

---

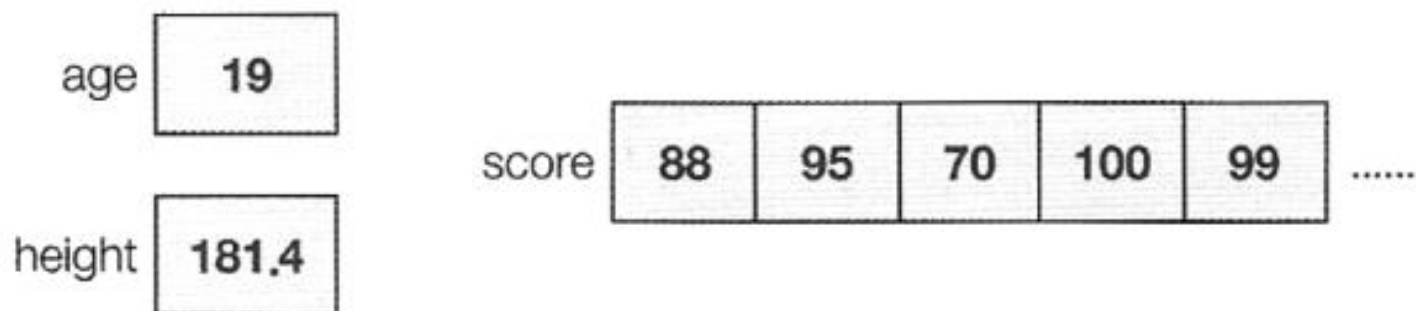
# 리스트와 튜플

# 리스트

---

## ❖ 자료의 집합

```
score = [ 88, 95, 70, 100, 99 ]  
name = [ "최상미", "이한승", "김기남" ]
```



```
namescore = [ "최상미", 88, "이한승", 95 ]
```

# 리스트

## ❖ 자료의 집합

```
score = [88, 95, 70, 100, 99]  
total = 0
```

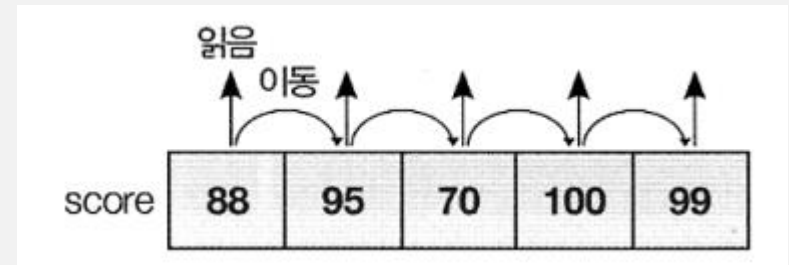
```
for s in score:  
    total += s
```

```
print("총점: ", total)  
print("평균: ", total/len(score))
```

```
총점: 452  
평균: 90.4
```

```
print(list("Korea"))
```

```
['K', 'o', 'r', 'e', 'a']
```



# 리스트

---

## ❖ 자료의 집합

```
s = "python"
for c in s:
    print(c, end = ",")
```

p,y,t,h,o,n,

```
s = "python"
for i in range(len(s)):
    print(s[i], end = ",")
```

p,y,t,h,o,n,

# 리스트

## ❖ 리스트의 요소

- 리스트[인덱스]
- 리스트[begin:end:step]

```
score = [88, 95, 70, 100, 99]
print(score[0])
print(score[2])
print(score[-1])
```

88  
70  
99

0	1	2	3	4
88	95	70	100	99
-5	-4	-3	-2	-1

# 리스트

---

## ❖ 리스트의 요소

```
nums = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(nums[:])
print(nums[2:5])
print(nums[:4])
print(nums[6:])
print(nums[1:7:2])
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[2, 3, 4]
[0, 1, 2, 3]
[6, 7, 8, 9]
[1, 3, 5]
```

```
score = [88, 95, 70, 100, 99]
print(score[2])
score[2] = 55
print(score[2])
```

```
70
55
```

