

파이썬 프로그래밍

클래스와 객체



한국기술교육대학교
온라인평생교육원

■ 생성자와 소멸자

1. 생성자와 소멸자의 메소드

- `__init__`: 생성자 메소드

객체가 생성될 때 자동으로 불리어지는 메소드

`self` 인자가 정의되어야 함

`__del__`: 소멸자 메소드

객체가 소멸 (메모리에서 해제)될 때 자동으로 불리어지는 메소드

`self` 인자가 정의되어야 함

개발자가 특별히 작성하지 않아도 될 메소드

이유: 파이썬에서는 메모리나 기타 자원들의 해제가 자동으로 되기 때문에

[참고] `__` (연속된 두 개의 언더라인)의 의미: 예약된 이름

다음 코드에 대한 설명

`mylife = Life()` 로서 인스턴스 `mylife`가 생성되는 순간 `__init__` 생성자 메소드 호출

`sleep(3)`에 의해 3초간 `sleep` 상태

3초 이후 함수가 리턴됨 --> 로컬 변수가 메모리에서 해제됨

--> `__del__` 소멸자 메소드 호출

- 생성자 메소드는 많이 정의가 됨
- `'__'`로 시작 → 예약된 이름들 (직접 다시 정의할 필요 X)

■ 생성자와 소멸자

1. 생성자와 소멸자의 메소드

```
# -*- coding:utf-8 -*-
from time import ctime, sleep

class Life:
    def __init__(self):      # 생성자
        self.birh = ctime()  # 현재시간에 대한 문자열을 얻는다.
        print 'Birthday', self.birh # 현재 시간 출력
    def __del__(self):      # 소멸자
        print 'Deathday', ctime() # 소멸 시간 출력

def test():
    mylife = Life()
    print 'Sleeping for 3 sec'
    sleep(3) #3초간 sleep(block)상태에 있음 (CPU 점유 못함)

test()
```

```
Birthday Fri Dec  5 11:06:36 2014
Sleeping for 3 sec
Deathday Fri Dec  5 11:06:39 2014
```

- ctime의 c는 current → 현재시간
- 생성자는 self 부분에 생성하고자 하는 객체 들어감
- 리턴이 없어도 객체가 생성되어 mylife에 할당됨
- 소멸자가 호출되어지는 코드가 명시적으로 존재하지 X
- 함수가 끝나면, 함수 안에 정의되어진 객체는 자동으로 없어짐
- 소멸되기 직전에 del이라는 함수가 수행되는 것
- 프로그램이 끝난 것을 확인하기 전까지 계속 수행

■ 생성자와 소멸자

1. 생성자와 소멸자의 메소드

- 인자를 받는 생성자 호출 가능
- [참고] `__str__`: print 예약어나 `str()` 내장함수 호출에 대응되는 메소드

```
class Integer:
    def __init__(self, i):
        self.i = i
    def __str__(self):
        return str(self.i)
```

```
i = Integer(10)
print i
print str(i)
```

```
10
10
```

- 생성자를 정의할 때 인자를 `self` 말고 인자 하나 더 삽입
- '`__str__`' 메소드 → 프린트 `i`, 내장함수 `str()` 사용 시 `str` 메소드 자동 호출