

Field Error

`{{ field.errors }}` - It outputs a `<ul class="errorlist">` containing any validation errors corresponding to this field. You can customize the presentation of the errors with a `{% for error in field.errors %}` loop. In this case, each object in the loop is a string containing the error message.

Example:- `{{ form.name.errors }}`

```
<ul class="errorlist">
  <li>Enter Your Name</li>
</ul>
```

`{{ form.non_field_errors }}` - This should be at the top of the form and the template lookup for errors on each field.

Example:- `{{ form.non_field_errors }}`

```
<ul class="errorlist nonfield">
  <li>Generic validation error</li>
</ul>
```

Styling Django Form Errors

If you render a bound Form object, the act of rendering will automatically run the form's validation if it hasn't already happened, and the HTML output will include the validation errors as a `<ul class="errorlist">` near the field.

We can use this class `errorlist` to style error.

Styling Django Form Errors

`error_css_class` and `required_css_class` – These Form class hooks can be used to add class attributes to required rows or rows with errors. Rows will be given "error" and/or "required" classes, as needed.

Example:-

```
class StudentRegistration(forms.Form):
```

```
    error_css_class = 'error'
```

```
    required_css_class = 'required'
```