02-3 리스트 자료형

지금까지 우리는 숫자와 문자열에 대해서 알아보았다. 하지만 숫자와 문자열만으로 프로그래밍을 하기엔 부족한 점이 많다. 예를 들어 1부터 10까지의 숫자 중 홀수 모음인 1, 3, 5, 7, 9라는 집합을 생각해 보자. 이런 숫자 모음을 숫자나 문자열로 표현하기는 쉽지 않다. 파이썬에는 이러한 불편함을 해소할 수 있는 자료형이 존재한다. 그것이 바로 이 절에서 공부하게될 리스트(List)이다.

리스트는 어떻게 만들고 사용할까?

리스트를 이용하면 1, 3, 5, 7, 9라는 숫자 모음을 다음과 같이 간단하게 표현할 수 있다.

```
>>> odd = [1, 3, 5, 7, 9]
```

리스트를 만들 때는 위에서 보는 것과 같이 대괄호([ ])로 감싸 주고 각 요소값들은 쉼표(,)로 구분해 준다.

```
리스트명 = [요소1, 요소2, 요소3, ...]
```

여러 가지 리스트의 생김새를 살펴보면 다음과 같다.

```
>>> a = [ ]
>>> b = [1, 2, 3]
>>> c = ['Life', 'is', 'too', 'short']
>>> d = [1, 2, 'Life', 'is']
>>> e = [1, 2, ['Life', 'is']]
```

리스트는 a처럼 아무것도 포함하지 않는, 비어 있는 리스트([ ])일 수도 있고 b처럼 숫자를 요소값으로 가질 수도 있고 c처럼 문자열을 요소값으로 가질 수도 있다. 또한 d처럼 숫자와 문자열을 함께 요소값으로 가질 수도 있으며 e처럼 리스트 자체를 요소값으로 가질 수도 있다. 즉, 리스트 안에는 어떠한 자료형도 포함시킬 수 있다.

(※ 비어 있는 리스트는 a = list()로 생성할 수도 있다.)

리스트의 인덱싱과 슬라이싱

리스트도 문자열처럼 인덱싱과 슬라이싱이 가능하다. 백문이 불여일견. 말로 설명하는 것보다 직접 예를 실행해 보면서 리스트의 기본 구조를 이해하는 것이 쉽다. 대화형 인터프리터로 따라 하며 확실하게 이해하자.

리스트의 인덱싱

리스트 역시 문자열처럼 인덱싱을 적용할 수 있다. 먼저 a 변수에 [1, 2, 3]이라는 값을 설정 한다.

```
>>> a = [1, 2, 3]
>>> a
[1, 2, 3]
```

a[0]은 리스트 a의 첫 번째 요소값을 말한다.

```
>>> a[0]
1
```

아래의 예는 리스트의 첫 번째 요소인 a[0]과 세 번째 요소인 a[2]의 값을 더한 것이다.

```
>>> a[0] + a[2]
4
```

이것은 1 + 3으로 해석되어 4라는 값을 출력한다.

문자열을 공부할 때 이미 살펴보았지만 파이썬은 숫자를 0부터 세기 때문에 a[1]이 리스트 a의 첫 번째 요소가 아니라 a[0]이 리스트 a의 첫 번째 요소임을 명심하자. a[-1]은 문자열에서와 마찬가지로 리스트 a의 마지막 요소값을 말한다.

```
>>> a[-1]
3
```

이번에는 아래의 예처럼 리스트 a를 숫자 1, 2, 3과 또 다른 리스트인 ['a', 'b', 'c']를 포함하도록 만들어 보자.

```
>>> a = [1, 2, 3, ['a', 'b', 'c']]
```

다음의 예를 따라 해보자.

```
>>> a[0]
1
>>> a[-1]
['a', 'b', 'c']
>>> a[3]
['a', 'b', 'c']
```

예상한 대로 a[-1]은 마지막 요소값인 ['a', 'b', 'c']를 나타낸다. a[3]은 리스트 a의 네 번째 요소를 나타내기 때문에 마지막 요소를 나타내는 a[-1]과 동일한 결과값을 보여 준다.

그렇다면 여기서 리스트 a에 포함된 ['a', 'b', 'c']라는 리스트에서 'a'라는 값을 인덱싱을 이용해 끄집어낼 수 있는 방법은 없을까? 다음의 예를 보자.

```
>>> a[-1][0]
'a'
```

위와 같이 하면 'a'를 끄집어낼 수 있다. a[-1]이 ['a', 'b', 'c'] 리스트라는 것은 이미 말했다. 바로 이 리스트에서 첫 번째 요소를 불러오기 위해 [0]을 붙여준 것이다.

아래의 예도 마찬가지 경우이므로 어렵지 않게 이해가 될 것이다.

```
>>> a[-1][1]
'b'
>>> a[-1][2]
'c'
```

[삼중 리스트에서 인덱싱하기]

조금 복잡하지만 다음의 예를 따라 해보자.

```
>>> a = [1, 2, ['a', 'b', ['Life', 'is']]]
```

리스트 a안에 ['a', 'b', ['Life', 'is']]라는 리스트가 포함되어 있고, 그 리스트 안에 다시 ['Life', 'is']라는 리스트가 포함되어 있다. 삼중 구조의 리스트이다.

이 경우 'Life'라는 문자열만 끄집어내려면 다음과 같이 해야 한다.

```
>>> a[2][2][0]
'Life'
```

위의 예는 리스트 a의 세 번째 요소인 리스트 ['a', 'b', ['Life', 'is']]에서 세 번째 요소인 리스트 ['Life', 'is']의 첫 번째 요소를 나타낸다.

이렇듯 리스트를 삼중으로 중첩해서 쓰면 혼란스럽기 때문에 자주 사용하지는 않지만 알아두는 것이 좋다.

리스트의 슬라이싱

문자열과 마찬가지로 리스트에서도 슬라이싱 기법을 적용할 수 있다. 슬라이싱은 "나눈다"는 뜻이라고 했다.

자, 그럼 리스트의 슬라이싱에 대해서 살펴보자.

```
>>> a = [1, 2, 3, 4, 5]
>>> a[0:2]
[1, 2]
```

앞의 예를 문자열에서 슬라이싱했던 것과 비교해 보자.

```
>>> a = "12345"
>>> a[0:2]
'12'
```

2가지가 완전히 동일하게 사용됨을 눈치챘을 것이다. 문자열에서 했던 것과 사용법이 완전히 동일하다.

몇 가지 예를 더 들어 보도록 하자.

```
>>> a = [1, 2, 3, 4, 5]
>>> b = a[:2]
>>> c = a[2:]
>>> b
[1, 2]
>>> c
[3, 4, 5]
```

b라는 변수는 리스트 a의 첫번째 요소부터 두 번째 요소인 a[1]까지 나타내는 리스트이다. 물론 a[2] 값인 3은 포함되지 않는다. c라는 변수는 리스트 a의 세 번째 요소부터 끝까지 나타내는 리스트이다.

