한 문자열 메서드 중 몇 가지에 대해서 알아보도록 하겠습니다.

문자열 메서드에 대해서 알아보기 전에, `str` 키워드에 대해서 알아보도록 하겠습니다. `str`은 클래스로 정의되어 있습니다. 어떤 입력값을 문자열로 변환하기 위해서 사용될 수 있습니다.

```
a = str(10)
print(a) # '10'
```

format

우선, `format` 메서드는 문자열 포매팅을 할 수 있는 메서드입니다. 문자열 안에 중괄호({})로 포맷을 정해놓고, 나중에 해당 포맷에 해당하는 값을 넣는 형태로 코드를 작성할 수 있습니다.

```
sum = "1 + 1 = {}"  
print(sum.format("귀요미")) # '1 + 1 = 귀요미'
```

split

다음은, `split` 메서드입니다. `split` 메서드는, 인자로 받은 특정 문자를 기준으로 문자열을 구분하여 리스트로 반환합니다. 기본 형식은 `str.split(구분자)`입니다. 말이 조금 어려울 수 있는데요, 코드를 통해 알아보도록 하겠습니다.

```
a = "Life is too short"  
print(a.split(' ')) # [Life, is, too, short]  
b = "Py,th,on"  
print(b.split(',')) # [Py, th, on]
```

문자열 안에 특정 문자를 통해 문자열을 분리하여 조작하기 위해 자주 사용되는 문자열 메서드입니다. 기억해주세요! :)

replace

다음은 `replace` 메서드입니다. `replace` 메서드는 문자열 안에 들어있는 특정 문자를 치환하여 반환합니다. 기본 형식은 `str.replace('기존 문자', '치환할 문자')`입니다.

```
a = '1 + 1 = 귀요미'  
b = a.replace('귀요미', '2')  
print(b) # 1 + 1 = 2
```

문자열 안에 특정 문자를 치환하기 위해 자주 사용되는 문자열 메서드입니다. 기억해주세요! :)

파이썬 언어에서 지원하는 문자열 메서드의 종류는 그 내용을 여기에 정리하기엔 굉장히 많기 때문에, 따로 URL을 첨부하도록 하겠습니다. 문자열 메서드와 관련된 내용이 더 궁금하시다면, [여기](https://docs.python.org/3.7/library/stdtypes.html#string-methods) (<https://docs.python.org/3.7/library/stdtypes.html#string-methods>)를 참고해주세요.

우리가 사용하는 자연어를 프로그램 코드를 통해서 자유롭게 조작하고, 가공할 수 있는 부분은 굉장히 많은 것들을 가능케 합니다. 예를 들어, 인공지능, 번역 처리 등이 있습니다. 하지만, 그만큼 많은 것들에 대한 지식을 필요로 합니다. 코드 작성뿐만 아니라, 자연어 처리를 위해서 필요한 지식도 함께 공부해야 합니다.(제가 지금 그렇습니다..^^)

문자열 처리와 관련된 다양한 알고리즘 문제들도 어렵지 않게 접하실 수 있으므로, 문자열 처리 관련 알고리즘 문제도 공부해보시는 걸 추천드립니다. 감사합니다. :)