



# SW멘토링 프로그램

---

비전공자를 위한 파이썬 기초  
한림대학교 영어영문학과 정민



# 리스트 자료형

---

- 리스트 자료형 표현법(대괄호에 넣어서 표현)
  - [1, 2, 3, 4, 5]
- 리스트의 다양한 형태
  - []
  - [1,2,3]
  - ['python', 'is' 'powerful']
  - [1, 2, 'python']
  - [1,2,['python', "is"]]

# 리스트의 인덱싱

---

- 문자열 인덱싱과 방법이 똑같음.

```
In [1]: a = [4, 8, 7]
```

```
In [2]: a
```

```
Out [2]: [4, 8, 7]
```

```
In [3]: a[0]
```

```
Out [3]: 4
```

```
In [6]: a[-1]
```

```
Out [6]: 7
```

# 리스트의 인덱싱

---

- 결과값을 뽑아서 사칙연산을 할 수 있음.

```
In [4]: a[1] + a[2]
```

```
Out [4]: 15
```

```
In [5]: a[1] * a[2]
```

```
Out [5]: 56
```

# 이중 리스트의 인덱싱

- 리스트 안에 리스트가 있을 때는 대괄호를 2번 사용하여 결과값을 가져옴.

```
In [7]: a = [1, 2, 3, ['a', 'b', 'c']]
In [8]: a
Out [8]: [1, 2, 3, ['a', 'b', 'c']]
In [9]: a[-1]
Out [9]: ['a', 'b', 'c']
In [10]: a[-1][0]
Out [10]: 'a'
```

# 삼중 리스트의 인덱싱

---

- 이중 리스트 인덱싱을 할 때처럼 대괄호 3번으로 결과값을 뽑아냄.

```
In [11]: a = [1, 2, ['a', 'b', ['Life', 'is']]]
```

```
In [13]: a
```

```
Out [13]: [1, 2, ['a', 'b', ['Life', 'is']]]
```

```
In [15]: a[-1][-1][0]
```

```
Out [15]: 'Life'
```

# 리스트의 슬라이싱

---

- 인덱싱처럼 문자열의 슬라이싱과 완전히 같음.

```
In [16]: a = [1,2,3,4,5]
```

```
In [17]: a[0:2]
```

```
Out [17]: [1, 2]
```

```
In [18]: a[:2]
```

```
Out [18]: [1, 2]
```

```
In [19]: a[2:]
```

```
Out [19]: [3, 4, 5]
```

# 이중 리스트의 슬라이싱

---

- 인덱싱과 똑같이 이중 리스트를 인덱싱 해준 후에 슬라이싱을 하면 됨.

```
In [20]: a = [1, 2, 3, ['a', 'b', 'c'], 4, 5]
```

```
In [21]: a[2:5]
```

```
Out [21]: [3, ['a', 'b', 'c'], 4]
```

```
In [22]: a[3][:2]
```

```
Out [22]: ['a', 'b']
```

# 리스트 연산

```
In [23]: a = ['a', 'b', 'c']  
        b = ['d', 'e', 'f']
```

- 리스트도 문자열처럼 연산할 수 있습니다.
  - 리스트 더하기

```
In [24]: a + b
```

```
Out [24]: ['a', 'b', 'c', 'd', 'e', 'f']
```

- 리스트 반복하기

```
In [25]: a * 3
```

```
Out [25]: ['a', 'b', 'c', 'a', 'b', 'c', 'a', 'b', 'c']
```

- 리스트 길이 구하기

```
In [26]: len(a)
```

```
Out [26]: 3
```

# 리스트 요소 수정 & 삭제

---

- 리스트 요소 수정

```
In [27]: a = [1,2,3]
```

```
In [28]: a[2] = 4
```

```
In [29]: a
```

```
Out [29]: [1, 2, 4]
```

# 리스트 요소 수정 & 삭제

---

- 리스트 요소 삭제

```
In [30]: a = [1,2,3]
```

```
In [31]: del a[1]
```

```
In [32]: a
```

```
Out [32]: [1, 3]
```

```
In [33]: a = [1,2,3,4,5]
```

```
In [34]: del a[2:]
```

# 리스트 관련 함수들

---

- 리스트 요소 추가(.append())

```
In [36]: a = [1,2,3]
```

```
In [37]: a.append(4)
```

```
In [38]: a
```

```
Out [38]: [1, 2, 3, 4]
```

```
In [39]: a.append([5,6])
```

```
In [40]: a
```

```
Out [40]: [1, 2, 3, 4, [5, 6]]
```

# 리스트 관련 함수들

---

- 리스트 정렬(.sort())

```
In [41]: a = [1,5,3,4,2]
```

```
In [42]: a.sort()
```

```
In [43]: a
```

```
Out [43]: [1, 2, 3, 4, 5]
```

```
In [44]: a = ['b', 'a', 'c']
```

```
In [45]: a.sort()
```

```
In [46]: a
```

```
Out [46]: ['a', 'b', 'c']
```

