

1. 함수와 메소드 차이

- 객체에 속해 있는지 여부
- 결국 큰 의미로 보면 메소드는 함수

함수

- 특정 작업을 수행하는 코드블럭.
- 독립된 기능을 수행

사용법: `output = func(input)`

메소드

- 객체와 연관되어 사용. (보통 메소드는 객체에 속해있는 함수라고 말함)
- 클래스에 묶여서 클래스의 인스턴스와 관계되는 일을 하는 함수.
- 클래스, 구조체 등에 포함되어 있는 함수.
- 다른 말로 `클래스 함수` 라고도 함.(중요하진 않음)

사용법: `object.method()`

2. 그럼 왜 괄호는 쓰는거죠?

- 파이썬 인터프리터에 다음과 같이 작성해보세요.

```
>>> help(list)
```

```
# list 클래스가 어떻게 구성되어 있고 가지고 있는 함수(메소드)들도 볼 수 있는데 이 메소드들에 들어가는 인자(매개변수)들을 확인해보세요. 대표적으로 None 을 호출하는 sort()를 살펴볼까요?
```

```
| sort(...)  
|     L.sort(key=None, reverse=False) -> None -- stable sort *IN PLACE*
```

```
# 보시면 정렬 후 리턴은 None 으로 한다는 듯한 문구가 적혀있습니다.
```

```
# 이외에도 괄호에 아무것도 안 넣을 때 어떤 값이 들어가고 리턴되는지 등등 한번 보시면 아! 하실 거예요.
```

- 반대로 함수인 `sorted()` 는 이렇게도 입력해보세요.

```
>>> type(sorted)
```

```
<class 'builtin_function_or_method'>
```

```
>>> help(sorted)
Help on built-in function sorted in module builtins:

sorted(iterable, /, *, key=None, reverse=False)
    Return a new list containing all items from the iterable in ascending
    order.

    A custom key function can be supplied to customize the sort order, and
    the
    reverse flag can be set to request the result in descending order.

# Return a new list 구문 보이시나요?
# 이처럼 다양하게 확인해 보시면 어떤 메소드는 None 을 리턴하고 어떤 메소드는 변수를 변경한 값을 리
턴하고를
# 확인할 수 있습니다. 이제 함수나 메소드나 모두 괄호가 있어야 하는 이유를 조금 아시겠나요?
```

3. 결론

```
>>> str = 3
>>> str('1')
TypeError: 'int' object is not callable
# int 객체는 call, 즉 () 로 호출 할 수 없기 때문
```

```
iu.name      #=> 변수 자체 리턴

iu.greeting  #=> 함수 자체 리턴

iu.greeting() #=> 함수 호출
```

돌아와서 정리하자면 괄호를 쓰고 안 쓰기의 차이는

함수 호출(call)은 괄호가 있을 때 발생

함수 자체만 쓰면 그건 함수의 객체

내부적으로 궁금하신 것 같아 준비했지만 사실 프로그래밍 언어가 가지고 있는 규칙 정도라고 생각하시는 것도 좋습니
다.

함수의 괄호라는 것 자체가 output 을 위한 input box 라고 생각하시는 것도 좋을 것 같아요.

파이썬 일급 객체 함수 라고 구글링 하셔서 공부하시면 더 이해가 쉬우실겁니다.

```
# 충격적인(?) 파이썬의 일급 객체 함수 예제
>>> s = sum
>>> s(range(4))
>>> ???
```

파이썬은 모든 것이 객체입니다 !