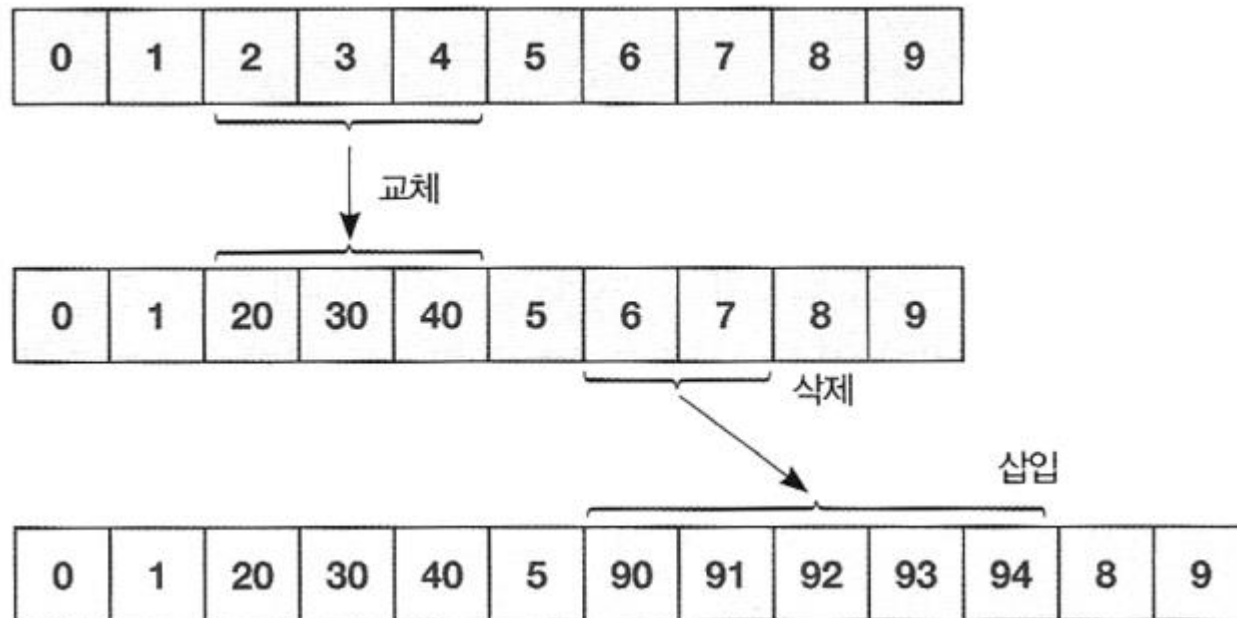
# 리스트

## ❖ 리스트의 요소

```
# 기존값을 삭제하고, 새로운 값으로 대체(삽입)  
nums = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
nums[2:5] = [20, 30, 40]  
print(nums)  
nums[6:8] = [60, 70, 80, 90]  
print(nums)
```

[0, 1, 20, 30, 40, 5, 6, 7, 8, 9]

[0, 1, 20, 30, 40, 5, 60, 70, 80, 90, 8, 9]



# 리스트

---

## ❖ 리스트의 요소

```
list1 = [1, 2, 3, 4, 5]
list2 = [10, 11]
listadd = list1 + list2
print(listadd)
listmulti = list2*3
print(listmulti)
```

```
[1, 2, 3, 4, 5, 10, 11]
[10, 11, 10, 11, 10, 11]
```



