함수 인수

6-1 인수와 리스트 튜플 언패킹

- 함수 인수
 - ✓ 3개의 매개 변수를 받는 함수를 아래와 같이 정의한다.

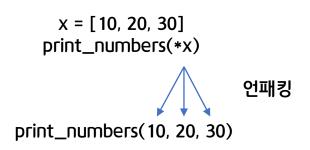
```
def print_nums(a,b,c):
    print(a,b,c)
    print_nums(10,40,70)
```

```
[결과]
10 40 70
```

- 언패킹 활용
 - ✓ 인수를 순서대로 넣을 때는 리스트나 튜플을 이용할 수 있다.
 - ✓ 아래와 같이 리스트 또는 튜플 앞에 *을 붙여서 사용할 수 있다.

```
data1 = [10, 20, 30]
print_nums(data1[0], data1[1], data1[2])
print(*data1)
print_nums(*data1)
```

```
[결과]
10 20 30
10 20 30
10 20 30
```



6-1 인수와 리스트 튜플 언패킹

- 가변 인수 함수
 - ✓ 튜플 언패킹은 인수의 개수가 정해지지 않은 가변 인수(variable argument)에 이용한다.
 - ✓ 아래와 같이 가변 인수 함수는 매개변수 앞에 *을 붙여서 사용한다.

```
def add(*data):
    print(type(data))
    result = 0
    for d in data:
        result += d
    return result

print(add(10,20))
print(add(10,20,30,40))
x = (40,50,70)
print(add(*x))
```

```
[결과]

<class 'tuple'>

30

<class 'tuple'>

100

<class 'tuple'>

160
```

6-2 키워드 인수

- 키워드 인수
 - ✓ 파이썬에서는 인수의 순서와 용도를 매번 기억하지 않도록 키워드 인수(keyword argument)라는 기능을 제공한다.

```
def person_print(name, age, address):
    print('name:',name)
    print('age:',age)
    print('address:',address)

person_print('kim', 30, 'suwon')
    person_print(name='lee', age=40, address='incheon')
    person_print(age=40, address='incheon',name='lee')
```

[결과]

name: kim

age: 30

address: suwon

name: lee age: 40

address: incheon

name: lee

age: 40

address: incheon

6-2 키워드 인수

- 키워드 인수와 딕셔너리 언패킹
 - ✓ 딕셔너리에 '키워드': 값 형식으로 인수를 저장하고, 앞에 **를 붙여서 함수에 넣는다.
 - ✓ 딕셔너리의 키워드(키)는 반드시 문자열이여야 한다.

```
data1 = {'name':'홍길동', 'age':30, 'address':'서울시'}
person_print(data1['name'], data1['age'], data1['address'])
person_print(name=data1['name'], age=data1['age'], address=data1['address'])
person_print(**data1)
```

[결과]

name: 홍길동

age: 30

address: 서울시

name: 홍길동

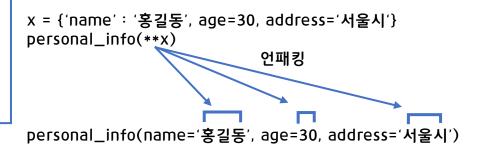
age: 30

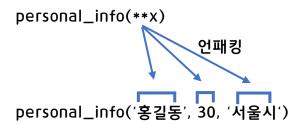
address: 서울시

name: 홍길동

age: 30

address: 서울시





6-2 키워드 인수

- 키워드 인수와 가변 인수 함수
 - ✓ 키워드 인수를 사용하는 가변 인수 함수는 매개변수 앞에 **를 붙여서 만들 수 있다.
 - ✓ 매개변수 이름은 원하는 대로 지어도 되지만 관례적으로 keyword arguments를 줄여서 kwargs로 사용한다.

```
def city_info(**data):
    for k, v in data.items():
        print(k,':',v)
    print()

city_info(seoul=90000, incheon=30230, busan=56222)
city_info(seoul=90000, busan=56222)
data2 = {'seoul':90000, 'suwon':32523, 'incheon':30230, 'busan':56222}
city_info(**data2)
```

[결과]

seoul : 90000 incheon : 30230 busan : 56222

seoul : 90000 busan : 56222

seoul : 90000 suwon : 32523 incheon : 30230 busan : 56222

6-3 디폴트 매개 변수

- 디폴트 매개 변수
 - ✓ 매개변수를 정의할 때 값을 할당해 놓으면 디폴트 매개 변수가 된다.

```
def person_info(name, age=40, address='seoul'):
    print('name:',name)
    print('age:',age)
    print('address:',address)

person_info('kim', 50, 'busan')
person_info('lee')
person_info(age=20, name='lee')
```

[결과]

name: kim

age: 50

address: busan

name: lee

age: 40

address: seoul

name: lee

age: 20

address: seoul

6-3 디폴트 매개 변수

- 디폴트 매개 변수
 - ✓ 디폴트 매개변수를 지정할 때 한 가지 주의할 점은 초깃값이 지정된 매개변수 다음에는 초깃값이 없는 매개변수 앞에 올 수 없다.

```
def person_info(name, age=40, address):
    print('name:',name)
    print('age:',age)
    print('address:',address)
```

[결과]

SyntaxError: non-default argument follows default argument

- ✓ 위와 같이 함수를 만들면 문법 에러가 발생한다.
- ✓ 디폴트 매개 변수를 구성할 때는 뒤쪽에 몰아주어 구성한다.

```
def person_info(name, age, address='비공개'):
def person_info(name, age=0, address='비공개'):
def person_info(name='아무개', age=0, address='비공개'):
```