[중첩된 리스트에서 슬라이싱하기]

리스트가 포함된 중첩 리스트 역시 슬라이싱 방법은 똑같이 적용된다.

```
>>> a = [1, 2, 3, ['a', 'b', 'c'], 4, 5]
>>> a[2:5]
[3, ['a', 'b', 'c'], 4]
>>> a[3][:2]
['a', 'b']
```

위의 예에서 a[3]은 ['a', 'b', 'c']를 나타낸다. 따라서 a[3][:2]는 ['a', 'b', 'c']의 첫 번째 요소부터 세 번째 요소 직전까지의 값, 즉 ['a', 'b']를 나타내는 리스트가 된다.

리스트 연산자

리스트 역시 + 기호를 이용해서 더할 수 있고, * 기호를 이용해서 반복할 수 있다. 문자열과 마찬가지로 리스트에서도 되는지 직접 확인해 보자.

1) 리스트 더하기(+)

```
>>> a = [1, 2, 3]
>>> b = [4, 5, 6]
>>> a + b
[1, 2, 3, 4, 5, 6]
```

리스트 사이에서 + 기호는 2개의 리스트를 합치는 기능을 한다. 문자열에서 `"abc" + "def" = "abcdef"` 가 되는 것과 같은 이치이다.

2) 리스트 반복하기(*)

```
>>> a = [1, 2, 3]
>>> a * 3
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

위에서 볼 수 있듯이 [1, 2, 3]이라는 리스트가 세 번 반복되어 새로운 리스트를 만들어낸다. 문자열에서 `"abc" * 3 = "abcabcabc"` 가 되는 것과 같은 이치이다.

[초보자가 범하기 쉬운 리스트 연산 오류]

다음과 같은 소스 코드를 입력했을 때 결과값은 어떻게 나올까?

```
>>> a = [1, 2, 3]
>>> a[2] + "hi"
```

a[2]의 값인 3과 문자열 hi가 더해져서 3hi가 출력될 것이라고 생각할 수 있다. 하지만 다음의 결과를 보자. 형 오류(TypeError)가 발생했다. 오류의 원인은 무엇일까?

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

a[2]에 저장된 값은 3이라는 정수인데 "hi"는 문자열이다. 정수와 문자열은 당연히 서로 더할 수 없기 때문에 형 오류가 발생한 것이다.

만약 숫자와 문자열을 더해서 '3hi'처럼 만들고 싶다면 숫자 3을 문자 '3'으로 바꾸어 주어야 한다. 다음과 같이 할 수 있다.

```
>>>str(a[2]) + "hi"
```

str()은 정수나 실수를 문자열의 형태로 바꾸어 주는 파이썬의 내장 함수이다.

리스트의 수정, 변경과 삭제

다음의 예들은 서로 연관되어 있으므로 따로따로 실행하지 말고 차례대로 진행해야 한다.

1. 리스트에서 하나의 값 수정하기

```
>>> a = [1, 2, 3]
>>> a[2] = 4
>>> a
[1, 2, 4]
```

a[2]의 요소값 3이 4로 바뀌었다.

2. 리스트에서 연속된 범위의 값 수정하기

```
>>> a[1:2]
[2]
>>> a[1:2] = ['a', 'b', 'c']
>>> a
[1, 'a', 'b', 'c', 4]
```

a[1:2]는 a[1]부터 a[2]까지를 말하는데 a[2]는 포함하지 않는다고 했으므로 a = [1, 2, 4]에서 a[1]의 값인 2만을 의미한다. 즉, a[1:2]를 ['a', 'b', 'c']로 바꾸었으므로 a 리스트에서 2라는 값 대신에 ['a', 'b', 'c']라는 값이 대입되었다.

[리스트 수정할 때 주의할 점]

2번 예제에서 리스트를 a[1:2] = ['a', 'b', 'c']로 수정하는 것과 a[1] = ['a', 'b', 'c']로 수정하는 것은 전혀 다른 결과값을 갖게 되므로 주의해야 한다. a[1] = ['a', 'b', 'c']는 리스트 a의 두 번째 요소를 ['a', 'b','c']로 바꾼다는 말이고 a[1:2]는 a[1]에서 a[2] 사이의 리스트를 ['a', 'b', 'c']로 바꾼다는 말이다. 따라서 a[1] = ['a', 'b', 'c']로 수정하게 되면 위와는 달리 리스트 a가 [1, ['a', 'b', 'c'], 4]라는 값으로 변하게 된다.

```
>>> a[1] = ['a', 'b', 'c']
>>> a
[1, ['a', 'b', 'c'], 4]
```

3. [ ] 사용해 리스트 요소 삭제하기

```
>>> a[1:3] = [ ]
>>> a
[1, 'c', 4]
```

2번까지 진행한 리스트 a의 값은 [1, 'a', 'b', 'c', 4]였다. 여기서 a[1:3]은 a의 인덱스 1부터 3까지(1<=a<3), 즉 a[1], a[2]를 의미하므로 a[1:3]은 ['a', 'b']이다. 그런데 위의 예에서 볼 수 있듯이 a[1:3]을 [ ]으로 바꿔 주었기 때문에 a에서 ['a', 'b']가 삭제된 [1, 'c', 4]가 된다.

4. del 함수 사용해 리스트 요소 삭제하기

```
>>> a
[1, 'c', 4]
>>> del a[1]
>>> a
[1, 4]
```

del a[x]는 x번째 요소값을 삭제한다. del a[x:y]는 x번째부터 y번째 요소 사이의 값을 삭제한다. 여기서는 a 리스트에서 a[1]을 삭제하는 방법을 보여 준다. del 함수는 파이썬이 자체적으로 가지고 있는 삭제 함수이며 다음과 같이 사용한다.

```
del 객체
```

(※ 객체란 파이썬에서 사용되는 모든 자료형을 말한다.)

리스트 관련 함수들

문자열과 마찬가지로 리스트 변수명 뒤에 '.'를 붙여서 여러 가지 리스트 관련 함수들을 이용할 수 있다. 유용하게 사용되는 리스트 관련 함수 몇 가지에 대해서만 알아보기로 하자.

리스트에 요소 추가(append)

append를 사전에서 검색해 보면 "덧붙이다, 첨부하다"라는 뜻이 있다. 이 뜻을 안다면 아래의 예가 금방 이해가 될 것이다. append(x)는 리스트의 맨 마지막에 x를 추가시키는 함수이다.

```
>>> a = [1, 2, 3]
>>> a.append(4)
>>> a
[1, 2, 3, 4]
```

리스트 안에는 어떤 자료형도 추가할 수 있다.

아래의 예는 리스트에 다시 리스트를 추가한 결과이다.

```
>>> a.append([5,6])
>>> a
[1, 2, 3, 4, [5, 6]]
```

리스트 정렬(sort)

sort 함수는 리스트의 요소를 순서대로 정렬해 준다.

