

파이썬

7강. 변수와 자료형(문자열)

1. 문자열 (string)

- 문자열은 문자들의 집합으로 단일 따옴표('), 이중 따옴표("), 삼중 따옴표('' or ''')를 사용하여 문자들 을 감싼 형태로 표현한다.
- 여기서 삼중 따옴표는 여러 줄의 문자열을 표현할 때 사용한다.

chapter02.lecture.step04_string.py ↵

Python Console ↵

```
# 문자열 유형 ↵  
oneLine = "this is one line string" ↵  
print(oneLine) ↵
```

this is one line string ↵

```
multiLine = """this is  
multi line ↵  
string""" ↵  
print(multiLine) ↵
```

this is
multi line
string ↵

```
multiLine2 = "this is\nmulti line\nstring" ↵  
print(multiLine2) ↵
```

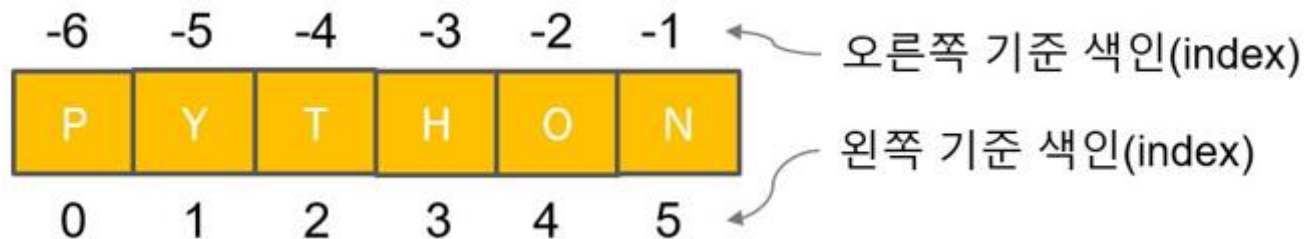
this is
multi line
string ↵

1. 문자열 (string)

- 문자열 유형
 - oneLine 변수는 한 줄의 문자열 객체를 할당받는 변수이다.
 - 단일 따옴표를 이용하여 'this is one line string' 형식으로 표기할 수 있다.
 - multiLine과 multiLine2 변수는 여러 줄의 문자열 객체를 할당받는 변수이다.
 - 두 변수는 모두 동일한 3줄의 문자열을 갖는 변수로 첫 번째 변수는 따옴표 안에서 강제로 줄바꿈을 했으며, 두 번째 변수는 줄바꿈의 기능을 갖는 이스케이프 문자(\n)를 이용했다.

2. 문자열의 특징

- 문자열은 문자들의 집합으로 구성되며 이러한 문자들은 일정한 순서를 가지고 있다. 이러한 순서는 색 인(index)이라는 용어로 불려진다. 색인을 이용하면 문자열을 구성하는 특정 문자를 참조할 수 있다.
- 참조할 수 있다는 의미는 특정 문자를 추출하거나 부분 문자열 객체를 만들 수도 있다는 포괄적 의미가 내포되어 있다. "PYTHON"이라는 문자열은 6개의 문자로 구성된다. 문자열의 왼쪽을 기준으로 첫 번째 'P'는 색인 0을 갖고, 6번째 마지막 'N'은 색인 5을 갖는다. 또한 문자열의 오른쪽으로 기준으로 첫 번째 'N'은 색인 -1을 갖는다.



2. 문자열의 특징

chapter02.lecture.step04_string.py ↵

Python Console ↵

```
# (1) 문자열 색인 ↵  
string = "PYTHON"  
print(string[0])  
print(string[5])  
print(string[-1])  
print(string[-6]) ↵
```

```
↵  
↵  
P  
N  
N  
P ↵
```

```
# (1) 문자열 연산 ↵  
print("python" + " program") # 결합연산자 ↵  
# print("python-" + 3.7 + ".exe") # error  
print("python-" + str(3.7) + ".exe") # python-3.7.  
exe ↵
```

```
↵  
python program  
  
python-3.7.exe ↵
```

```
print("-"*30) # 반복연산자 ↵
```

```
↵  
----- ↵
```

2. 문자열의 특징

- 문자열 색인과 연산 예
- (1) 문자열 색인
 - 문자열 색인에 대한 내용을 코드로 작성하여 출력한 결과이다. `string[0]`은 `string` 객체의 첫 번째 문자를 가리키는 색인이고, `string[-1]`은 `string`의 마지막 문자를 가리키는 색인이다.
- (2) 문자열 연산
 - 문자열과 문자열을 대상으로 덧셈(+) 연산자를 사용할 경우 두 개의 문자열을 결합하는 결합연산자 역할을 한다. 만약 문자열과 숫자를 결합할 경우에는 오류가 발생한다. 이러한 오류를 해결하기 위해서는 `str()` 함수에 숫자를 인자로 넣고, 문자형으로 변환한 후 결합연산을 하면 된다. 또한 문자열과 숫자 사이에 곱셈(*) 연산자를 사용할 경우 숫자의 크기만큼 문자열을 반복 출력하는 반복 연산자 역할을 한다.

