

구조적 자료형

리스트, 튜플, 딕셔너리

리스트 (list)

- 리스트
 - 자료들의 열을 담을 수 있는 자료형
 - 형식: [자료, 자료, ...]
 - 예제
 - [] ← 빈 리스트
 - nums = [1, 2, 3.14, 4, 5]
- 인덱싱 (indexing) 연산
 - 리스트 내에 특정 위치의 자료를 지정하는 것
 - 형식: 변수[위치]
 - 예제
 - idx1 = nums[1]
 - idx2 = nums[-2]

nums	1	2	3.14	4	5
정방향	0	1	2	3	4
역방향	-5	-4	-3	-2	-1



리스트 (list)

- 슬라이싱 (slicing)
 - 리스트 내 특정 위치의 자료를 끊어 오는 것
 - 형식: [시작:끝:간격]
 - 예제
 - nums = [1, 2, 3.14, 4, 5]
 - nums[0]
 - nums[:]
 - nums[0:2] nums[:2] nums[1::2]
 - nums[-1] nums[-3:] nums[::-1]
- 연산자: +, *, in
 - nums = nums+[6,7] nums = [1,2]*3
 - 3.14 in nums

nums	1	2	3.14	4	5
정방향	0	1	2	3	4
역방향	-5	-4	-3	-2	-1



리스트 (list)

- List is mutable!
 - 리스트 내의 특정 값을 인덱싱을 통해 바꿀 수 있음
 - 예제
 - nums = [1, 2, 3.14, 4, 5]
 - nums[2] = 3
 - nums[0], nums[4] = -1, -2
- 리스트 내장 함수
 - 파이썬에 기본적으로 탑재되어 있는 리스트 관련 함수
 - 예제
 - nums = range(5)
 - s = sum(nums)
 - m = max(nums)

nums	-1	2	3	4	-2
정방향	0	1	2	3	4
역방향	-5	-4	-3	-2	-1

```
[0, 1, 2, 3, 4]
10
4
```



리스트 (list)

• 대표적인 리스트 메소드

메소드(함수)	설명
a.append(x)	리스트 a의 마지막에 x추가
a.sort()	리스트 a를 정렬, 내림차순: a.sort(reverse=True)
a.reverse()	리스트 a의 순서를 거꾸로 만든다.
a.index(x)	리스트 a에서 x를 찾아서 그 위치 반환
a.insert(i, x)	리스트 a에서 i 위치에 x 삽입
a.remove(x)	리스트 a에서 처음 나오는 x 삭제
a.pop()	리스트 a의 맨 마지막 요소 반환하고 마지막 요소 삭제
a.count(x)	리스트 a 안에 x가 몇 개 있는지를 반환
a.extend(x)	리스트 a에 리스트 x를 더함(확장)

리스트 (list)

```
>>> a = [1, 2, 3]
>>> a.append(4)
>>> a
[1, 2, 3, 4]
>>> a.append([5, 6])
>>> a
[1, 2, 3, 4, [5, 6]]
>>> a = [1, 4, 2, 3]
>>> a.sort()
>>> a
[1, 2, 3, 4]
>>> a = ['a', 'c', 'b']
>>> a.sort()
>>> a
['a', 'b', 'c']
>>> a = ['abc', 123, 'you need python']
>>> a.sort()
>>> a
[123, 'abc', 'you need python']

>>> a.reverse()
>>> a
['you need python', 'abc', 123]
>>> a = [1, 2, 3]
>>> a.index(3)
2
>>> a.index(1)
0
>>> a = [1, 2, 3]
>>> a.insert(0, 4)
>>> a
[4, 1, 2, 3]
>>> a.insert(3, 5)
>>> a
[4, 1, 2, 5, 3]
>>> a = [1, 2, 3, 1, 2, 3]
>>> a.remove(3)
>>> a
[1, 2, 1, 2, 3]
>>> a.remove(3)
>>> a
[1, 2, 1, 2]
```

튜플 (tuple)

• 튜플(또는 터플로 발음)

- 리스트와 유사하지만 immutable한 자료
- 형식: (자료, 자료, ...)
- 예제
 - () ← 빈 튜플
 - nums = (1, 2, 3)
 - colors = ("red", "green", "blue")

• 인덱싱, 슬라이싱: 리스트와 동일

- nums[1] = 2.4 (X) ← **immutable!**

```
(1,2,3,4)[0] # 1      (a,b,c) = (1,2,3)
(1,2,3,4)[2] # 3      (a,b,c) = 1,2,3
(1,2,3,4)[1:3] # (2,3) a,b,c = (1,2,3)
                  a,b,c = [1,2,3]
```

딕셔너리 (dictionary)

• 딕셔너리

- key, value 형태의 자료를 담을 수 있는 자료형
- 해싱(hashing) 함수를 이용하여 key를 통해 value에 접근
- 형식: {key:value, key:value, ...}
- 예제
 - {}, dict() ← 빈 딕셔너리
 - d = {"pi":3.14, "e":2.7}

선언

```
d = {'pi':3.14, 'e':2.7}
d = dict(pi=3.14, e=2.7)
d = dict([('pi',3.14),('e',2.7)])
```

인덱싱

```
d['pi']
d['pi'] = 3.0
d['zero'] = 0.0
d[(1,2)] = "one and two"
```

딕셔너리 (dictionary)

• 대표적인 딕셔너리 메소드

메소드(함수)	설명
a.keys()	딕셔너리 a의 key들을 모아놓은 dict_keys를 돌려줌
a.values()	딕셔너리 a의 value들을 모아놓은 dict_values를 돌려줌
a.items()	딕셔너리 a의 (key, value) 튜플을 모아놓은 dict_items를 돌려줌
a.clear()	딕셔너리 a의 모든 key:value 쌍들을 삭제하고 딕셔너리를 초기화
a.get(x)	딕셔너리 a의 key가 x인 것의 value를 돌려줌

딕셔너리 (dictionary)

```
d={"pi":3.14, "e":2.7}

k=d.keys()
k_list=list(k)
print(k_list)

v=d.values()
v_list=list(v)
print(v_list)

i=d.items()
i_list=list(i)
print(i_list)

g=d.get("e")
print(g)

['pi', 'e']
[3.14, 2.7]
[('pi', 3.14), ('e', 2.7)]
2.7
```

리스트로 형변환!

실습

- 아래와 같은 리스트의 내용을 역순으로 출력하는 프로그램을 작성하시오.
 - nums = [1, 2, 3, 4, 5]
 - strs = ['abc', 'def', 'ghi']

?

실습

- 다음과 같은 학점 데이터를 딕셔너리에 저장한 후, 이름 순으로 첫번째 학생의 학점을 출력하는 프로그램을 작성하시오.
 - "홍길동":3.4, "임꺽정":4.0, "김철수":2.8, "이영희":3.9

?

질의응답

Q & A

Homepage: <http://nlp.konkuk.ac.kr>
E-mail: nlpdrkim@konkuk.ac.kr



Edited by Harksoo Kim