

파이썬 프로그래밍

16차시

여러 자료 값을
편리하게 처리하는
리스트



⚠ 학습개요

- ... 리스트 개념 이해
- ... 여러 생성 방법
 - 항목이 있는 리스트
 - 빈 리스트
 - 추가 메소드 `append()`
- ... 메소드 `count()` `index()`
- ... 항목 참조와 수정
- ... 중첩 리스트

! 학습목표

- ... 자료형 리스트를 이해하고 생성할 수 있다.
- ... 빈 리스트를 생성하고 메소드 `append()`로 항목을 추가할 수 있다.
- ... 메소드 `count()`와 `index()`를 사용할 수 있다.
- ... 리스트의 항목을 참조하고 수정할 수 있다.
- ... 중첩된 리스트를 활용할 수 있다.

Chapter 1.

리스트 개요

P Y T H O N P R O G R A M M I N G

⚠️ 관련된 나열 항목을 관리하는 리스트

+ 항목의 나열인 시퀀스, 리스트

- 리스트는 **콤마**로 구분된 항목(또는 원소)들의 리스트로 표현
- 항목은 정수, 실수, 문자열, 리스트 등이 모두 가능
- 항목 순서는 의미가 있으며, 항목 자료 값은 **중복 되도 무관**
 - 리스트는 **대괄호(square brackets) []** 사이에 항목을 기술

⚠️ 관련된 나열 항목을 관리하는 리스트

```
menu = ['coffee', 'beverage', 'ade']  
coffee = ['에스프레소', '아메리카노', '카페라테', '카페모카']  
coffeeprice = [['에스프레소', 2500], ['아메리카노', 2800], ['카페라테', 3200]]
```

```
>>> coffee = ['에스프레소', '아메리카노', '카페라테', '카페모카']  
>>> print(coffee)  
['에스프레소', '아메리카노', '카페라테', '카페모카']  
>>> type(coffee)  
<class 'list'>
```


⚠ 간단한 커피 메뉴 만들기 1

[코딩실습] 간단한 커피 메뉴 만들기 1

난이도 기본

```
1. menu = ['COFFEE', 'BEVERAGE', 'ADE']
2. coffee = ['에스프레소', '아메리카노', '카페라테', '카페모카']
3.
4. print('=' * 45)
5. for category in menu:
6.     print('{:^15s}'.format(category), end = ' ')
7. print()
8. print('=' * 45)
9.
10. for ckind in coffee:
11.     print('{:^10s}'.format(ckind))
```

⚠ 간단한 커피 메뉴 만들기 1

주의	1, 2번 줄의 리스트를 만드는 첫 문장에 주의하자.		
결과	COFFEE	BEVERAGE	ADE
	에스프레소 아메리카노 카페라테 카페모카		

Chapter 2.

빈 리스트와 인덱싱

P Y T H O N P R O G R A M M I N G

⚠ 빈 리스트의 생성과 항목 추가

+ []

- 빈 대괄호로 빈 리스트 생성

```
>>> pl = [] # 빈 리스트  
>>> print(pl)  
[]
```

+ list()

- 인자가 없는 내장 함수 **list()**로도 빈 리스트를 생성

⚠ 빈 리스트의 생성과 항목 추가

+ append()

- 리스트의 메소드 **append**(삽입할 항목)
- 리스트의 가장 뒤에 항목을 추가

```
>>> pl = list()
>>> pl.append('C++')
>>> pl.append('java')
>>> print(pl)
['C++', 'java']
```

⚠ 일상 코딩: 리스트의 메소드 `append()`로 편의점의 품목 리스트 생성

[코딩실습] 리스트로 편의점에서 구입할 품목 만들기

난이도 기본

```
1. goods = []
2. for i in range(3):
3.     item = input('구입할 품목은 ? ')
4.     goods.append(item)
5.     print(goods)
6. print('길이: %d' % len(goods))
```

주의 2번 줄의 변수 `i`는 사용하지 않으므로 변수 `_`를 사용할 수 있다

결과

```
구입할 품목은? 과자
['과자']
구입할 품목은? 우유
['과자', '우유']
구입할 품목은? 세면도구
['과자', '우유', '세면도구']
길이 : 3
```

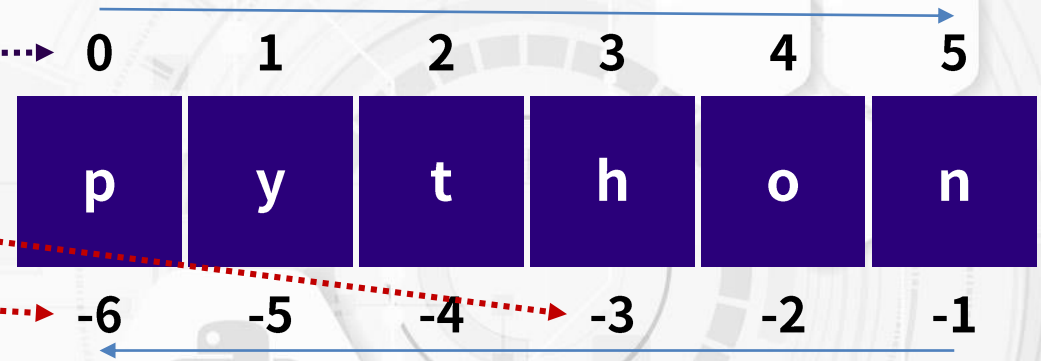
⚠ 리스트 항목 참조

+ 문자열 'python'으로 만들어진 리스트 py

- 길이가 6이므로 [0~5] 그리고 [-6~-1]까지 첨자를 사용

```
>>> py = list('python')
>>> print(py[0], py[5])
p n
>>> print(py[-3], py[-1])
h n
>>> print(py[-len(py)], py[len(py)-1])
p n
```