# 리스트

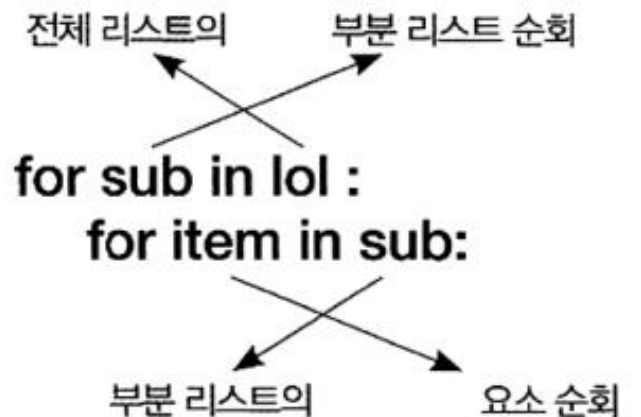
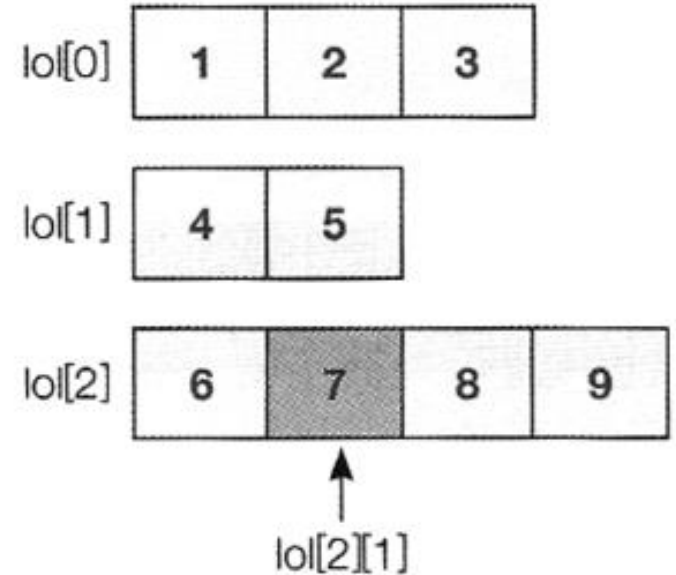
## ❖ 이중 리스트

```
lol = [  
    [1, 2, 3],  
    [4, 5],  
    [6, 7, 8, 9]  
]
```

```
print(lol[0])  
print(lol[2][1])
```

```
for sub in lol:  
    for item in sub:  
        print(item, end=' ')  
    print()
```

```
[1, 2, 3]  
7  
1 2 3  
4 5  
6 7 8 9
```



# 리스트

## ❖ 이중 리스트

```
score = [  
    [88, 76, 92, 98],  
    [65, 70, 58, 82],  
    [82, 80, 78, 88]  
]
```

```
total = 0  
totalsub = 0
```

```
for student in score:  
    subject_total = 0  
    for subject in student:  
        subject_total += subject  
  
    subjects = len(student)  
    print("총점 %d, 평균 %.2f" % (  
        subject_total, subject_total/subjects))  
    total += subject_total  
    totalsub += subjects  
  
print("전체평균 %.2f" % (total/totalsub))
```

88	76	92	98	→ 학생 총점
65	70	58	82	
82	80	78	88	
↓ 과목 총점				

총점 354, 평균 88.50

총점 275, 평균 68.75

총점 328, 평균 82.00

전체평균 79.75

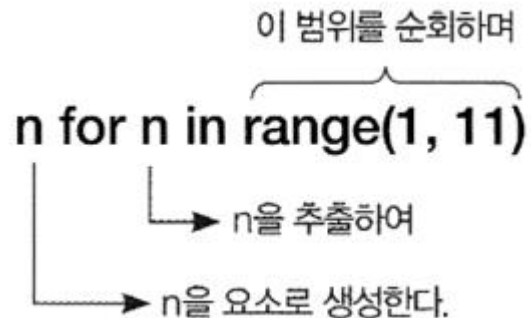
# 리스트

## ❖ 컴프리헨션

- [수식 for 변수 in 리스트 if 조건]
  - 내부의 리스트를 순회하며 각 요소에 대한 수식을 적용하여 최종 요소를 생성
  - if 조건을 추가하면 조건을 만족하는 요소만 추가

```
print([n for n in range(1, 11)])
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```



# 리스트

---

## ❖ 컴프리헨션

```
nums = [ n*2 for n in range(1, 11)]  
for i in nums:  
    print(i, end = ', ')
```

2, 4, 6, 8, 10, 12, 14, 16, 18, 20,

```
nums = []  
for n in range(1, 11):  
    nums.append(n*2)  
  
print(nums)
```

[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

```
print([n for n in range(1, 11) if n % 3 == 0])
```

[3, 6, 9]

# 리스트 관리

## ❖ 삽입

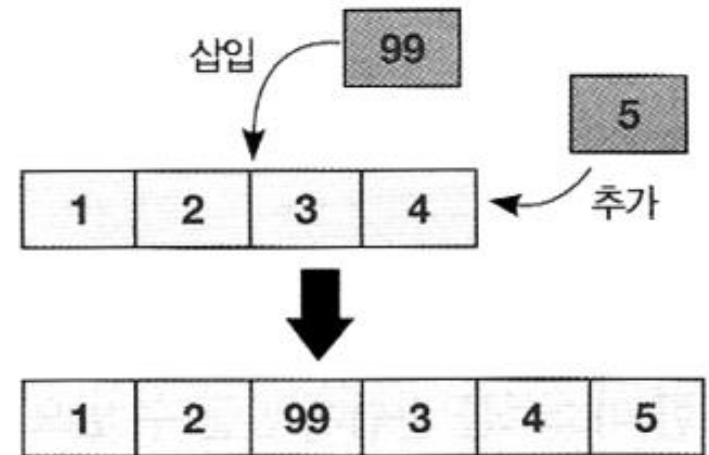
- `.append(값)`
  - 리스트의 끝에 값을 추가
- `.insert(위치, 값)`
  - 지정한 위치에 값을 삽입

