

1. 매직 메소드

-> Python에서 \_\_function\_\_ 처럼 double underscore로 둘러싸인 함수

ex)

```
var_1 = 'hello'
var_2 = 'world'
var_1 + var_2 = var1.__add__(var2)
-> 매직 메소드가 호출된 것이다.
```

ex) 매직 메소드 예시

- 1) 'abc' in var = var.\_\_contains\_\_('abc')
- 2) var == 'abc' = var.\_\_eq\_\_('abc')
- 3) var[1] = var.\_\_getitem\_\_(1)
- 4) var[1:3] = var.\_\_getslice\_\_(1,3)
- 5) len(var) = var.\_\_len\_\_()
- 6) print(var) = var.\_\_repr\_\_()

2. with 구문

- 특정 object는 끝났을 때 닫아야 한다.
  - File object
  - Network Connection
- > with 구문은 끝났을 때 객체를 닫아준다.

ex)

```
with open('filename.txt') as fh:
    for line in fh:
        line = line.strip()
        print(line)
print('done')
```

3. 예외 처리

- 1) 하나의 except 문으로 다양한 에러를 처리할 수 있다.

```
try:
    arg = sys.argv[1]
    num = int(arg)
except(IndexError, ValueError):
    sys.exit('Error')
```

- 2) Exception Handling이 아름다운 이유는호출 stack의 어디에서도 Error을 처리할 수 있다는 점이다.

```
try:
    main()
except ZeroDivisionError:
    print('Oops!')
```

- 3) 에러에 대해서 자세한 설명을 해야 할 경우 우리가 직접 Error 메세지와 함께 낼 수 있다.

```
try:
    formatted_line = delim.join(list_to_join)
except TypeError:
    raise TypeError("make_delim_line(): arg 1 must be a list")
return formatted_line
```