

# 리스트 활용하기

## 들어가기

계속해서 리스트와 관련된 내장 함수들을 살펴본다.

## 학습목표

리스트를 이해하고, 내장 함수를 활용해 리스트를 만들 수 있다.


## 핵심 키워드

- 리스트
- 내장 함수

## 학습내용


## 연산자 활용

### 리스트 병합

리스트 타입도  연산자를 활용해서 서로 다른 리스트를 더할 수 있다.

```
a = [1, 2, 3]
b = [4, 5, 6]
c = a + b
print(c)
# [1, 2, 3, 4, 5, 6]로 출력됩니다.
```

## 리스트 슬라이싱

리스트도  콜론을 이용해서 자를 수 있다.

중요한 것은 `t[1:3]` 같은 경우, 3번째 인덱스에 해당하는 항목은 포함되지 않는다.

```
t = [9, 41, 12, 3, 74, 15]
print(t[1:3])
print(t[:4])
```

```
print(t[3:])
print(t[:])

# [41, 12]
# [9, 41, 12, 3]
# [3, 74, 15]
# [9, 41, 12, 3, 74, 15] 로 출력됩니다.
```

## dir() 메서드

특정 타입에서 사용할 수 있는 메서드의 목록을 볼 수 있는 함수이다.

```
x = list()
print(dir(x))
```

## 리스트 만들기

빈 리스트 만들기 → 항목 추가하기 → 항목 정렬하기 → `in` 을 활용해 'Glenn'이 친구 목록에 있는지 확인하기

```
friends = list()
friends.append('Joseph')
friends.append('Glenn')
friends.append('Sally')
print(friends)
# ['Joseph', 'Glenn', 'Sally']
friends.sort()
print(friends)
# ['Glenn', 'Joseph', 'Sally']
print('Glenn' in friends)
# True로 출력됩니다.
```