```
nums = [1, 2, 3, 4]
nums.append(5)
print(nums)
```

```
nums.insert(2, 99)
print(nums)
```

```
[1, 2, 3, 4, 5]
```

```
[1, 2, 99, 3, 4, 5]
```



# 리스트 관리

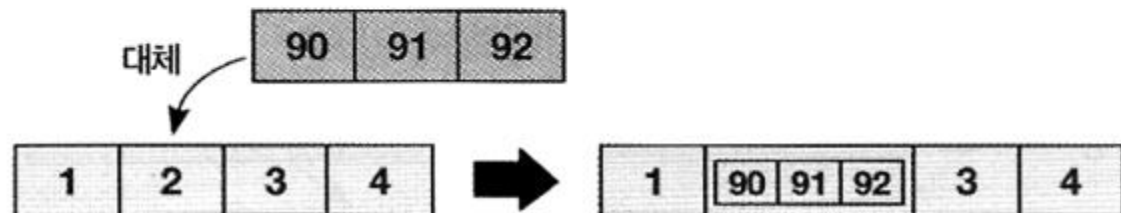
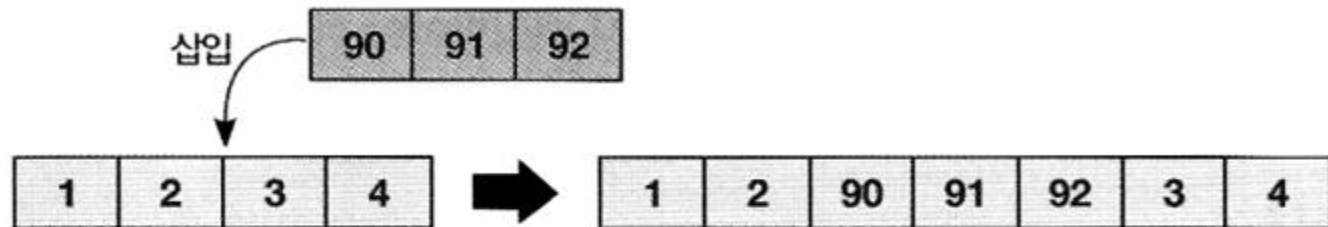
## ❖ 삽입

```
nums = [1, 2, 3, 4]
nums[2:2] = [90, 91, 92] # 새로운 값들을 삽입
print(nums)
```

[1, 2, 90, 91, 92, 3, 4]

```
nums = [1, 2, 3, 4]
nums[2] = [90, 91, 92] # 지정한 위치의 엘리먼트에 리스트 대체
print(nums)
```

[1, 2, [90, 91, 92], 4]



# 리스트 관리

---

## ❖ 리스트 연결

- 리스트1.extend(리스트2)
- 리스트1 = 리스트1 + 리스트2

```
list1 = [1, 2, 3, 4, 5]
list2 = [10, 20, 30]
list1.extend(list2)
print(list1)
```

[1, 2, 3, 4, 5, 10, 20, 30]

# 리스트 관리

---

## ❖ 리스트 연결

```
list1 = [1, 2, 3, 4, 5]
list2 = [10, 11]
list3 = list1 + list2  # 새로운 리스트를 리턴
print(list3)
list1.extend(list2)    # 기존 리스트를 확장
print(list1)
```