```
>>> a = [1, 4, 3, 2]
>>> a.sort()
>>> a
[1, 2, 3, 4]
```

문자 역시 알파벳 순서로 정렬할 수 있다.

```
>>> a = ['a', 'c', 'b']
>>> a.sort()
>>> a
['a', 'b', 'c']
```

리스트 뒤집기(reverse)

reverse 함수는 리스트를 역순으로 뒤집어 준다. 이때 리스트 요소들을 순서대로 정렬한 다음 다시 역순으로 정렬하는 것이 아니라 그저 현재의 리스트를 그대로 거꾸로 뒤집을 뿐이다.

```
>>> a = ['a', 'c', 'b']
>>> a.reverse()
>>> a
['b', 'c', 'a']
```

위치 반환(index)

index(x) 함수는 리스트에 x라는 값이 있으면 x의 위치값을 리턴한다.

```
>>> a = [1,2,3]
>>> a.index(3)
2
>>> a.index(1)
0
```

위의 예에서 리스트 a에 있는 3이라는 숫자의 위치는 a[2]이므로 2를 리턴하고, 1이라는 숫자의 위치는 a[0]이므로 0을 리턴한다.

아래의 예에서 0이라는 값은 a 리스트에 존재하지 않기 때문에 값 오류(ValueError)가 발생한다.

```
>>> a.index(0)
```

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: 0 is not in list
```

리스트에 요소 삽입(insert)

insert(a, b)는 리스트의 a번째 위치에 b를 삽입하는 함수이다. 파이썬에서는 숫자를 0부터 센다는 것을 반드시 기억하자.

```
>>> a = [1, 2, 3]
>>> a.insert(0, 4)
[4, 1, 2, 3]
```

위의 예는 0번째 자리, 즉 첫 번째 요소(a[0]) 위치에 4라는 값을 삽입하라는 뜻이다.

```
>>> a.insert(3, 5)
[4, 1, 2, 5, 3]
```

위의 예는 리스트 a의 a[3], 즉 네 번째 요소 위치에 5라는 값을 삽입하라는 뜻이다.

리스트 요소 제거(remove)

remove(x)는 리스트에서 첫 번째로 나오는 x를 삭제하는 함수이다.

```
>>> a = [1, 2, 3, 1, 2, 3]
>>> a.remove(3)
[1, 2, 1, 2, 3]
```

a가 3이라는 값을 2개 가지고 있을 경우 첫 번째 3만 제거되는 것을 알 수 있다.

```
>>> a.remove(3)
[1, 2, 1, 2]
```

remove(3)을 한 번 더 실행하면 다시 3이 삭제된다.

리스트 요소 꼬집어내기(pop)

pop()은 리스트의 맨 마지막 요소를 돌려 주고 그 요소는 삭제하는 함수이다.

```
>>> a = [1,2,3]
>>> a.pop()
3
>>> a
[1, 2]
```

a 리스트 [1,2,3]에서 3을 꼬집어내고 최종적으로 [1, 2]만 남는 것을 볼 수 있다.

pop(x)는 리스트의 x번째 요소를 돌려 주고 그 요소는 삭제한다.

```
>>> a = [1,2,3]
>>> a.pop(1)
2
>>> a
[1, 3]
```

a.pop(1)은 a[1]의 값을 꼬집어낸다. 다시 a를 출력해 보면 꼬집어낸 값이 삭제된 것을 확인할 수 있다.

리스트에 포함된 요소 x의 개수 세기(count)

count(x)는 리스트 내에 x가 몇 개 있는지 조사하여 그 개수를 돌려주는 함수이다.

```
>>> a = [1,2,3,1]
>>> a.count(1)
2
```

1이라는 값이 리스트 a에 2개 들어 있으므로 2를 돌려준다.

리스트 확장(extend)

extend(x)에서 x에는 리스트만 올 수 있으며 원래의 a 리스트에 x 리스트를 더하게 된다.

```
>>> a = [1,2,3]
>>> a.extend([4,5])
>>> a
[1, 2, 3, 4, 5]
>>> b = [6, 7]
>>> a.extend(b)
>>> a
[1, 2, 3, 4, 5, 6, 7]
```

a.extend([4,5])는 a += [4,5]와 동일하다.

리스트의 요소를 제거하는 3가지 방법

지금까지 알아본 리스트의 요소를 제거하는 방법은 총 3가지이다.

1. 리스트의 remove 함수 이용하기
2. 리스트의 pop 함수 이용하기
3. del을 이용하기

이 3가지 방법은 결과적으로 리스트의 요소가 삭제된다는 점에서는 동일하지만 삭제에 사용되는 입력값에는 큰 차이점이 있다. 예제로 알아보자.

```
>>> a = [1, 2, 3, 'a', 'b', 'c']
>>> a.remove('a')
>>> a
[1, 2, 3, 'b', 'c']
```

a.remove('a')는 a 리스트의 'a'라는 요소값을 삭제한다. 즉, a.remove(x)의 x에는 인덱스가 아닌 요소값 만을 사용할 수 있다. 참고로 다음과 같이 리스트의 요소를 인덱스로 삭제하려고 하면 다음과 같은 오류가 발생한다.

```
>>> a.remove(4)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: list.remove(x): x not in list
```

pop은 remove와 반대로 요소값으로 삭제할 수는 없고 인덱스로만 삭제가 가능하다. 즉, a.pop(x)의 x에는 a리스트의 인덱스만 가능하다. 다음의 예제로 확인해 보자.

```
>>> a = [1, 2, 3, 'a', 'b', 'c']
>>> a.pop(4)
'b'
>>> a
[1, 2, 3, 'a', 'c']
```

a.pop(4)는 a리스트의 5번째 요소를 삭제한다. 단, pop함수는 삭제된 요소를 리턴받는 특징이 있다.

만약 pop 함수 사용시 인덱스가 아닌 요소값을 이용한다면 다음과 같은 오류를 만나게 될 것이다.

```
>>> a.pop('b')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object cannot be interpreted as an integer
```

del은 리스트의 요소를 삭제하는 함수이므로 당연히 인덱스만 가능하다.

```
>>> a = [1, 2, 3, 'a', 'b', 'c']
>>> del a[4]
>>> a
[1, 2, 3, 'a', 'c']
```

del a[4] 는 a리스트의 5번째 요소를 삭제한다는 것을 알 수 있다.

즉, 결론은 다음과 같다.

1. a.remove(x) - x는 a리스트의 요소값

2. a.pop(x) - x는 a리스트의 인덱스
3. del a[x] - x는 a리스트의 인덱스

연습문제

(연습문제 풀이 : https://wikidocs.net/17090)

[문제1] 리스트 인덱싱

다음과 같은 리스트 a가 있다.

```
>>> a = ['Life', 'is', 'too', 'short', 'you', 'need', 'python']
```

a 리스트를 이용하여 다음과 같은 문자열을 출력하시오.

```
you too
```

[문제2] 리스트 조인

['Life', 'is', 'too', 'short'] 라는 리스트를 Life is too short라는 문자열로 만들어 출력해 보자.

