# Django Form API – Create Django Form

# Code are at ch 27

# Video 33

Django provides a **Form API** (django.forms) that allows us to:

1. Generate HTML forms automatically.
2. Handle validation.
3. Render forms in different formats (as\_p, as\_table, as\_ul).
4. Customize how fields are displayed (auto\_id, label\_suffix, field\_order, initial, etc.).

## 🔹 Step 1: forms.py

**from** django **import** forms

class **Registration**(*forms***.***Form*)**:**

    first\_name **=** forms**.**CharField()

    last\_name **=** forms**.**CharField()

    email **=** forms**.**EmailField()

    city **=** forms**.**CharField()

*# Django automatically:*

*# - Converts snake\_case → "First name"*

*# - CharField → <input type="text">*

*# - EmailField → <input type="email">*

class **login**(*forms***.***Form*)**:**

    email **=** forms**.**EmailField()

    passward **=** forms**.**CharField()

👉 Explanation:

* CharField() → <input type="text">
* EmailField() → <input type="email">
* Field labels auto-generated:
  + first\_name → **First name** (Django converts underscore \_ to space and capitalizes first letter).

## 🔹 Step 2: Registration.html (Rendering Methods)

**<**h1**>**Registration page**</**h1**>**

**<**!-- Method 1 --**>**

**<**form action**=**""**>**

    {{form}}

**<**button**>**Submit**</**button**>**

**</**form**>**

**<**!-- Method 2 --**>**

**<**form action**=**""**>**

**<**table**>**

        {{form**.**as\_table}}

**</**table**>**

**<**button**>**Submit**</**button**>**

**</**form**>**

**<**!-- Method 3 --**>**

**<**form action**=**""**>**

**<**ul**>**

        {{form**.**as\_ul}}

**</**ul**>**

**<**button**>**Submit**</**button**>**

**</**form**>**

**<**!-- Method 4 --**>**

**<**form action**=**""**>**

    {{form**.**as\_p}}

**<**button**>**Submit**</**button**>**

**</**form**>**

**<**!-- Method 5 --**>**

**<**form action**=**""**>**

    {{form**.**first\_name}}

**<**button**>**Submit**</**button**>**

**</**form**>**

### 👉 Explanation of methods:

1. **{{ form }}**
   * Default output (similar to as\_table).
   * Each field rendered with <label> and <input>.
2. **{{ form.as\_table }}**
   * Each field inside <tr><td> elements.
   * Suitable if you want a table layout.
3. **{{ form.as\_ul }}**
   * Each field inside <li> elements.
   * Useful for unordered list layouts.
4. **{{ form.as\_p }}**
   * Each field inside <p> tags.
   * Most commonly used (cleanest).
5. **Direct access ({{ form.first\_name }})**
   * Renders **only one field** (<input type="text" name="first\_name">).
   * Useful when designing custom form layouts.

## 🔹 Step 3: login.html

**<**h1**>**Login Form**:</**h1**>**

**<**form action**=**""**>**

    {{loginform}}

**<**button**>**Submit**</**button**>**

**</**form**>**

## 🔹 Step 4: Views.py

**from** django**.**shortcuts **import** render

**from** student**.**forms **import** Registration**,** login

*# Registration view*

**def** registration(**req**)**:**

*# Default*

*# fm = Registration()*

*# Change field order*

    fm **=** Registration(**field\_order=**['email'**,** 'city'])

**return** render(req**,** 'student/registration.html'**,** {'form'**:** fm})

*# Login view*

**def** loginpage(**req**)**:**

*# Different ways of customizing forms:*

*# 1. auto\_id=True  → default id="id\_fieldname"*

*# lp = login(auto\_id=True)*

*# 2. auto\_id='id\_%s' → id="id\_email", id="id\_passward"*

*# lp = login(auto\_id='id\_%s')*

*# 3. auto\_id='talha\_%s' → id="talha\_email", id="talha\_passward"*

*# lp = login(auto\_id='talha\_%s')*

*# 4. auto\_id=False → No <label for=""> attribute connected to input*

*# lp = login(auto\_id=False)*

*# 5. label\_suffix='A' → Adds "A" after every label (e.g. "EmailA")*

*# lp = login(label\_suffix='A')*

*# 6. label\_suffix=' ' → Removes colon (default `:`), just space*

    lp **=** login(**label\_suffix=**' ')

*# 7. initial values → Pre-fill the form fields*

*# lp = login(initial={'email': 'sonam@example.com', 'passward': '1234'})*

**return** render(req**,** 'student/login.html'**,** {'loginform'**:** lp})

## 🔹 Step 5: View Source Example (with label\_suffix='A')

**<**div**>**

**<**label **for**="id\_email"**>**EmailA**</**label**>**

**<**input *type*="email" name="email" maxlength="320" required id="id\_email"**>**

**</**div**>**

**<**div**>**

**<**label **for**="id\_passward"**>**PasswardA**</**label**>**

**<**input *type*="text" name="passward" required id="id\_passward"**>**

**</**div**>**

## 🔹 Detailed Explanation of Options

### 1. auto\_id

* Default: auto\_id='id\_%s'
* %s replaced with field name.

| **Code** | **Output in HTML** |
| --- | --- |
| login(auto\_id=True) | <input id="id\_email"> |
| login(auto\_id='id\_%s') | <input id="id\_email"> |
| login(auto\_id='talha\_%s') | <input id="talha\_email"> |
| login(auto\_id=False) | No id attribute in input |

### 2. label\_suffix

* Default = :
* Changes what comes after the label.

| **Code** | **Label Example** |
| --- | --- |
| login(label\_suffix='A') | EmailA |
| login(label\_suffix=' ') | Email |
| login(label\_suffix='-->') | Email--> |

### 3. field\_order

* Changes the order of fields.

fm **=** Registration(**field\_order=**['email'**,**'city'])

👉 Renders email and city first, then first\_name, last\_name.

### 4. initial

* Pre-fills fields with values.

lp **=** login(**initial=**{'email'**:** 'test@example.com'**,** 'passward'**:** '1234'})

👉 Input boxes will already contain these values when the page loads.

# ✅ Summary Notes

* **Form Rendering Methods:**
  + {{form}} → Full form.
  + as\_table, as\_ul, as\_p → Different HTML structures.
  + {{form.field}} → Single field.
* **Customization Options:**
  + auto\_id='id\_%s' → Controls input id.
  + label\_suffix='X' → Controls label suffix.
  + field\_order=[...] → Change order of fields.
  + initial={...} → Pre-fill values.
* **Advantages:**
  + No manual HTML needed.
  + Auto validation.
  + Easy customization.