**Form validations using cleaned\_data**

The **cleaned\_data** is used to validate certain fields. Basically, any information entered into a form will be sent to the server as strings. It makes no difference which type of form field was utilized when the form was created. Everything would ultimately be converted to strings by the browser.

So, we can say that Django automatically changes data to the appropriate type when it cleans it. This cleansed and validated data is referred to as cleaned data in Django. And, the cleaned data dictionary provides access to cleaned data.

**Syntax**

cleaned\_data[‘field-name’]

**models.py**

from django.db import models

# Create your models here.

class Employee(models.Model):

    name=models.CharField(max\_length=20)

    email=models.EmailField()

    contact=models.CharField(max\_length=40)

forms.py

from django.forms import ModelForm

from app1 import models

class EmpForm(ModelForm):

    class Meta:

        model=models.Employee

        fields="\_\_all\_\_"

views.py

from django.shortcuts import render

from app1.forms import EmpForm

# Create your views here.

def empview(request):

    if request.method=="POST":

        eform=EmpForm(request.POST)

        if eform.is\_valid():

            name=eform.cleaned\_data['name']

            email=eform.cleaned\_data['email']

            contact=eform.cleaned\_data['contact']

            print(name,type(name))

            print(email,type(email))

            print(contact,type(contact))

    else:

        eform=EmpForm()

        return render(request,"home.html",context={"form":eform})

template

home.html

<form action="/eview" method="post">

    {% csrf\_token %}

    {{form.as\_p}}

<input type="submit" value="print">

</form>

**Settings.py**

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'app1',

]

import os.path

template\_path=os.path.join(BASE\_DIR,"templates")

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [template\_path],

        'APP\_DIRS': True,

        'OPTIONS': {

**Urls.py**

from django.contrib import admin

from django.urls import path

from app1 import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('eview',views.empview),

]

**Django form validation error message**

The **error\_message** attribute permits you to add custom error messages to the fields. This feature allows you to override the predefined error message and define your own. And, the error message is sent in the form of a dictionary data type.