

장고 차근차근 시작하기

Second Edition

당신의 파이썬/장고 페이스메이커가 되겠습니다. ;)

EP27. 개발환경에서 static 캐싱 무효화하기

Private browser caches (참고: [MDN HTTP Caching](#))

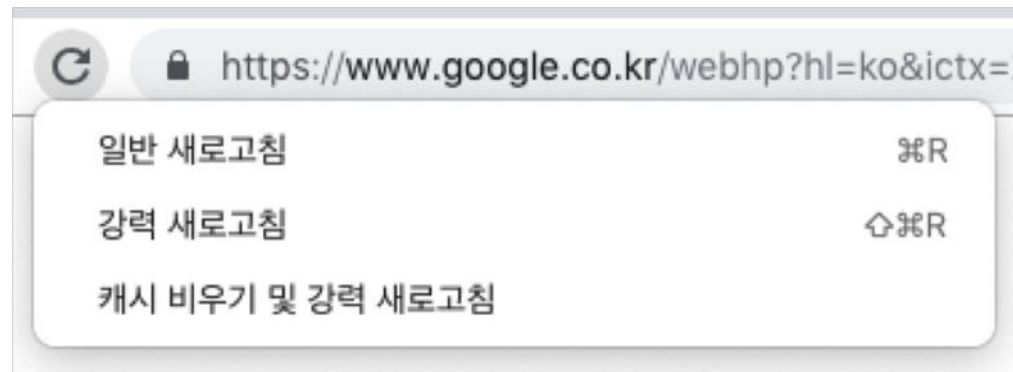
- 브라우저 단에서는 콘텐츠를 캐싱하여, 매번 서버로 콘텐츠를 요청하지 않고 캐싱된 콘텐츠를 사용함으로써, 페이지 렌더링 시간을 단축시킵니다.
- 캐싱 Key : 요청 URL
- 캐싱 만료 정책 : Cache-Control 헤더 (참고: [MDN Cache-Control](#))

개발할 때에는, 실재없이 변경되는 ~

- 하지만, 종종 이전 내역이 브라우저 캐싱되어, 변경된 내역이 반영되지 않기도 하죠.
- 이때, 변경된 내역이 반영되게 할려면?
 - 방법1) 브라우저의 캐시 내역을 강제로 비우기.
 - 크롬 브라우저의 "강력 새로 고침"
 - 방법2) 해당 정적 파일 응답에서 Cache-Control 헤더 조절하기
 - 방법3) 해당 정적 파일의 파일명을 변경
 - 방법4) 해당 정적 파일, 요청 URL에 대해 Dummy QueryString을 추가
 - Query String 값이 변경되면, 브라우저에서는 새로운 리소스로 인식합니다.
 - 웹프론트엔드에서 같은 URL로 Ajax 요청시마다 dummy QueryString을 붙이는 것과 같은 이치

크롬 브라우저의 "강력 새로고침"

- 단축키
 - 윈도우 : Ctrl+Shift+R
 - 맥 : Command+Shift+R
- 단축키를 사용하지 않고, 하기
 - 개발자도구를 연 후에, 새로고침 아이콘에서 우클릭



