

# 장고 차근차근 시작하기

Second Edition

당신의 파이썬/장고 페이스메이커가 되겠습니다. ;)

EP23. 자주 사용하는 템플릿 필터

# 장고 템플릿 필터 [\(장고 공식문서\)](#)

- 함수 형태로 구현하여, 템플릿에 등록
- 언제 사용하는가?
  - 템플릿 단에서 출력된 값에 대해서, **값 변환**이 필요할 때
  - ex) 개행 적용, 숫자에 콤마찍기, 소스코드 highlight 등
- 필터에서 취하는 인자 (1개~2개)
  - 인자 A : 변환할 값
  - 인자 B (옵션) : 추가 옵션

# add

정수 뿐만 아니라, 다양한 같은 타입에 대한 + 연산 지원

```
{{ value|add:"2" }}
```

```
# django/template/defaultfilters.py#658
@register.filter(is_safe=False)
def add(value, arg):
    """Add the arg to the value."""
    try:
        return int(value) + int(arg)
    except (ValueError, TypeError):
        try:
            return value + arg
        except Exception:
            return ''
```

# cut

정수 뿐만 아니라, 다양한 같은 타입에 대한 + 연산 지원

{{ value|cut:" " }}

```
# django/template/defaultfilters.py#378
@register.filter
@stringfilter
def cut(value, arg):
    """Remove all values of arg from the given string."""
    safe = isinstance(value, SafeData)
    value = value.replace(arg, '')
    if safe and arg != ';':
        return mark_safe(value)
    return value
```

# 유용한 값 처리

- [default](#) : 값이 False판정일 때, 인자로 지정한 디폴트값을 사용.
  - [default if none](#) : 값이 None판정일 때, 인자로 지정한 디폴트값을 사용
- [filesizeformat](#) : 숫자를 파일크기로서 단위를 붙임.
  - 지원 단위 : KB, MB, GB, TB, PB
  - ex) 123456789 → 117.7MB
- [join](#) : 문자열 join과 유사
  - 언제리스트를 하나로 합쳐서 표현하고자 할 때
- [linebreaks](#) : 1개 개행은 <br>태그, 2개 개행은 <p>태그로 변환
  - 줄바꿈 시에 유용
- [linebreaksbr](#) : 모든 개행을 <br>태그로 변환
- [pprint](#) : [pprint.pprint\(\)](#) 래핑. 디버깅 목적의 출력.
- [truncatechars](#) : 지정 글자수 만큼을 자르고, 말줄임표(...)를 붙임.
  - [truncatechars\\_html](#) : 글자 단위로 html 요소를 살려서 자르기
  - [truncatewords](#) : 단어 단위로 자르기
  - [truncatewords\\_html](#) : 단어 단위로 html 요소를 살려서 자르

# URL 링크 만들기

- [urlize](#) : URL과 이메일을 a 태그로 변환
  - URL은 링크 문자열이 필히 필히 http://, https://, www. 로 시작해야함.
  - 최상위 도메인이 .com, .edu, .gov, .int, .mil, .net, .org 중 하나여야함.
  - 생성된 링크에 rel="nofollow"가 추가 → 크롤러에게 추적방지 요청
  - 보다 유연한 링크 변환을 위해, javascript 라이브러리를 사용해볼 수도.
- [urlizetrunc](#)
  - urlize와 동일하지만, 링크 텍스트를 지정 길이로 자르기

# escape 처리와 SafeText

- **escape** : 문자열의 HTML요소를 변환 ⇔ safe
  - 이를 통해 사이트 개발자가 의도치않은 HTML/Javascript 수행을 방지
    - < (&lt;), > (&gt;), 홑따옴표(&#39;), 쌍따옴표(&quot;), & (&amp;) 를 변환
  - 장고 템플릿 언어에서는 디폴트로 escape 처리합니다.
  - autoescape 태그를 통한 처리도 가능
- **safe** : escape처리되지 않도록, SafeText로 래핑
  - 파이썬 객체의 값을

# date

- 지정 포맷으로 날짜 포맷팅
- 관련 settings
  1. DATE\_FORMAT : 디폴트 "N j, Y" (예: "Feb. 4, 2003")
  2. DATETIME\_FORMAT : 디폴트 "N j, Y, P" (예: "Feb. 4, 2003, 4 p.m.")
  3. SHORT\_DATE\_FORMAT : 디폴트 "m/d/Y" (예: "12/31/2003")
  4. SHORT\_DATETIME\_FORMAT : 디폴트 "m/d/Y P" (예: "12/31/2003 4 p.m.")
- 위 포맷 중 택일 지정하거나, 커스텀 지정. 미지정시에 DATE\_FORMAT으로 지정
- settings.USE\_L10N=True 시에는 settings.LANGUAGE\_CODE에 맞춰, 번역 사용