[문제3] 리스트의 갯수

다음과 같은 리스트 a가 있다.

```
>>> a = [1, 2, 3]
```

이 리스트의 갯수(사이즈)를 구하시오.

[문제4] 리스트의 append와 extend

다음과 같은 리스트 a가 있다.

```
>>> a = [1, 2, 3]
```

리스트 a에 [4, 5]를 append 했을 때와 extend했을 때의 차이점은 무엇인가?

[문제5] 리스트의 더하기와 extend

다음과 같은 리스트 a가 있다.

```
>>> a = [1, 2, 3]
```

리스트 a에 [4, 5]를 `+` 기호를 이용하여 더한 결과는 다음과 같다.

```
>>> a = [1, 2, 3]
>>> a = a + [4,5]
>>> a
[1, 2, 3, 4, 5]
```

리스트 a에 [4,5]를 extend를 이용하여 더한 결과는 다음과 같다.

```
>>> a = [1, 2, 3]
>>> a.extend([4, 5])
>>> a
[1, 2, 3, 4, 5]
```

`+` 기호를 이용하여 더한것과 extend한 것의 차이점이 있을까? 있다면 그 차이점에 대해서 얘기해 보자.

[문제6] 리스트 정렬

[1, 3, 5, 4, 2]라는 리스트를 [5, 4, 3, 2, 1]로 만들어보자. (힌트. 리스트의 내장함수인 sort와 reverse를 활용해 보자.)

[문제7] 리스트 삭제

[1, 2, 3, 4, 5]라는 리스트를 [1, 3, 5]로 만들어 보자.

마지막 편집일시 : 2018년 4월 23일 3:42 오후

댓글 76 피드백

위의 예는 리스트 a의 a[3], 즉 세 번째 자리에 5라는 값을 삽입하라는 뜻이다.->

위의 예는 리스트 a의 a[3], 즉 네 번째 자리에 5라는 값을 삽입하라는 뜻이다.

가 맞는것 아닌가요? - 웃는걸음, 2007년 12월 17일 3:20 오후

네, 맞습니다. 수정했습니다.

감사합니다. - 박응용, 2007년 12월 18일 11:20 오후

안녕하세요. 종수라고합니다.
응용님 덕택에 파이썬 공부하고있습니다.
리스트는 lisp하고 비슷한점이 참 많군요;;

근데;;한가지 질문이 있습니다... '리스트에서 몇번째 요소를 삭제해라' 라는 함수는 없나요? - 종수, 2008년 7월 12일 5:53 오후

pop(x)를 사용하시면 됩니다.

pop(x)는 리스트의 x번째 값을 리턴하고 그 값은 삭제합니다. - 박응용, 2008년 7월 12일 9:49 오후

또는 del a[x]를 사용합니다.

리스트 a의 x번째 값을 삭제합니다. - 박응용, 2008년 7월 12일 9:51 오후

글쓴님 덕분에 정말 잘 공부하고 있습니다.
질문이 있어서요.
리스트에서 []가 여러개 있을때, 세는 법을 모르겠습니다.ㅠㅠ
아래 예제와 같이 리스트에 []가 여러개 있을때, 어떻게 세는 건가요?
a[3][:2] 의 의미를 좀 해석해주세요~

```
>>> a = [1, 2, 3, ['a', 'b', 'c'], 4, 5]
>>> a[2:5]
[3, ['a', 'b', 'c'], 4]
>>> a[3][:2]
```

['a', 'b'] - IHSV, 2008년 9월 4일 2:29 오후

우선 a[3]만 뽑아내면 ['a', 'b', 'c'] 가 됩니다.

```
>>> a[3]
['a', 'b', 'c']
```

아래처럼 aa라는 변수를 새로 만들어 볼께요. aa에는 a[3]의 내용이 대입됩니다.

```
>>> aa = a[3]
>>> aa
['a', 'b', 'c']
```

따라서 아래와 같이 될겁니다.

```
>>> aa[:2]
['a', 'b']
```

그런데 aa는 a[3]과 같죠?
그래서 아래도 마찬가지 결과값을 보여줍니다.

```
>>> a[3][:2]
['a', 'b']
```

- 박응용, 2008년 9월 4일 3:38 오후

---

reverse의 결과값이 약간의 오타가 있는듯 합니다.

```
>>>a=['a','b','c']

>>>a.reverse()

>>>a
```

['c','b','a'] - 종, 2008년 12월 10일 7:53 오후

예제에 보시면
a = ['a', 'b', 'c'] 가 아니라 a= ['a','c', 'b']로 되어 있습니다.
이것은 reverse라는 함수가 역순소트가 아니라 단지 역순으로만 만들어 준다는 것을
강조하기 위해서 일부러 'a','b','c'순이 아닌 'a','c','b'순으로 만든것입니다.
감사합니다. - 박응용, 2008년 12월 11일 9:53 오전

위의 초보가 범하기 쉬운 연산 오류 부분에서
`a[2]` + "hi" 를 입력하면

a[2]hi가 출력 됩니다. 아마도 a[2]자체를 문자열로 인식하는 거 같은데.... - 민윤홍, 2009년 1월 11일 2:20 오전

`a[2]`는 문자열 3을 리턴합니다.

`문자는 키보드 자판의 숫자 1 바로 왼쪽에 있는 backquote입니다. - 박응용, 2009년 1월 11일 11:24 오후

님의 강의(?) 감사히 잘 보고 있습니다.
내용물 많아 저는 프린터 하며 잘 보고 있는데 프린터를 하면 꼭 Powered by pyframe 의 카피 문구가 마지막장의 중간에 프린터 됩니다.
도강 하는것 같아 이런것 까지 수정 해 줬으면 하는 맘이 염치가 없네요
저는 파이어 폭스의 기본 인쇄기능으로 인쇄를 하고 있습니다. - 조영갑, 2009년 5월 15일 9:57 오전

네, 감사합니다. ^^
프린트 문제는 제가 어떻게 수정해야 할 지 감이 잘 안오네요.
이번에 PDF를 이용한 출력에 대해서 준비중입니다.
이것이 완성되면 PDF파일이나 아니면 인쇄된 책도 구하실 수 있을 것 같습니다. - 박응용, 2009년 5월 15일 10:45 오전

'리스트 인덱싱 알아보기'의 예제
>>> a = [1, 2, 3]>>> a[1, 2, 3]
이 개행처리 되어 있지 않네요.

```
>>> a = [1, 2, 3]
>>> a
[1, 2, 3]
```