오른차순 첨자: 0~[len(시퀀스)-1]



내림차순 첨자: -1 ~ -[len(시퀀스)]

⚠ 리스트 항목 참조

[코딩실습] 프로그래밍 언어 리스트에서 첨자로 항목 참조

난이도 기본

```
1. pl = ['C', 'C++', 'Python', 'Java']
2. print(pl[0])
3. print(pl[2])
4. print()
5.
6. for i in range(len(pl)):
7.     print(pl[i])
```

결과 예측해 보고 교재 p165 참고

⚠ 일상 코딩: 가위바위보 게임을 위한 준비

+ random 모듈의 메소드 **choice**(리스트)

- 리스트 중의 항목 하나를 무작위(랜덤)로 반환

[코딩실습] 가위바위보 리스트 항목 참조

난이도 기본

```
1. rsp = ['가위', '바위', '보']
2. for i in range(len(rsp)):
3.     print(rsp[i], end = ' ')
4. print()
5.
6. from random import choice
7. print('컴퓨터의 가위 바위 보 5개')
8. for i in range(5):
9.     print(choice(rsp))
```

결과 예측해 보고 교재 p166 참고

Chapter 3.

메소드 `count()`, `Index()`

P Y T H O N P R O G R A M M I N G

⚠ 리스트의 메소드 `count()`와 `index()`

+ 리스트 메소드 `count()`

- 값을 갖는 항목의 수

+ `index()`

- 인자인 값의 항목이 위치한 첨자를 반환
- 동일한 값이 여러 개이면 첫 번째로 나타난 위치의 첨자 반환

⚠ 리스트의 메소드 count()와 index()

```
>>> top = ['BTS', '볼빨간사춘기', 'BTS', '블랙핑크', '태연', 'BTS']
>>> top.count('BTS')
3
>>> top.index('볼빨간사춘기')
1
>>> top.index('BTS')
0

>>> top.index('여자친구')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: '여자친구' is not in list
```

⚠ 일상 코딩: 중국집에서 주문한 음식을 다시 주문 추가하고 삭제하기

```
food = ['짜장면', '짬뽕', '우동', '울면']  
# 우동을 물만두로 주문 변경  
food[food.index('우동')] = '물만두'
```



⚠ 일상 코딩: 중국집에서 주문한 음식을 다시 주문 추가하고 삭제하기

[코딩실습] 중국집에서 음식 주문하기

난이도 응용

```
1. food = ['짜장면', '짬뽕', '우동', '울면']
2. print(food)
3. # 탕수육 주문 추가
4. food.append('탕수육')
5. print(food)
6. # 짬뽕을 굴짬뽕으로 주문 변경
7. food[1] = '굴짬뽕'
8. print(food)
9.
10. # 우동을 물만두로 주문 변경
11. food[food.index('우동')] = '물만두'
12. print(food)
```

주의 7번 줄에서 첨자는 0부터 시작이므로 food[1]은 '짬뽕'을 수정한다.

⚠ 리스트의 항목으로 리스트 구성

+ 리스트 내부에 다시 리스트가 항목으로 사용 가능

```
>>> animal = [['사자', '코끼리', '호랑이'], '조류', '어류']
>>> print(animal)
[['사자', '코끼리', '호랑이'], '조류', '어류']
>>> print(animal[0])
['사자', '코끼리', '호랑이']
```

+ animal에서 '코끼리'를 참조

- 첨자를 두 번 사용해 `animal[0][1]`로 표시

```
>>> print(animal[0][1])
코끼리
```

⚠ 일상 코딩: 동물의 분류와 동물 이름을 리스트로 처리

[코딩실습] 동물의 분류를 리스트로 처리

난이도 응용

```
1. animal = [['사자', '코끼리', '호랑이'], '조류', '어류']
2. print(animal)
3.
4. for s in animal:
5.     print(s)
6. print()
7.
8. brid = ['독수리', '참새', '까치']
9. fish = ['갈치', '붕어', '고등어']
10. animal[1:] = [brid, fish]
11. print(animal)
12.
13. for lst in animal:
14.     for item in lst:
15.         print(item, end = ' ')
16.     print()
17. print()
```

⚠ 일상 코딩: 동물의 분류와 동물 이름을 리스트로 처리

결과

```
[[ '사자', '코끼리', '호랑이'], '조류', '어류']
```

```
['사자', '코끼리', '호랑이']
```

```
조류
```

```
어류
```

```
[[ '사자', '코끼리', '호랑이'], ['독수리', '참새', '까치'], ['갈치', '붕어', '고등어']]
```

```
사자 코끼리 호랑이
```

```
독수리 참새 까치
```

```
갈치 붕어 고등어
```

! 리스트

- ... 개념 이해
- ... 여러 생성 방법
 - 항목이 있는 리스트
 - 빈 리스트
 - 추가 메소드 `append()`
- ... 메소드 `count()` `index()`
- ... 항목 참조와 수정
- ... 중첩 리스트