

# 결론 및 요약

함수의 파라미터 순서: 일반 변수, \*변수, \*\*변수

\*변수 -> 여러개가 아규먼트로 들어올 때, 함수 내부에서는 해당 변수를 '튜플'로 처리한다.

\*\*변수 -> "키워드=" 로 입력할 경우에 그것을 각각 키와 값으로 가져오는 '딕셔너리'로 처리한다.

이 상태에서 name과 \*blogs의 순서를 바꾸면 어떻게 될까요?

```
In [7]: def blog_printer(*name, blogs):
        print(name)
        for post in blogs:
            print(post)

        주인장이름 = '내 이름은 블로그 주인'
        블로그1 = '첫번째 블로그지름'
        블로그2 = '성공해서 두번째 블로그도 만들었지롱'
        블로그3 = '세번째도 만들었지롱'

        blog_printer(주인이름, 블로그1, 블로그2, 블로그3)

TypeError                                 Traceback (most recent call last)
<ipython-input-7-e5d5b1fe30d5> in <module>
      9 블로그3 = '세번째도 만들었지롱'
     10
--> 11 blog_printer(주인이름, 블로그1, 블로그2, 블로그3)

TypeError: blog_printer() missing 1 required keyword-only argument: 'blogs'
```

영? 순서만 바꿨는데 오류가 난다?

네 오류가 납니다.

\*args는 일반 변수보다는 뒤에 위치해있어야 합니다.

파이썬은 어디서부터 어디까지를 \*변수에 담을지 알 수 없습니다.

그래서 맨 앞에 특정 변수를 명시해두고, 그 뒤에는 \*args로 아규먼트를 넣어줘야합니다.

```
In [11]: def blog_printer(name, *blogs, **blog_benefits):
...
    name: 블로그 주인장의 이름
    *blogs: 블로그를 만들 때 설명
    **blog_benefits: 해당 블로그의 수익
...

    print(name)
    for post in blogs:
        print(post)
    for blog, benefits in blog_benefits.items():
        print(blog, '수익은 >>', benefits)

주인장이름 = '블로그 주인'
블로그1 = '첫번째 블로그지롱'
블로그2 = '성공해서 두번째 블로그도 만들었지롱'
블로그3 = '세번째도 만들었지롱'

help(blog_printer)
blog_printer(주인장이름, 블로그1, 블로그2, 블로그3, 블로그수익1=30, 블로그수익2=40, 블로그수익3=100)

Help on function blog_printer in module __main__:

blog_printer(name, *blogs, **blog_benefits)
    name: 블로그 주인장의 이름
    *blogs: 블로그를 만들 때 설명
    **blog_benefits: 해당 블로그의 수익

블로그 주인
첫번째 블로그지롱
성공해서 두번째 블로그도 만들었지롱
세번째도 만들었지롱
블로그수익1 수익은 >> 30
블로그수익2 수익은 >> 40
블로그수익3 수익은 >> 100
```

함수명 바로 아래에 다중 문자열("")을 넣어서 함수의 docstring으로 만들어 줍니다.

저렇게 함수에다가 docstring을 넣으면 help()함수로 함수의 설명을 읽을 수 있습니다.

\*blogs에는 블로그1, 블로그2, 블로그3을 아규먼트로 넣었고,  
\*\*blogs\_benefits는 블로그수익1, 블로그수익2, 블로그수익3 으로 각각 키워드를 넣어서 입력했습니다.

보다시피 \*blogs , \*\*blogs\_benefits를 순서대로 넣은 것을 볼 수 있습니다.