3. 문자열 처리함수

- 문자열 처리 함수는 문자열을 수정하거나 조작하는 함수들을 말하며, 다음과 같은 형식으로 해당 함수 들을 호출할 수 있다.
- 함수의 인수는 선택사항이다. 주요 문자열 처리 함수와 기능은 이러한 문자열 처리 함수는 형태소 분석이나 자연어 처리 시 유용하게 이용될 수 있다.

형식 ↵

문자열객체.함수 ([인수]) ↵

3. 문자열 처리함수

함수명 ↴	기능 ↴
s.lower() ↴	문자열 s를 모두 소문자로 변경한다. ↴
s.upper() ↴	문자열 s를 모두 대문자로 변경한다. ↴
s.swapcase() ↴	문자열 s의 대문자는 소문자로, 소문자는 대문자로 변경한다. ↴
s.lstrip() ↴	문자열 s의 왼쪽 공백을 모두 지운다. ↴
s.rstrip() ↴	문자열 s의 오른쪽 공백을 모두 지운다. ↴
s.strip() ↴	문자열 s의 양쪽 공백을 모두 지운다. ↴
s.count(x) ↴	문자열 s에서 x와 일치하는 문자의 개수를 반환한다. ↴
s.find(x) ↴	문자열 s에서 문자 x가 처음으로 나온 위치를 반환한다. (없으면 -1 반환) ↴
s.rfind(x) ↴	문자열 s에서 문자 x가 처음으로 나온 위치를 오른쪽 끝에서 부터 찾는다. ↴
s.index(x) ↴	문자열 s에서 문자 x가 처음으로 나온 위치를 반환한다. (없으면 에러 발생) ↴
'구분자'.join(s ↴	문자열 s를 구성하는 요소 문자를 '구분자'로 결합시킨다. ↴
s.replace(x, r) ↴	문자열 s를 대상으로 x라는 문자를 r이라는 문자로 교체하다. ↴
s.split('구분자') ↴	문자열 s에서 '구분자'를 기준으로 문자열을 분리시킨다. ↴
s.startswith('H' ↴)	H라는 문자로 시작되는지 여부를 True 또는 False로 반환한다. ↴

3. 문자열 처리함수

chapter02.lecture.step04_string.py ↵

Python Console ↵

```
# (1) 특정 글자 수 반환↵
oneLine = "this is one line string"
print('t 글자 수 : ', oneLine.count('t'))↵      t 글자 수 : 2

# (2) 접두어 문자 비교 판단
print(oneLine.startswith('this'))              True↵
print(oneLine.startswith('that'))↵             False↵

# (3) 문자열 교체↵
print(oneLine.replace('this', 'that'))↵         that is one line string↵

# (4) 문자열 분리(split) : 문단 -> 문장↵
multiLine = """this is
multi line
string"""↵
sent = multiLine.split('\n')
print('문장 : ', sent)↵                        문장 : ['this is', 'multi li
ne', 'string']↵

# (5) 문자열 분리(split)2 : 문장 -> 단어↵
words = oneLine.split(' ') # split(sep = '
') : default↵
print('단어 : ', words)↵                      단어 : ['this', 'is', 'one',
'line', 'string']↵

# (6) 문자열 결합(join) : 단어 -> 문장↵
sent2 = ','.join(words) # '구분자'.join(strin↵
g)↵
print(sent2) # this,is,one,line,string↵      this,is,one,line,string↵
```

2. 문자열의 특징

- 해설 문자열 처리 함수

(1) 특정 글자 수 반환

oneLine 문자열에서 count() 함수는 't'문자가 출현된 수를 반환하는 함수이다.

(2) 접두어 문자 비교 판단

oneLine 문자열에서 startswith() 함수는 접두어가 'this'이면 True를 반환하고, 다른 접두어이면 False를 반환한다.

(3) 문자열 교체

oneLine 문자열에서 replace() 함수는 'this'를 'that'으로 교체하는 역할을 한다.

(4) 문자열 분리(split) : 문단 -> 문장

multiLine 문자열을 대상으로 split() 함수는 줄바꿈('\n') 구분자를 기준으로 문자열을 분리한다. 여러 줄의 문장을 갖는 문단으로부터 줄 단위 문장을 만들 때 이용한다.

2. 문자열의 특징

- 해설 문자열 처리 함수

(5) 문자열 분리(split)2 : 문장 -> 단어

multiLine 문자열을 대상으로 split() 함수는 공백(' ') 구분자를 기준으로 문자열을 분리한다. 문단으로부터 단어를 만들 때 이용한다.

