

# LLM 활용 인공지능 서비스 개발자 양성과정

도봉 SeSAC 캠퍼스 X **Saltlux**

강사 최동혁

# Unpacking

- 튜플이나 리스트와 같은 시퀀스 자료형의 요소들을 여러 변수에 나누어 할당하는 것

```
# 튜플 생성  
my_tuple = (1, 2, 3)
```

```
# 언팩킹  
a, b, c = my_tuple
```

```
print(a) # 1  
print(b) # 2  
print(c) # 3
```

# 확장된 Unpacking

- \* 연산자를 이용한 확장된 Unpacking
- 이 경우, first 변수는 튜플의 첫 번째 요소를 받고, rest는 나머지 모든 요소를 리스트로 받는다.

```
# 튜플 생성
my_tuple = (1, 2, 3, 4, 5)

# 첫 번째 요소와 나머지를 언팩킹
first, *rest = my_tuple

print(first) # 1
print(rest)  # [2, 3, 4, 5]
```

## 중첩된 튜플 언팩킹

튜플 내에 또 다른 튜플이 있는 경우, 중첩된 구조로 언팩킹할 수 있다.

```
# 중첩된 튜플
nested_tuple = (1, (2, 3), 4)

# 중첩된 구조로 언팩킹
a, (b, c), d = nested_tuple

print(a) # 1
print(b) # 2
print(c) # 3
print(d) # 4
```

## 가변인자

- 함수가 임의의 개수의 인자를 받아들일 수 있도록 도와줌.
- `*args`와 `**kwargs`를 사용하여 가변 인자를 구현할 수 있다.
- `*args`는 여러 개의 위치 인자를 튜플로 받고,
- `**kwargs`는 키워드 인자를 딕셔너리로 받는다.

*\*args*

```
def sum_numbers(*args):  
    return sum(args)  
print(sum_numbers(1, 2, 3))    # 6  
print(sum_numbers(1, 2, 3, 4, 5))  # 15
```

## 가변인자

*\*\*kwargs*

```
def print_info(**kwargs):  
    for key, value in kwargs.items():  
        print(f"{key}: {value}")  
  
print_info(name="Alice", age=30)
```

- print\_info 함수는 키워드 인자들을 받아서 출력
- \*\*kwargs는 함수에 전달된 키워드 인자들을 딕셔너리로 저장

## 가변 인자와 튜플 언팩킹

- 가변 인자와 튜플 언팩킹을 함께 사용하면, 함수로 전달된 여러 인자를 한 번에 처리할 수 있다.

```
def multiply(*args):  
    product = 1  
    for num in args:  
        product *= num  
    return product  
numbers = (2, 3, 4)  
print(multiply(*numbers)) # 24
```

- multiply 함수는 여러 숫자들의 곱을 반환한다.
- \*numbers를 사용하여 numbers 튜플의 각 요소를 함수의 인자로 언팩킹

- JupyterHub
- 앞으로 수업 계획