이렇게 되어야 맞다고 생각합니다만... - 중달, 2009년 8월 11일 5:10 오후

@중달 네, 수정했습니다. ^^
감사합니다. - 박응용, 2009년 8월 12일 9:33 오전

예 2) 리스트 수정2

1. >>> a[1:2]
   2 ====> [2] 대괄호 빠졌습니다. - doolyes, 2009년 11월 25일 11:14 오후

>>> a = [1,2,3]>>> a.extend([4,5]) ====> 여기도 개행 처리 안됐네요
>>> a
[1, 2, 3, 4, 5] - doolyes, 2009년 11월 25일 11:37 오후

@doolyes 네, 수정했습니다.
감사합니다. ^^ - 박응용, 2009년 11월 26일 9:28 오전

안녕하세요 하하; 또 궁금한게 생겨서요..

del 객체, 초보가 범하기 쉬운 리스트 연산 오류

우선 다음과 같은 예를 먼저 만들어보자. 여기에서요

str(a[2])+"hi" 를 하면 '3hi' 가 되는데요

'a[2]'+"hi" 를 하면 'a[2]hi' 가 돼요 ㅇㅅㅇ;; 뭐가 잘못된거죠; - 김영국, 2010년 2월 5일 7:19 오후

위 예제는 `a[2]`+"hi"입니다. 'a[2]'+"hi"와 다릅니다.
`a[2]`, 즉 Back Quote (`) 숫자 키 "1" 바로 왼쪽에 있는 키보드 자판입니다. - 박응용, 2010년 2월 8일 11:46 오후

아~~~ 그렇군요! 1옆에 있는 (`) 이거였군요.. 감사합니다 알려주셔서 ^^ - 김영국, 2010년 2월 9일 10:29 오전

덕분에 프로그램밍에 흥미를 느끼고 있습니다. 감사합니다.

공부하면서 오타부분이 있어 알려드립니다. ^^

[3]리스트

예 2) 리스트 수정2

>>> a[1:2] = ['a', 'b', 'c']

>>> a <===============여기가 빠져있습니다.

[1, 'a', 'b', 'c', 4] - 이, 2010년 2월 13일 8:02 오전

수정했습니다.
감사합니다. - 박응용, 2010년 2월 16일 9:43 오전

네, 둘 다 같은 "삭제"입니다. - 박응용, 2011년 6월 13일 9:57 오전

에쿠,, 뭐 잘못 눌러서 질문이 삭제되었네요. 빠른 답변 감사합니다. - 김초보, 2011년 6월 13일 10:04 오전

'a[2]'+"hi" 를 하면 'a[2]hi' 가 돼요 ㅇㅅㅇ;; 뭐가 잘못된거죠;

Commented by 박응용 at 2010년 2월 8일 11:46:00 오후
위 예제는 `a[2]`+"hi"입니다. 'a[2]'+"hi"와 다릅니다.
`a[2]`, 즉 Back Quote (`) 숫자 키 "1" 바로 왼쪽에 있는 키보드 자판입니다.

Python 3.1.2 테스트 중인데,
`a[2]`+"hi" 입력시
File "<stdin>", line 1

'a[2]' + "hi"
^
SyntaxError: invalid syntax
와 같이 표시됩니다. 버전업되면서 Back Quote(`) 키 변경이 있는거 같습니다. - 박동열, 2011년 6월 22일 2:43 오후

추가적으로
>>> a = ['abc', 123, 'you need python']
>>> a.sort()
>>> a
[123, 'abc', 'you need python']

위 부분에서 python 3.1.2 테스트시
a.sort()
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: unorderable types: int() < str()

위와 같이 표시되며, 정렬이 되지 않습니다. - 박동열, 2011년 6월 22일 2:48 오후

숫자와 문자열이 정렬이 되지 않는데 3.0 버전에서 정렬할 수 있는 방법이 있나요? - 나그네, 2011년 6월 26일 8:05 오후

Tracebace이란 문장부터 ValueError라는 문장까지가 에러메시지이다.
오타 발견이요 ! ㅎ
Traceback이란 문장부터 ValueError라는 문장까지가 에러메시지이다.
로 수정하면 맞지 않나요?ㅎ

잘 보고 있습니다 ~ 감사합니다. - lmi, 2011년 7월 3일 10:29 오후

---

a.extend([4,5])는 a + [4,5]와 동일하다.

동일하지 않습니다.
a.extend([4,5])는 a 자체가 확장되지만
a + [4,5]는 a 자체에는 변경이 없습니다.
동일한 결과를 얻으려면 a = a + [4, 5]라고 해야 합니다. - eye4eye, 2011년 10월 19일 6:15 오전

[eye4eye] 네, 그렇네요 ^^
수정하도록 하겠습니다.
알려주셔서 감사합니다. - 박응용, 2011년 10월 19일 8:55 오전

빈 리스트는
empty_list1 = []
empty_list2 = list()
이렇게 만들 수 있다는 내용을 추가해 주세요 - 포인트, 2012년 5월 27일 8:16 오후

음, 혼란이 시작되었어요. ^^ del a[x]로 x위치의 원소를 삭제, pop(x)는 둥근괄호를 써서 x위치의 원소를 삭제, insert(i, x)는 i위치에 x를 삽입, remove(y)는 아이템의 위치를 모를 때, 값으로 아이템(원소)를 삭제합니다.
* python3에서의 a.sort()는 숫자는 되는데, 문자가 있으면 sort()가 안 되죠. python3에서는 어떻게 sort()하죠? - starmomo, 2013년 5월 8일 5:41 오후

안녕하세요!
열심히 파이썬 공부 중인 대학생입니다.
여러모로 많이 도움을 받고 있습니다. 감사합니다!

다름이 아니라, 질문이 하나 있습니다.
문자열, 예를 들어 '124, 21, 4, /n, '214', 'xc' 등이 있을 때에 이것들을
(newline 기호를 제외하고) 하나의 리스트로, 가령 [124, 21, 4, 214, xc] 등으로 변환하거나 새로운 리스트에 삽일할 수 없나요?
사실 과제 중에 메모장에 있는 숫자열 들을 읽어와서 리스트로 변환하려고 하는데
도무지 어떤 함수를 써야할 지 혹은 어떤 방향으로 해야할 지 모르겠네요.
newline 기호를 없애는 것도 그렇고 자꾸 ['124, 21, 4, 214, xc'] 또는 [1, 2, 4, 2, 1, 4, ..., c] 와 같은
형식으로만 변경이 되어서 정말 골치가 아파요!
어떠한 방향으로 손을 대야할 지 지도해 주시면 감사하겠습니다! - Saekyul Jung, 2013년 11월 6일 4:25 오후

//Saekyul Jung 님, 다음처럼 필터를 이용하는 것도 좋을듯 합니다.