```
[1, 2, 3, 4, 5, 10, 11]
```

```
[1, 2, 3, 4, 5, 10, 11]
```



# 리스트 관리

---

## ❖ 삭제

- `.remove(값)`
  - 리스트에서 값을 찾아 첫번째 해당 요소를 제거
- `del(리스트[인덱스])`
  - 지정한 인덱스의 요소를 제거
- `[시작:끝] = []`
  - 지정한 범위의 요소를 제거

# 리스트 관리

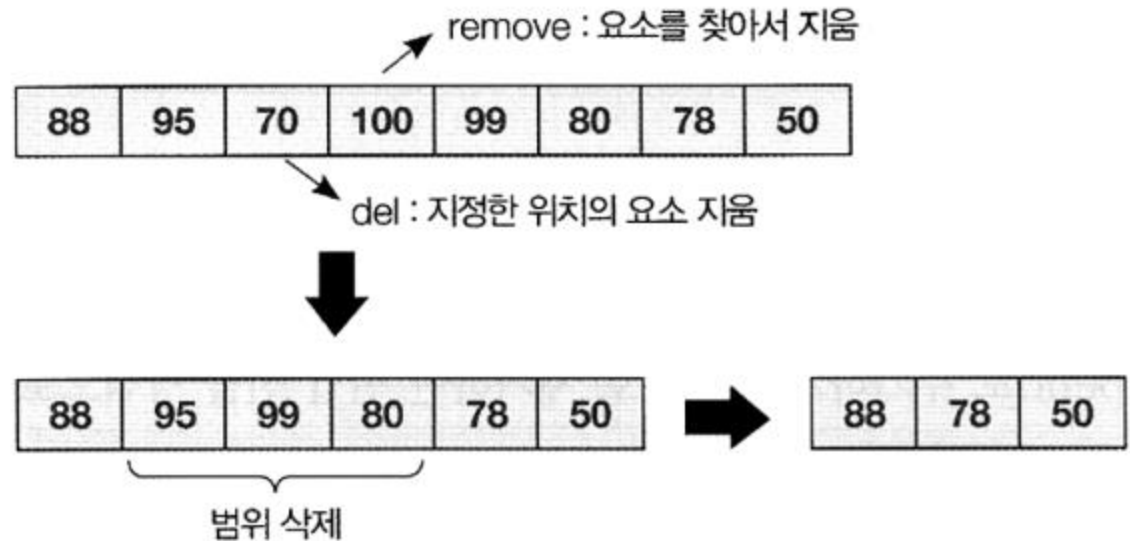
## ❖ 삭제

```
score = [88, 95, 70, 100, 99, 88, 78, 50]
score.remove(100)
print(score)
```

```
del(score[2])
print(score)
```

```
score[1:4] = []
print(score)
```

```
[88, 95, 70, 99, 88, 78, 50]
[88, 95, 99, 88, 78, 50]
[88, 78, 50]
```



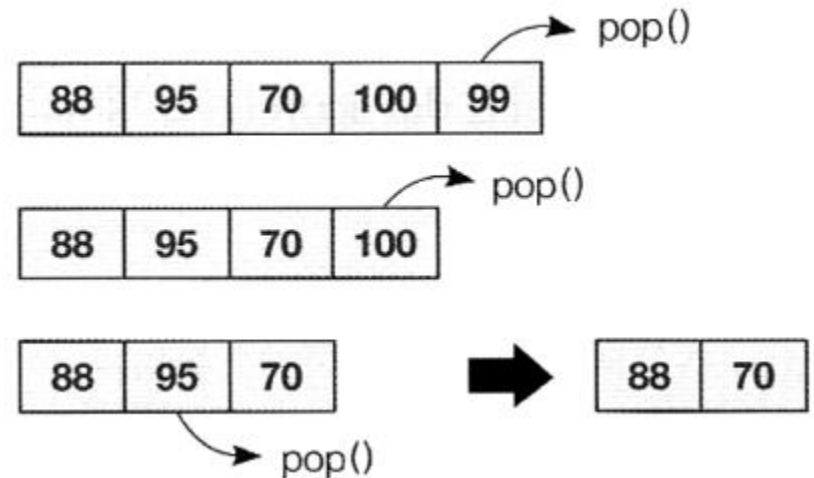
# 리스트 관리

## ❖ 삭제

- `.pop()`
  - 리스트의 끝 요소를 삭제하고 삭제한 요소를 리턴
- `.pop(인덱스)`
  - 지정한 인덱스의 끝 요소를 삭제하고 삭제한 요소를 리턴

```
score = [88, 95, 70, 100, 99]
print(score.pop())
print(score.pop())
print(score.pop(1))
print(score)
```