# time

- 지정 포맷으로 시간 포맷팅
- 관련 settings
  1. TIME\_FORMAT : 디폴트 "P" (예: "4 p.m")
- 커스텀 지정. 미지정시에 TIME\_FORMAT으로 지정
- settings.USE\_L10N=True 시에는 settings.LANGUAGE\_CODE에 맞춰, 번역 사용

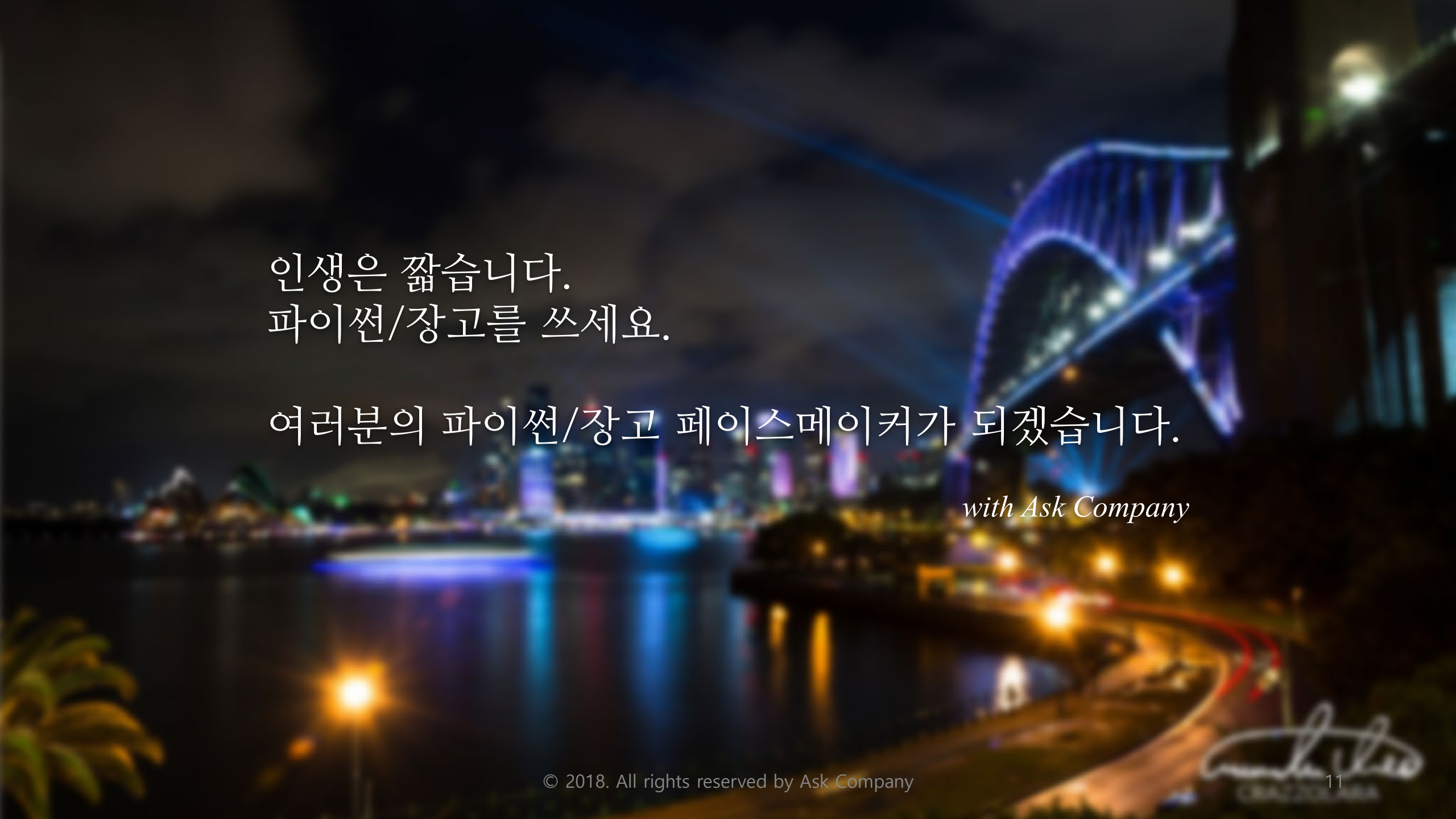
# json\_script 필터 (장고 2.1에 추가)

- 파이썬 객체를 JSON으로의 처리
  - 파이썬 문법과 JSON문자열 문법이 비슷한 측면이 있기에, 그대로 템플릿에 출력하여, 사용하기도 했었음.
  - django.core.serializers.json.DjangoJSONEncoder를 통한 json 직렬화
  - 변환된 JSON 문자열에 대해서, '>', '<', '&' 문자열 ESCAPE 처리 (XSS 공격 방지)

```
{{ value|json_script:"my-id" }}
```

```
<script id="my-id" type="application/json">{"hello": "world"}</script>
```

```
var value = JSON.parse(document.getElementById('my-id').textContent);
```



인생은 짧습니다.  
파이썬/장고를 쓰세요.

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

*with Ask Company*