>>> lambda x: x=='\n'
<function <lambda> at 0x01714070>
>>> filter(lambda x: x!='\n', [1,2,3])
[1, 2, 3]
>>> filter(lambda x: x!='\n', [1,2,3, '\n'])

[1, 2, 3]
>>>

'\n' 줄바꿈 기호를 리스트에서 제거하는 코드입니다. - 박응용, 2013년 11월 6일 4:30 오후

친절한 설명 감사합니다! 많은 도움이 되었습니다! - Saekyul Jung, 2013년 11월 11일 9:38 오전

좋은 강좌 감사합니다. 무료로 이런 것들을 제공하는게 쉽지 않은 일인데요. - janghyunq, 2014년 1월 13일 8:10 오후

한가지 질문이 있습니다.
몇번째 요소를 삭제해라 중에
del 이 있고 pop 이 있는데
단순히 pop 은 del과는 다르게 그 값을 리턴하는거에 의의가 있어서 사용되는건가요?
아니면 del 과는 뭔가 또 다른점이 있는건가요
- Penron, 2014년 2월 24일 4:34 오후

@Penron님, 말씀하신 것처럼 삭제한 요소를 리턴으로 받아야 할 경우에는 pop, 필요없을 경우에는 del 을 사용하시면 될 것 같습니다. 별다른 의미는 없는것 같네요. 여기도 참고해 보세요: http://stackoverflow.com/questions/11520492/difference-between-del-remove-and-pop-on-lists - 박응용, 2014년 2월 24일 4:44 오후

---

a[3][:2]는 ['a', 'b', 'c']의 a[0]에서 a[2]까지의 값 즉, ['a', 'b']를 나타내는 리스트가 된다. => a[0]에서 a[1]까지의 값으로 보입니다(아니면 a[0]부터 a[2] 직전까지의 값..) ~ 잘 보고 있습니다 감사합니다. - YongTak, 2014년 3월 7일 6:11 오후

@YongTak 님 좋은 의견 감사합니다. 좀 더 명확하게 "~까지의" 라는 말로 바꾸었습니다. - 박응용, 2014년 3월 9일 12:58 오후

좋은 자료 잘 보고 있습니다.
질문이 있습니다.

list = [[1, 2, 3],[4, 5, 6],[7, 8, 9],[10,11,12],[13,14,15]]

위 리스트에서 list내 3, 6, 9, 12, 15만을 따로 추출하여 새로운 리스트는 어떻게 만들 수 있을까요?

감사합니다~~~ - Yumi, 2014년 3월 11일 10:29 오후

@Yumi님 아직 설명되지 않았지만 lambda 함수를 이용하면 다음처럼 구할 수 있습니다.

>>> list = [[1, 2, 3],[4, 5, 6],[7, 8, 9],[10,11,12],[13,14,15]]

>>> map(lambda a:a[-1], list)

[3, 6, 9, 12, 15]

>>> - 박응용, 2014년 3월 12일 12:14 오전

아하~!!!
너무너무 감사합니다.
바로 실행해보았는데, 원하던 바데로 나왔어요!!!
정말 감사드립니다~!!! - Yumi, 2014년 3월 12일 12:22 오전

("Toby likes his art") 이라는 문장을 ["Toby","likes","his","art"] 같은 리스트로 만드는 방법이 있나요? - Jinyoung, 2014년 3월 30일 12:31 오후

@진영님, "Toby likes his art".split() 해 보세요. - 박응용, 2014년 3월 30일 10:35 오후

---

list에 포함된 객체 수는 어찌 알아내는 것이 맞을까요?
저는 __len__() 이라는 함수를 써서 알아내고 있는데 __ 가 붙은 함수를 쓰는 것이 찜찜하군요.
>>> [1, 2, 4, 4].__len__()
4
- BJ, 2014년 4월 9일 9:46 오후

@BJ 님, len([1,2,4,4]) 처럼 사용하시면 됩니다. - 박응용, 2014년 4월 10일 10:04 오전

아하! 답변 감사드립니다. 살아 움직이는 책 좋군요. - BJ, 2014년 4월 10일 11:42 오전

마지막에

a.extend([4,5])는 a = a + [4,5]와 동일하다.

보완하자면, 완전히 동일하지는 않습니다.

```
>>> a = b = [1,2,3]
>>> a = a + [4,5]
>>> b.extend([4,5])
>>> a == b
True
>>> a is b
False
```

a.extend(x) 는 a가 같은 객체를 그대로 참조하지만,

a = a + x의 경우는 a 가 가리키는 객체가 달라집니다.

... 뭐 굳이 이런 것 까지 따질 필요는 없을 것 같지만, 한번 끄적거려봤어요

저도 파이썬 배우는 입장인지라 여기서 많이 도움 얻고 갑니다. - 한재흥, 2014년 5월 4일 12:17 오전

@한재흥 님, 알려주신 내용도 본문에 추가하여 보다 정확한 정보를 제공하는것이 좋겠네요. 알려주셔서 감사합니다. - 박응용, 2014년 5월 7일 9:16 오전

리스트를 만들 때 숫자를 넣어서 만들수는 없나요?
그러니까 for 문을 이용해서 a_i = lines[i].split() 이런식으로 각 line에 해당되는 값들을 공백 빼고 저장하려 하는데... - Yunbin, 2014년 6월 5일 4:55 오후

요즘 파이썬에 관심이 생겨서 여기 사이트를 보구 재미있게 공부하고 있습니다.

그런데, 예제를 따라하다가 서로 다른 타입이 섞여진 리스트일 때 sort에서 에러가 있음을 확인했습니다.

python 3.4 에서 int와 str 타입이 섞여 있는 리스트일 때 sort를 실행하면

TypeError: unorderable types: int() < str() 에러가 발생합니다.

그래서 구글링을 했는데, sort 말구 sorted를 사용해서 해결했습니다.

```
>>> a = ['a','b',1,'c']
>>> sorted(a, key=lambda x:str(x))
[1,'a','b','c']
```

그런데 sorted() 말구, a.sort() 로는 할 수 있는 방법이 없을까요?

참고사이트 http://www.peterbe.com/plog/sorting-mixed-type-lists-in-python-3 - Jason, 2014년 7월 4일 3:00 오후

오타가 있네요: "a[1:3]은 a[1]부터 a[3]까지를 나타내므로" --> "a[1:3]은 a[1]부터 a[2]까지를 나타내므로" - Joshua, 2014년 7월 29일 4:04 오후

@Joshua Huh 님, 알려주셔서 감사합니다. 본문을 수정했습니다. - 박응용, 2014년 8월 4일 8:53 오전

여기다가 질문 해도 되는건가요??

파이썬 공부를 하다 보니 이런 구문이 있는데... 간단한듯 보이면서도, 이해가 너무 안가서 질문드립니다.

```
>>> def mycmp(a1, a2):
return cmp(a2, a1)
```

```
>>> L = [1,5,3,2,4,6]
>>> L.sort(mycmp)
>>> print L
[6, 5, 4, 3, 2, 1]
```

위 구문에서 보시면... mycmp에 반드시 인수가 들어가야 되는 함소인 거 같은데, sort 메소드 안에 함수명만 달랑 들어가서 처리가 되는데,
대략 이해는 가지만, 구체적으로 sort메소드 안에서 mycmp함수와 어떻게 작동되서 저런 결과가 나오는 건지 궁금합니다....

