

`__init__` is not the only constructor

`__new__(cls[, ...])`

- "controls object creation."
- 클래스 cls 의 새 인스턴스를 만들기 위해 호출됩니다.
- 스택 메서드입니다.
- 암시적으로 첫 번째 인자에는 만들려고 하는 인스턴스의 클래스(cls)가 전달되며, 해당 클래스의 인스턴스를 생성합니다.
- `__init__` 동작 전에 `__new__` 메서드가 호출되는데, `__init__` 이 호출되기 전에 `__new__` 가 메모리를 할당하기 때문입니다.
- 주로 불변형(int, str, tuple과 같은)의 서브 클래스가 인스턴스 생성을 커스터마이징할 수 있도록 하는 데 사용됩니다. 또한, 사용자 정의 메타 클래스에서 클래스 생성을 커스터마이징(controls object creation) 하기 위해 자주 사용됩니다.

`__init__(self[, ...])`

- (`__new__()` 에 의해) 인스턴스가 만들어진 후 호출되는 함수.
- 객체에서 사용할 초기 값들을 초기화 (즉, 새로 초기화 된 인스턴스를 얻을 수 있다.)
- 객체를 초기화하고 메모리를 할당하지 않으면 즉시 호출됩니다.

"Because **new()** and **init()** work together in constructing objects. (**new()** to create it, and **init()** to customize it)" - Python 공식문서

파이썬 객체의 구성은 단순히 새로운 인스턴스를 생성(할당)(`__new__`) 한 다음 해당 인스턴스를 초기화(`__init__`)하는 것

사용 예시

```
class Person:

    def __init__(self):
        print('init')
        super().__init__()

    def __new__(cls):
        print('new')
        return super().__new__(cls)

user = Person()
```

```
# ouput
```

```
new  
init
```

Summary

- 끝없는 논쟁이지만 이 논쟁의 중심은 python 의 `__init__` 이 Java, C#, C++ 와 같은 언어들의 생성자(이하 constructor)가 하는 것과 거의 같은 종류의 작업을 수행하기 때문입니다.
- 파이썬 공식 문서에서는 `__new__` 와 `__init__` 에 대해 어느 것도 **constructor** 라고 정의 내리고 있지 않습니다.
- 다만, `Foo()` 와 같이 객체를 만드는데 사용 되는 호출(the call to the class)을 `constructor expression` 이라고만 언급합니다.
- 그리고 공식문서는 constructing objects 를 위해 `__new__` 와 `__init__` 가 협력(new 는 create, init 은 customize)하고 있다고 표현하는 것으로 보아하니 `constructor = create + customize` 의 개념으로 전달하고자 합니다.
- 기술적으로는 `__init__` 이 호출될 때까지 객체가 이미 생성되어 있으며, 클래스의 새 인스턴스에 대한 유효한 참조가 이미 있기 때문에 constructor 라 부를 수 없습니다.
- 하지만 우리는 대부분 `__init__` 을 중심으로 사용하고 있고, (왜냐하면 우리가 할당에 신경을 쓰지 않으니) 일반적인 대화에서는 `__init__` 을 constructor 라고 불러도 크게 문제는 없습니다. 우리는 파이썬에서 메모리에 관해 크게 신경 쓰지 않아도 되기 때문입니다. (그래도 기술 서적과 같은 문서에서 constructor 라고 정의 해버리는 건 위험.)

참고문헌

<https://docs.python.org/3/reference/datamodel.html#basic-customization>

<https://dev.to/delta456/python-init-is-not-a-constructor-12on>

<https://stackoverflow.com/questions/6578487/init-as-a-constructor>

<https://diveinto.org/python3/iterators.html#init-method>