```
99
100
95
[88, 70]
```



# 리스트 관리

---

## ❖ 삭제

```
score = [88, 95, 70, 100, 99]
score.append(50)
```

```
print('큐', score)
print('큐', score.pop(0))
print('큐', score)
```

```
큐 [88, 95, 70, 100, 99, 50]
큐 88
큐 [95, 70, 100, 99, 50]
```

```
score = [88, 95, 70, 100, 99]
score.append(50)
```

```
print('스택', score)
print('스택', score.pop())
print('스택', score)
```

```
스택 [88, 95, 70, 100, 99, 50]
스택 50
스택 [88, 95, 70, 100, 99]
```

# 리스트 관리

---

## ❖ 검색

- `.index(값)`
  - 지정한 값을 찾아 해당 요소를 리턴, 없으면 예외 발생
- `.count(값)`
  - 지정한 값이 리스트에 몇 번 나오는지 계산하여 리턴

```
score = [88, 95, 70, 100, 99, 88, 78, 50]

perfect = score.index(100)
print("만점 받은 학생은 %d번 입니다."%perfect)

pernum = score.count(100)
print("만점자 수는 %d명 입니다."%pernum)
```

만점 받은 학생은 3번 입니다.  
만점자 수는 1명 입니다

# 리스트 관리

---

## ❖ 검색

- len(시퀀스)
  - 시퀀스의 길이(요소수) 리턴
- max(시퀀스)
  - 시퀀스 요소중 최대값 리턴
- min(시퀀스)
  - 시퀀스 요소중 최소값 리턴

```
score = [88, 95, 70, 100, 99, 88, 78, 50]
print("학생수는 %d명 입니다."%len(score))
print("최고 점수는 %d점 입니다."%max(score))
print("최소 점수는 %d점 입니다."%min(score))
```

학생수는 8명 입니다.  
최고 점수는 100점 입니다.  
최소 점수는 50점 입니다.

# 리스트 관리

---

## ❖ 검색

- 값 in 시퀀스, 값 not in 시퀀스
  - 값이 시퀀스에 포함되어 있는지 여부를 True/False로 리턴

```
ans = input("결제 하시겠습니까? ")  
  
if ans in ['yes', 'y', 'ok', '예']:  
    print("결제를 진행합니다.")  
else:  
    print("결제를 취소합니다")
```

결제 하시겠습니까? y  
결제를 진행합니다.

# 리스트 관리

## ❖ 정렬

- `.sort([reverse=True][key=키에 적용할 함수])`
  - 리스트를 정렬(디폴트는 오름차순), `reverse=True`로 오름차순/내림차순 선택
- `.reverse()`
  - 리스트의 순서를 역으로 바꿈
- `sorted(시퀀스)`
  - 지정한 시퀀스를 정렬하여 새로운 리스트로 리턴

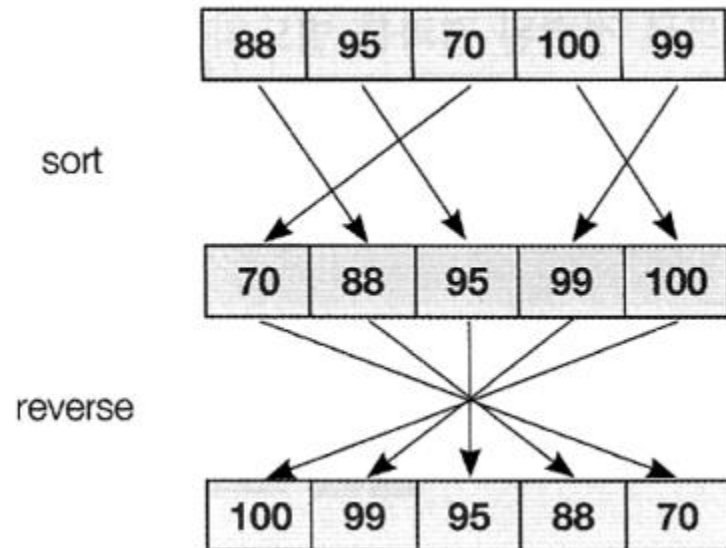
```
score = [88, 95, 70, 100, 99]
```

