

# Ask Django

호출 가능한 (**Callable**) 객체

# 호출가능한 객체

## 호출문법 : **obj()**

- 함수를 호출해서, 리턴값을 취한다.

```
>>> def mysum(x, y):  
        return x + y
```

```
>>> mysum(1, 2)  
3
```

- 클래스를 호출해서, 인스턴스를 생성한다.
- 하지만, 클래스의 인스턴스를 호출할 수는 없다.

```
class Calculator(object):  
    def __init__(self, base):  
        self.base = base
```

```
calculator = Calculator(10)  
print(calculator(1, 2)) # TypeError: 'Calculator' object is not callable.
```

# 인스턴스를 호출가능토록 만들기

- 인스턴스를 호출가능토록 만들려면 **\_\_call\_\_** 멤버함수를 구현
  - 인스턴스를 호출하면, 파이썬에서 멤버함수 **\_\_call\_\_** 를 호출해줍니다.

```
class Calculator(object):
    def __init__(self, base):
        self.base = base

    def __call__(self, x, y):
        return self.base + x + y
```

```
>>> calculator = Calculator(10)
>>> calculator(1, 2)          # 실제로 calculator.__call__(1, 2) 이 호출됩니다.
13
>>> calculator.__call__(1, 2)
13
```

## 예) 상태값을 유지하는 함수

```
class Calculator(object):  
    def __init__(self, base):  
        self.base = base  
  
    def __call__(self, x, y):  
        self.base += (x + y)  
        return self.base
```

```
>>> calc = Calculator(10)  
>>> calc(1, 2)  
13  
>>> calc(1, 2)  
16  
>>> calc(1, 2)  
19  
>>> calc(1, 2)  
22
```

## 예) 기존 함수와 비슷한 로직의 함수를 만들려면,

```
import requests
from bs4 import BeautifulSoup
from collections import Counter

def word_count(url):
    # 문자열 수집
    html = requests.get(url).text

    # 단어 리스트로 변환
    soup = BeautifulSoup(html, 'html.parser')
    words = soup.text.split()

    # 단어수 카운트
    counter = Counter(words)

    # 통계 출력
    return counter

>>> word_count('https://nomade.kr/vod/')
```

# 함수는 새로 구현할 수 밖에 없어요.

위 **word\_count** 함수를 그대로 복사해서, 해당 루틴만 새로 구현

```
import re
import requests
from bs4 import BeautifulSoup
from collections import Counter

def korean_word_count(url):
    # 문자열 수집
    html = requests.get(url).text

    # 단어 리스트로 변환
    soup = BeautifulSoup(html, 'html.parser')
    words = soup.text.split()

    # 한글 단어만 추출
    words = [word for word in words if re.match(r'^[ㄱ-힣]+$', word)] # 이 코드만 추가

    # 단어수 카운트
    counter = Counter(words)

    # 통계 출력
    return counter

>>> korean_word_count('https://nomade.kr/vod/')
```

# 비효율

코드 중복이 심해요.

# 단어수 세기 (클래스 버전)

클래스지만, 함수처럼 호출해서 쓸 수있어요.

```
import requests
from bs4 import BeautifulSoup
from collections import Counter

class WordCount(object):
    def get_text(self, url):
        '문자열 수집'
        html = requests.get(url).text
        soup = BeautifulSoup(html, 'html.parser')
        return soup.text

    def get_list(self, text):
        '단어 리스트로 변환'
        return text.split()

    def __call__(self, url):
        text = self.get_text(url)
        words = self.get_list(text)
        counter = Counter(words)
        return counter

>>> word_count = WordCount()
>>> word_count('https://nomade.kr/vod/')
```



# 한글 단어수 세기 (클래스 버전)

상속과 오버라이딩을 통해, 주요코드만 재정의

```
import re

class KoreanWordCount(WordCount):
    def get_list(self, text):
        '한글로만 구성된 단어만 추출'
        words = text.split()
        return [word for word in words if re.match(r'^[ㄱ-힣]+$ ', word)] # list comprehension 문법

>>> korean_word_count = KoreanWordCount()
>>> korean_word_count('https://nomade.kr/vod/')
```



*Life is short,  
use Python3/Django.*