Ask Django

여러분의 파이썬/장고 페이스메이커가 되겠습니다.
EP 2. 위젯 만들기 (기초)

위젯의 구조

EmailField의 예

Widget

django/forms/widgets.py

```
class EmailInput(Input):
    input_type = 'email'
    template_name = 'django/forms/widgets/email.html'
```

Widget Template

django/forms/templates/django/forms/widgets/email.html

```
{% include "django/forms/widgets/input.html" %}
```

※ *input* 태그의 *type*만 다르니깐, 별도 템플릿이 없습니다.

django/forms/templates/django/forms/widgets/input.html

```
<input type="{{ widget.type }}"
       name="{{ widget.name }}"
       {% if widget.value != None %}value="{{ widget.value|stringformat:'s' }}"{% endif %}
       {% include "django/forms/widgets/attrs.html" %} />
```

※ 원 소스코드에는 줄바꿈이 없으나, 가독성을 위해 줄바꿈을 했습니다.

django/forms/templates/django/forms/widgets/attrs.html

```
{% for name, value in widget.attrs.items %}
    {% if value is not False %}
        {{ name }}
        {% if value is not True %}="{{ value|stringformat:'s' }}"{% endif %}
    {% endif %}
{% endfor %}
```

※ 원 소스코드에는 줄바꿈이 없으나, 가독성을 위해 줄바꿈을 했습니다.

커스텀 위젯 만들기

포인트

장고 템플릿 내에서 백엔드에서 처리되는 코드와 프론트엔드에서 처리되는 코드를 분리해서 생각하실 수 있어야 합니다.

- 백엔드단에서 처리되는 코드 : Django Template 코드
  - `{{ }}`, `{% %}`
  - 장고 템플릿 코드는 뷰에서 응답을 생성할 때 수행됩니다. 이때 HTML/CSS/JavaScript 모두 단순한 문자열로 취급받습니다.
- 프론트단(브라우저)에서 처리되는 코드 : HTML/CSS/JavaScript
  - 서버로부터 받은 HTML 응답내의 HTML/CSS/JavaScript 코드를 읽어들여 순서대로 처리합니다.

Widget 클래스의 주요 클래스변수/멤버함수

django.forms.widgets.Widget

- `build_attrs(self, base_attrs, extra_attrs=None)`
  - 태그의 추가 속성/값 목록을 반환 (dict 타입)
  - `django/forms/templates/django/forms/widgets/attrs.html`에 의해 사용
- **`template_name`** : render시에 사용될 템플릿 경로
- **`get_context(self, name, value, attrs)`**
  - 템플릿 내에서 사용할 변수 목록을 반환 (dict 타입)
- `render(self, name, value, attrs=None, renderer=None)`
  - render된 html을 반환 (str 타입)

상속을 통한, 위젯 기본 코드

```
class TextInput(Input):
    input_type = 'text'
    template_name = 'django/forms/widgets/text.html'


class CustomTextInput(TextInput):
    template_name = '...'

    def build_attrs(base_attrs, extra_attrs=None):
        attrs = super().build_attrs(base_attrs, extra_attrs)
        attrs['...'] = '...'
        return attrs

    def get_context(self, name, value, attrs):
        context = super().get_context(name, value, attrs)
        context['...'] = '...'
        return context
```

글자수를 알려주는 위젯

```
class CounterTextInput(TextInput):
    template_name = 'widgets/counter_text.html'
```

템플릿경로/widgets/counter_text.html

```
{% include "django/forms/widgets/input.html" %}
<span id="counter_{{ widget.attrs.id }}"></span>

<script>
document.querySelector("#{{ widget.attrs.id }}").addEventListener("input", function () {
    document.querySelector('#counter_{{ widget.attrs.id }}').innerHTML = this.value.length + '글자';
})
</script>
```

한 Form 내에서 같은 위젯이 다수 사용될 수 있기 때문에, 추가로 사용하는 HTML 태그에는 **개별 id**를 부여해야만 합니다.