(6) 문자열 결합(join) : 단어 -> 문장

문자열이 분리된 words를 대상으로 join() 함수는 구분자 콤마(',')를 기준으로 다시 하나의 문자열로 결합한다. 단어를 문장으로 만들 때 이용한다.

3. 형태소분석

- 형태소란 뜻을 가진 가장 작은 말의 단위를 의미한다. 여기서 말의 단위는 단어를 의미한다. 단어는 다시 조사를 제외한 8개의 품사로 구분할 수 있다.
- 형태소 분석이란 어떤 문단이나 문장을 대상으로 명사 등과 같은 품사 단위로 분해해서 단어의 출현빈도 수를 분석하거나 단어와 단어의 연관성을 분석하는 학문영역을 의미한다.

4. 이스케이프 문자

- 이스케이프(escape)란 어떤 기능으로부터 벗어난 기능을 의미한다.
- 즉 프로그램의 명령어 이외에 특 수기능을 갖는 문자들을 이스케이프 문자(escape character)라고 한다.
- 이스케이프 문자들은 ₩(역슬 래시) 다음에 표시한다. 다음은 이스케이프 문자와 처리 내용이다.

4. 이스케이프 문자

이스케이프 문자 ↵	처리 내용 ↵
\n ↵	줄바꿈 처리(↵ 키 기능) ↵
\t ↵	Tab키 처리(↵ 키 기능) ↵
\r ↵	캐리지 리턴 처리(다음 줄 첫 칸으로 이동 기능) ↵
\f ↵	폼 피드 처리(한 페이지 넘김 기능) ↵
\b ↵	백 스페이스 처리(↵ 키 기능) ↵
\\ ↵ \\ ↵	문자 “\” 처리 ↵
\' ↵	문자 ‘ 처리 ↵
\“ ↵	문자 “ 처리 ↵
\\00 0 ↵	널 문자 처리 ↵

4. 이스케이프 문자

- 위와 같이 이스케이프 문자는 자체 특수한 기능을 가지고 있는데, 간혹 이러한 기능을 차단할 필요가 있다.
- 예를 들면 'wn'이라는 문자열을 콘솔에 출력하기 위해서 `print("wn")`이라고 작성하면 콘솔에는 줄바꿈으로 출력된다.
- 따라서 이스케이프 문자 자체를 출력하기 위해서는 이스케이프 문자 기능을 차단해야한다.
- 다음 형식은 이스케이프 문자 기능을 차단할 수 있는 두 가지 방법이 있다.

4. 이스케이프 문자

- 형식1은 ₩(역슬래시)를 하나 더 붙이는 방법이고, 형식2)는 이스케이프 문자 앞에 r문자를 붙이는 방법 이다. r은 raw를 의미한다.

형식 ↗

- | | |
|----|--------------|
| 1) | "\\이스케이프 문자" |
| 2) | r"이스케이프 문자" |

4. 이스케이프 문자

chapter02.lecture.step05_escape.py ↵

```
# (1) escape 문자 적용 ↵
print('escape 문자 차단 ') ↵
print('\n출력 이스케이프 문자') # \n : 줄 바꿈
기능 ↵
```

```
# (2) escape 문자 기능 차단
print('\n\n출력 이스케이프 기능
차단1')print(r'\n\n출력 이스케이프 기능
차단2') ↵
```

```
# (3) 경로 표현 : C:\Python\test
print('path =', 'C:\Python\test')
print('path =', 'C:\\Python\\test') ↵
print('path =', r'C:\Python\test') ↵
```

Python Console ↵

escape 문자 차단
출력 이스케이프 문자 ↵

\n\n출력 이스케이프 차단1 ↵
\n\n출력 이스케이프 차단2 ↵

path = C:\Python est
path = C:\Python\test
path = C:\Python\test ↵

4. 이스케이프 문자

- 이스케이프 문자 기능 차단 예
- (1) escape 문자 적용
 - “\n출력 이스케이프 문자” 형식으로 문자열을 출력하면 한 줄을 건너 뛴 다음에 ‘출력 이스케이프 문자’의 문자열이 출력된다. 즉 이스케이프 문자 기능이 적용된 것이다.
- (2) escape 문자 기능 차단
 - 이스케이프 문자 앞에 역슬래시(\)를 하나 더 붙이거나 문자열 앞에 r를 붙여서 줄바꿈 이스케이프 문자 기능을 차단하고, 문자열 ‘\n’을 출력한다.
- (3) 경로 표현
 - 폴더와 파일의 경로를 문자열로 출력하기 위해서 이스케이프 문자 기능을 차단하고 있다. 만약 차단하지 않으면 ‘\test’ 문자열의 ‘\t’에 의해서 탭(Tab) 키 기능이 적용된다.

THANK YOU