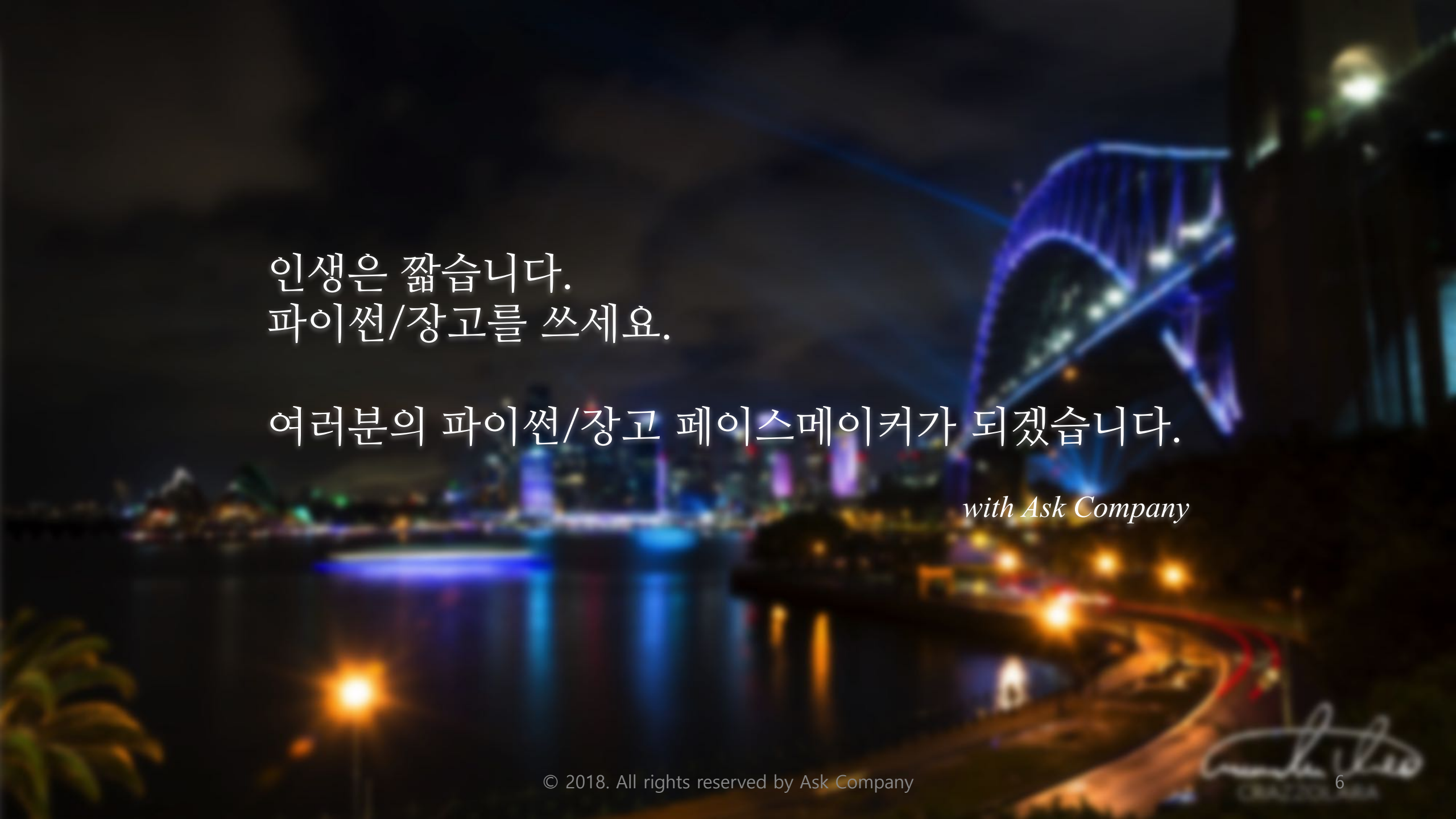
커스텀 Template Tag를 통해, Dummy Query String을 붙여봅시다.

```
from time import time
from django import template
from django.conf import settings
from django.template.tags.static import StaticNode

register = template.Library()

class FreshStaticNode(StaticNode):
    def url(self, context):
        url = super().url(context)
        if settings.DEBUG:
            url += '?_={}'.format(int(time())) # Dummy Query String 추가
        return url

@register.tag('fresh_static')
def do_static(parser, token):
    return FreshStaticNode.handle_token(parser, token)
```

A nighttime photograph of a cityscape featuring a bridge with blue and white lights. The lights are reflected in the water below. The sky is dark, and the overall scene is illuminated by the city lights.

인생은 짧습니다.
파이썬/장고를 쓰세요.

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

with Ask Company