- Jinpyo, 2015년 1월 9일 2:55 오후

@jinpyo님, 보통 L.sort(cmp=mycmp) 사용할 경우 mycmp는 두개의 인수를 받아서 양수, 음수, 0를 리턴하는 비교함수이어야 합니다. 양수나 0인 경우는 바꾸기가 일어나지 않고 음수인 경우 바꾸기가 발생합니다. (소트하기 위해서). 이게 L이라는 리스트의 요소 모두가 양수를 리턴해 줄 때까지 바꾸기가 진행됩니다. 따라서 mycmp를 어떻게 만드느냐에 의해서 소트결과가 정해지겠지요.. - 박응용, 2015년 1월 9일 3:15 오후

우와... 답변이 너무 빨리 달려서 놀랐습니다...^^;;

어쨌든 빠른 댓글 감사드립니다.

추가로 질문 드려보면..

L 리스트의 요소가 mycmp 인수(2개의 인수)에 어떤식으로 매칭이 되서 리턴을 하게 되는 것인지 설명이 가능하실까요??

예를 들어, a1과 a2의 인수에 L 리스트요소가 어떤식으로 연결되서 리턴값을 반환하는건지가...

아니면, sort메소드의 구조를 보면 이해가 더 명확해 질 것 같네요..

초보라서, sort메소드를 어디가면 볼 수 있을지... - Jinpyo, 2015년 1월 9일 3:27 오후

@jinpyo님, https://wiki.python.org/moin/HowTo/Sorting#The_Old_Way_Using_the_cmp_Parameter 여기에 sort cmp에 대한 좀 자세한 내용이 있네요.. - 박응용, 2015년 1월 9일 3:45 오후

안녕하세요. 오탈자를 하나 찾았네요. 헷갈릴수도있을것 같아요.
여기서는 a[2] 값인 '3'이 포함되지 않는다. c라는 변수는 리스트 2번째 요소부터 끝까지를 나타내는 리스트이다.
>> 여기서는 a[2] 값인 '3'이 포함되지 않는다. c라는 변수는 리스트 3번째 요소부터 끝까지를 나타내는 리스트이다. - Sangbeom, 2015년 2월 26일 4:19 오후

@Sangbeom님, 알려주셔서 감사합니다. 본문은 수정 해 놓았습니다. - 박응용, 2015년 2월 26일 5:26 오후

안녕하세요, 궁금한 점이 생겨서 질문드립니다.
a라는 리스트에 원소들이 있을 때

첫번째 코드
```
a.sort()
print (a)
```

두번째 코드
```
print (a.sort())
```

가 있으면 왜 두번째 코드는 제대로 작동하지 않고 None만 출력되는지 궁금합니다.
print (2*3)은 잘 출력되는 걸 보면 print 함수와 괄호 안 함수 간의 우선순위 다툼은 일어나지 않는 것 같은데, 의문입니다. 감사합니다. :) - 김현수, 2015년 3월 8일 5:14 오후

@김현수님, a.sort() 의 sort함수는 리턴값이 없기 때문입니다. a라는 리스트를 소트하기는 하지만 소트된 결과를 리턴하지 않습니다. 소트된 결과를 리턴받으려면 sorted(a) 와 같이 sorted 내장함수를 사용할 수 있습니다. - 박응용, 2015년 3월 8일 8:13 오후

안녕하세요 python 궁금한 점이 생겨서 글써봅니다.

제가 random.uniform생성한 난수를 array에 1000개 정도 집어넣으려하는데.. 어떻게해야할지 감이 안잡힙니다..
C언어라면 for문으로 주소값을 증가시키며 받겠는데, 이건 어떻게 해야할지.. 코멘트좀 부탁드립니다. - 정재, 2015년 3월 20일 1:04 오전

안녕하세요~~
점투 파이썬 정말 잘 보고있는데요..
제가 학교에서 파이썬을 배우거든요..근데 이번실습이 도저히 이해가 아가서요...ㅠㅠㅠ
실습 조건하고 실제 코딩 정답인데요...

혹시 시간 되시면, 코딩에 주석 좀 자세히 달아 주실 수 있을까요?
머리가 나빠서, 코딩을 보고도 뭘 말하는지 이해가 안가서요..ㅠㅠㅠ

최대한 쉽고 자세하게 하나하나 주석달아 주시면 감사하겠습니다.
정말 파이썬 열심히 하려고 하는데..머리가 나쁘니 고생하네요...ㅠㅠㅠ

Password Encryption/Decryption Program
•If number,
•encryption by “number+2”.
If "number+2" larger than 10, mod operation by 10.
•Decryption by “number -2”.
If “number-2” smaller than 0, add 10 to smaller number than 0.
•If character, use the “encryption_key” list.
•encryption_key= (('a','l'),('b','h'),('c','S'),('d','f'),('e','g'),('f','k'),
('g','b'),('h','A'),('i','j'),('j','H'),('k','e'),('l','r'),
('m','F'),('n','L'),('o','I'),('p','n'),('q','i'),('r','u'),
('s','R'),('t','x'),('u','z'),('v','U'),('w','v'),('x','d'),
('y','c'),('z','Y'),('A','p'),('B','D'),('C','K'),('D','O'),
('E','X'),('F','q'),('G','N'),('H','w'),('I','m'),('J','E'),
('K','B'),('L','s'),('M','Z'),('N','J'),('O','C'),('P','V'),
('Q','M'),('R','o'),('S','t'),('T','P'),('U','y'),('V','T'),
('W','Q'),('X','G'),('Y','a'),('Z','W'))
•Another character is keep the character.

```
import math as ma
password_out = ' '
encryption_key = (('a','l'),('b','h'),('c','S'),('d','f'),('e','g'),('f','k'),
            ('g','b'),('h','A'),('i','j'),('j','H'),('k','e'),('l','r'),
            ('m','F'),('n','L'),('o','I'),('p','n'),('q','i'),('r','u'),
            ('s','R'),('t','x'),('u','z'),('v','U'),('w','v'),('x','d'),
            ('y','c'),('z','Y'),('A','p'),('B','D'),('C','K'),('D','O'),
            ('E','X'),('F','q'),('G','N'),('H','w'),('I','m'),('J','E'),
            ('K','B'),('L','s'),('M','Z'),('N','J'),('O','C'),('P','V'),
            ('Q','M'),('R','o'),('S','t'),('T','P'),('U','y'),('V','T'),
            ('W','Q'),('X','G'),('Y','a'),('Z','W'))
print('This program will encrypt and decrypt user passwords')
which = input("Enter 'e' to encrypt a password, and 'd' to decrypt:")

while which != 'e' and which != 'd':
   print('INVALID ENTRY-')
   which = input("Enter 'e' to encrypt a password, and 'd' to decrypt:")

encrypting = (which == 'e')

password_in = input('Enter password:')

if encrypting:
   from_index = 0
   to_index = 1
else:
   from_index = 1
   to_index = 0


for ch in password_in:
   letter_found = False

   for t in encryption_key:
      if ('a' <= ch and ch <= 'z')and ch == t[from_index]:
         password_out = password_out + t[to_index]
         letter_found = True

      elif('A' <= ch and ch <='Z') and ch == t[from_index]:

         password_out = password_out + t[to_index]
         letter_found = True


   if('0' <= ch and ch <= '9'):
         if from_index == 0:
            if(10 <= int(ch) + 2):
```

```
            ch = int(ma.fmod((int(ch)+2),10))
            password_out = password_out + str(ch)
          else:
            password_out = password_out + str(int(ch) + 2)
        else:
          if(0 > int(ch) - 2):
            password_out = password_out + str((int(ch)-2) + 10)
          else:
            password_out = password_out + str(int(ch) - 2)

  elif not letter_found:
    password_out = password_out + ch



if encrypting:
  print('your encrypted password is :',password_out)
else:
  print('your decrypted password is :',password_out)
```