```
score.sort()  
print(score)
```

```
score.reverse()  
print(score)
```

```
[70, 88, 95, 99, 100]
```

```
[100, 99, 95, 88, 70]
```





# 리스트 관리

---

## ❖ 정렬

```
score = [88, 95, 70, 100, 99]
```

```
score.sort(reverse=True)  
print(score)
```

```
[100, 99, 95, 88, 70]
```

```
country = ["Korea", "japan", "CHINA", "america"]
```

```
country.sort()  
print(country)
```

```
country.sort(key = str.lower)  
print(country)
```

```
['CHINA', 'Korea', 'america', 'japan']
```

```
['america', 'CHINA', 'japan', 'Korea']
```

# 리스트 관리

---

## ❖ 정렬

```
score = [88, 95, 70, 100, 99]
sorted_score = sorted(score)
```

```
print(score)
print(sorted_score)
```

```
[88, 95, 70, 100, 99]
[70, 88, 95, 99, 100]
```

```
score = [88, 95, 70, 100, 99]
sorted_score = sorted(score, reverse=True)
```

```
print(score)
print(sorted_score)
```

```
[88, 95, 70, 100, 99]
[100, 99, 95, 88, 70]
```

# 튜플

---

## ❖ 불변 자료 집합

- (값, ... )
- 값, [...]
- 추가/수정/삭제 불가
- 읽기만 가능 --> 리스트보다 속도가 빠름

```
score = (88, 95, 70, 100, 99)
total = 0

for s in score:
    total += s

print("총점 : ", total)
print("평균 : ", total/len(score))
```

```
총점 : 452
평균 : 90.4
```

# 튜플

---

## ❖ 불변 자료 집합

```
score = 88, 95, 70, 100, 99  
print(score)
```

```
score = 88,  
print(score)
```

```
score = 88  
print(score)
```

```
(88, 95, 70, 100, 99)
```

```
(88,)
```

```
88
```

# 튜플

---

## ❖ 튜플로 가능한 일

```
tu = 1, 2, 3, 4, 5
```

```
print(tu[3])  
print(tu[1:4])  
print(tu + (6, 7))  
print(tu * 2)
```

```
4  
(2, 3, 4)  
(1, 2, 3, 4, 5, 6, 7)  
(1, 2, 3, 4, 5, 1, 2, 3, 4, 5)
```

```
tu[1] = 100 # 불가능  
del tu[1]   # 불가능
```

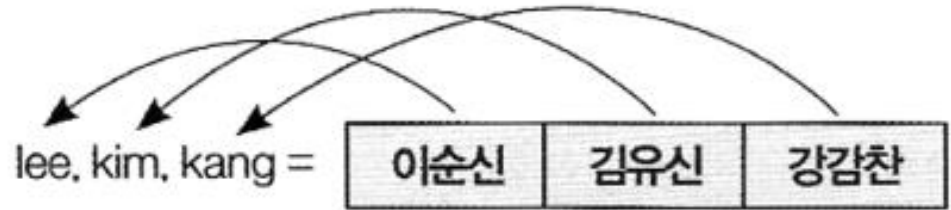
# 튜플

## ❖ 튜플로 가능한 일

```
names = "이순신", "김유신", "강감찬"  
lee, kim, kang = names # unpack
```

```
print(lee)  
print(kim)  
print(kang)
```

이순신  
김유신  
강감찬



```
a, b = 12, 34  
print(a, b)
```

```
a, b = b, a  
print(a, b)
```

12 34  
34 12

# 튜플

---

## ❖ 튜플로 가능한 일

```
import time

def gettime():
    now = time.localtime()
    return now.tm_hour, now.tm_min

result = gettime()
print("지금은 %d시 %d분 입니다."%(result[0], result[1]))

hour, minute = gettime()
print("지금은 %d시 %d분 입니다."%(hour, minute))
```

지금은 20시 29분 입니다.  
지금은 20시 29분 입니다.

# 튜플

---

## ❖ 튜플로 가능한 일

```
d, m = divmod(7, 3)    # return 7//3, 7%3
```

```
print("몫", d)  
print("나머지", m)
```

```
몫 2  
나머지 1
```

```
score = [88, 95, 70, 100, 99]  
tu = tuple(score)  
print(tu)
```

```
li = list(tu)  
li[0] = 100  
print(li)
```

```
(88, 95, 70, 100, 99)  
[100, 95, 70, 100, 99]
```