

07.Python 시퀀스 자료형과 문자열 연산

1. Python 시퀀스 자료형

1] 시퀀스 자료형의 개념

- 저장된 각 요소를 정수 Index를 이용하여 참조가 가능한 자료형
- 시퀀스(Sequence) 자료형 유형 : 문자열, 리스트, 튜플
- 문자열: 시퀀스 자료형의 대표적인 자료형
- 시퀀스 자료형이 가지고 있는 공통적인 연산 존재

2] 시퀀스 자료형이 가지는 공통적인 연산 ■ 인덱싱 (Indexing)

- 슬라이싱 (Slicing)
- 확장 슬라이싱 (Extended Slicing)
- 연결 (Concatenation)
- 반복 (Repetition)
- 멤버십 테스트 (Membership Test)
- 길이 정보 (Length)
- for ~ in 문

3] 슬라이싱 (Slicing)

```
s = 'abcdef'
L = [100, 200, 300]
print s[1:3]
print s[1:]
print s[:]
print s[-100:100]
print
print L[:-1] # L[:2] 와 동일
print L[:2]
```

```
s = 'abcd' print s[::2] #step:2
print s[::-1] #step:-1 - 왼쪽 방향으로 1칸씩
```

슬라이싱: '['가 쓰이고 안쪽에 반드시 ':'(콜론)'이 쓰임
시퀀스 자료형의 일부분을 가져옴
s[1:3] = 1은 start에 해당
s[1:3] = 3은 마지막 인덱스로 포함하지 않음
s[1:3] = 1과 2에 해당하는 문자열만 반환

L[start:end:step]: 인덱싱되어지는 각 원소들 사이의 거리가 인덱스 기준으로 step 만큼 떨어짐

1. Python 시퀀스 자료형

2] 시퀀스 자료형 의 멤버쉽

- 저장된 각 요소가 각 자료형에 포함/존재 하는지 검사

```
s = 'abcde'
print 'c' in s
t = (1,2,3,4,5)
print 2 in t
print 10 in t
print 10 not in t
```

결과

True
True
False
True

'c' in s : c라고 하는 문자열이 s 안에
존재하는지 확인
True 아니면 False로 반환
10 not in t : t 안에 10이 존재하지
않는지 확인

for~in 문

```
for c in 'abcd':
    print c,
```

결과

a b c d

-
- 컨테이너 종류: 시퀀스 자료형, 시퀀스 자료형이 아닌 것
 - 시퀀스 자료형 = 문자열, 리스트, 튜플
 - 시퀀스 자료형이 아닌 것 = 사전

2. Python 문자열 개념 및 기초 연산

1) 문자열 정의 및 기초 연산

- 단일 따옴표 및 이중 따옴표 사용
- *이중 따옴표 사용시 단일따옴표 문자 사용 가능*
- \ 기호는 다음 라인이 현재 라인의 뒤에 이어짐을 나타냄
- multiline은 `"""` 나 `'''` 사용해서 처리 가능
- 다국어 문자의 올바른 표현을 위하여 유니코드 타입 지원
- 문자열 변경을 위해서는 여러 Slicing 연결 활용
- 주의] 문자열 자체가 변경되는 것이 아니라 새로운 문자열을 생성하여 재 할당하는 것임

```
s = 'spam and egg'
s = s[:4] + ',cheese,' + s[5:]
print ( "s[:4] + ',cheese,' + s[5:]      " ,s)
```

결과

```
-----
spam, cheese, and egg
```

```
-----
s[:4] -> spam
s[5:]-> and egg
```