# 리스트 관련 함수들

---

- 리스트 뒤집기(.reverse())

```
In [47]: a = ['c', 'b', 'a']
```

```
In [48]: a.reverse()
```

```
In [49]: a
```

```
Out [49]: ['a', 'b', 'c']
```

# 리스트 관련 함수들

---

- 리스트 위치 반환(`.index(x)`) - `x`의 위치를 반환

```
In [50]: a = [1,2,3]
```

```
In [51]: a.index(2)
```

```
Out [51]: 1
```

```
In [52]: a.index(3)
```

```
Out [52]: 2
```

# 리스트 관련 함수들

---

- 리스트에 요소 삽입(`.insert(x,y)`) - `x`의 위치에 `y`를 삽입

```
In [53]: a = [1,2,3]
```

```
In [54]: a.insert(0,4)
```

```
In [55]: a
```

```
Out [55]: [4, 1, 2, 3]
```

```
In [56]: a.insert(3,5)
```

```
In [57]: a
```

```
Out [57]: [4, 1, 2, 5, 3]
```

# 리스트 관련 함수들

- 리스트 요소 제거(`.remove(x)`) - 리스트 안에 있는 `x`중에 가장 앞에 있는 요소 제거

```
In [58]: a = [1,2,3,1,2,3]
```

```
In [59]: a.remove(3)
```

```
In [60]: a
```

```
Out [60]: [1, 2, 1, 2, 3]
```

```
In [61]: a.remove(3)
```

```
In [62]: a
```

```
Out [62]: [1, 2, 1, 2]
```

# 리스트 관련 함수들

---

- 리스트 요소 끄집어내기(`.pop(x)`) - `x`번째 안에 있는 요소를 출력하고 삭제

```
In [63]: a = [1,2,3]
```

```
In [64]: a.pop()
```

```
Out [64]: 3
```

```
In [65]: a
```

```
Out [65]: [1, 2]
```

# 리스트 관련 함수들

---

- 리스트에 포함된 요소  $x$ 의 개수 세기(`.count(x)`)

```
In [66]: a = [1,1,1,2,3]
```

```
In [67]: a.count(1)
```

```
Out [67]: 3
```

- 리스트 확장

```
In [68]: a = [1,2,3]
```

```
In [69]: a.extend([4,5])
```

```
In [70]: a
```

```
Out [70]: [1, 2, 3, 4, 5]
```

# 리스트 실습

---

1. '닥터 스트레인지', '스플릿', '럭키' 를 포함한 리스트를 만드세요.
2. 리스트 뒤에 '배트맨' 요소를 추가하세요.
3. '닥터 스트레인지' 와 '스플릿' 사이에 '슈퍼맨'을 추가하세요.
4. '럭키' 요소를 삭제하세요.
5. '스플릿' 요소를 뽑아낸 후 삭제하세요.

# 튜플

- 튜플의 형태

```
In [1]: t1 = (1,)
```

```
In [2]: t2 = 1,2,3
```

```
In [3]: t3 = ('a', 'b', ('ab', 'cd'))
```

- 튜플은 한 개의 요소만을 가질 때는 요소 뒤에 ,(콤마)를 반드시 붙여야 한다.
- 괄호를 생략해도 무관하다.
- 리스트는 항목 값을 변경(삭제)할 수 있지만, 튜플은 불가능하다.
- 리스트와 매우 유사하다.

# 튜플의 인덱싱

---

- 인덱싱

```
In [4]: t1 = (1, 2, 'a', 'b')
```

```
In [5]: t1[0]
```

```
Out [5]: 1
```

```
In [6]: t1[3]
```

```
Out [6]: 'b'
```

- 슬라이싱

```
In [7]: t1[2:]
```

```
Out [7]: ('a', 'b')
```

# 튜플의 더하기, 곱하기

- 더하기

```
In [8]: t1 = (1, 2, 'a', 'b')  
        t2 = (3, 4)
```

```
In [9]: t1 + t2
```

```
Out [9]: (1, 2, 'a', 'b', 3, 4)
```

- 곱하기

```
In [10]: t2 = (3, 4)  
         t2 * 3
```

```
Out [10]: (3, 4, 3, 4, 3, 4)
```

# 튜플의 길이 구하기

---

- `len()` 함수를 이용한 길이 구하기

```
In [11]: t1 = (1, 2, 'a', 'b')  
         len(t1)
```

```
Out[11]: 4
```

# 리스트, 튜플

---

- 리스트에서 튜플 만들기

```
In [21]: a = [1,2,3,4,5,6,7]
```

```
In [22]: tuple(a)
```

```
Out [22]: (1, 2, 3, 4, 5, 6, 7)
```

- 튜플에서 리스트 만들기

```
In [23]: b = (1,2,3,4,5,6,7)
```

```
In [24]: list(b)
```

```
Out [24]: [1, 2, 3, 4, 5, 6, 7]
```

# 튜플 실습

---

- (1,3,5)라는 튜플에 7값을 추가하여 (1,3,5,7)처럼 만들어 출력해보세요.
- [1,2,3,4]라는 리스트를 튜플로 전환해보세요.
- (1,3,5,7)이라는 튜플을 리스트로 전환해보세요.