- Jae-Mn, 2015년 4월 23일 6:12 오후

맨 마지막에 나온 extend와 첫부분의 append 기능이 비슷한 것 같은데 용도의 차이같은게 있나요??? - script kiddy, 2015년 5월 19일 11:54 오전

막 python을 공부하고 있는 사람입니다. (버젼 2.7.10)
삭제관련 설명과 다른 결과가 나와서 고수님의 부연 설명을 부탁드립니다.

```
>>> e
[1, 2, ['a', 'b', ['life', 'is'], 3, ['c', 'd', ['best', 'enjoy', ['do', 'you', 'things'], 4, 5]]], 1, 2, 3]
>>> e[2][4][2][2][2]=[]
>>> e
[1, 2, ['a', 'b', ['life', 'is'], 3, ['c', 'd', ['best', 'enjoy', ['do', 'you', []], 4, 5]]], 1, 2, 3]
>>> del e[2][4][2][2][2]
>>> e
[1, 2, ['a', 'b', ['life', 'is'], 3, ['c', 'd', ['best', 'enjoy', ['do', 'you'], 4, 5]]], 1, 2, 3]
```

즉, []로 삭제할 때는 리스트의 구조가 남아있고 null list가 살아있는 것 같습니다. del을 할 경우는 구조도 삭제되는 현상이더군요.
부탁말씀은 왜 이런 현상이 나오는 지? 이 현상을 고려하여 개발시 주의할 사항이 혹 있는지 알고 싶습니다.
- 또 초보, 2015년 7월 6일 2:51 오후

"{0},{1}".format(x,y) 이런 코드를 실행하면

결과로 x, y가 나오게 되는데

결과 값으로 {x},{y} 이렇게 중괄호가 x와 y를 감싸게 하려면 어떻게 해야하나요 - 상훈, 2015년 11월 18일 11:21 오전

파이썬 2.7 입니다.

리스트에 한글 사용 관련 질문입니다.

리스트에 한글을 저장한 후 리스트 자체를 출력하면 저렇게 16진수(?)로 출력이 되는데

리스트의 각 요소를 출력하면 올바르게 출력되네요.

리스트를 출력해도 요소들의 한글이 정상적으로 출력되게 할 수는 없나요??

==========코딩==========

```
#-*- coding: utf-8 -*-

a = ['한글', '출력', 'english', 'life is good']

print a

for l in a:
```

print l

==========출력==========

['\xed\x95\x9c\xea\xb8\x80', '\xec\xb6\x9c\xeb\xa0\xa5', 'english', 'life is good']

한글

출력

english

life is good - 식섭~, 2016년 1월 23일 7:47 오후

이 책에서 공부한 내용으로.. keyword 만 포함된 텍스트 파일로 다른 txt 파일에 있는 정보를 가져오고 싶은데요,,,, readlines 로 가져온 다음에 list 를 서칭하는 부분에서 막혀버렸네요 ㅠㅠ list 를 쉽게 비교하는 방법이 없을까요 ㅠ - 이승룡, 2016년 3월 28일 5:55 오후

[1,2,3]+[4,5,6] 의 결과는 [1,2,3,4,5,6]인데요.

만약에 우리과 흔히 생각하는 [1+4,2+5,3+6]으로 만들 수 있는 명령어가 있을까요?
반복구문을 통해서 각각의 인덱스끼리 더하면 될텐데 한 번에 할 수 있는지가 궁금해서요. - 박, 2016년 9월 18일 1:50 오후

---

mylist = ['a', 'b', 'c', 'd'] 와 mylist = ['a','b','c','d'] 를 입력한 후 insert를 사용하면 전자는 에러가 안뜨는데 후자는 에러가 뜨네요.. 띄어쓰기가 반드시 입력할때는 필요한 건가요? - 장호, 2017년 1월 23일 4:45 오후

---

b라는 변수는 리스트 a의 처음 요소부터 두 번째 요소인 a[2]까지 나타내는 리스트이다.
두 번째 요소가 a[2]가 아닌데 오타가 났네요.
b = a[:2] 이니 a[0](첫 번째 요소)에서 a[1](두 번째 요소)까지 나타내는 것 아닌가요?
a[2] -> a[1] 로 고쳐야 할 것 같네요.
사족으로 "처음 요소" 도 "첫 번째 요소" 로 고치는 게 더 나을 것 같습니다.

글 잘 읽고 있습니다. - 보고룡, 2017년 5월 30일 8:48 오후

@보고룡님, 알려주셔서 감사합니다. 수정완료했습니다. - 박응용, 2017년 5월 30일 10:55 오후

---

del함수로 리스트의 오브젝트를 지우는 것과 list 함수에있는 remover로 리스트의 오브젝트를 지우는 것에 차이점이 있나요?

혹시 이런 자잘한 질문을 할만한 싸이트가 있을까요.. 코딩도장에는 연습문제 같은 것만 있어서 이런 자잘한 것을 질문할 만한 싸이트가 있으면 좋겠어요. - 지호, 2017년 10월 9일 3:15 오후

@지호님, del과 list의 remove는 지우는 방법에 차이가 있습니다. del은 리스트내의 요소를 인덱싱으로 삭게하고 remove(x)는 리스트내의 첫번째 x에 해당되는 값을 삭제하게 됩니다. 즉, 인덱싱으로 삭제할 것인지 요소값으로 삭제할것인지에 대한 차이가 있습니다.

"점프 투 파이썬"의 질문답변 게시판에 대해서 검토해 보겠습니다. 좋은의견 감사합니다. - 박응용, 2017년 10월 10일 9:56 오전

※ 댓글 작성은 로그인이 필요합니다. (또는 피드백을 이용해 주세요.)

이전글 : 02-2 문자열 자료형
다음글 : 02-4 튜플 자료형

TOP