

Ex:

D:\Django_20MAR_7PM>django-admin startproject advtemplateproject

D:\Django_20MAR_7PM>cd advtemplateproject

D:\Django_20MAR_7PM\advtemplateproject>py manage.py startapp testapp

-->Add app in settings.py

- **base.html**

```
<body>
  <nav class='navbar'>
    <a class='navbar' href="#">Mahesh News</a>
    <div class="container">
      <ul class='navbar-nav'>
        <li><a class='navbar' href="#">Movies</a></li>
        <li><a class='navbar' href="#">Sports</a></li>
        <li><a class='navbar' href="#">Politics</a></li>
      </ul>
    </div>
  </nav>
  {% block body_block %}
  {% endblock %}
</body>
```

- **views.py**

```
def base_view(request):
    return render(request, 'testapp/base.html')
```

- **urls.py**

```
path("", views.base_view)
```

- **movie.html**

```
<!DOCTYPE html>
{% extends 'testapp/base.html' %}
{% load static %}
{% block body_block %}
<h1>This is movies information</h1>
{% endblock %}
```

- **views.py**

```
def movie_view(request):  
    return render(request, 'testapp/movie.html')
```

- **urls.py**

```
path('movie/', views.movie_view)
```

- **sports.html**

```
<!DOCTYPE html>  
{% extends 'testapp/base.html' %}  
{% load static %}  
{% block body_block %}  
<h1>This is sports information</h1>  
{% endblock %}
```

- **views.py**

```
def sports_view(request):  
    return render(request, 'testapp/sports.html')
```

- **urls.py**

```
path('sports/', views.sports_view),
```

- **politics.html**

```
<!DOCTYPE html>  
{% extends 'testapp/base.html' %}  
{% load static %}  
{% block body_block %}  
<h1>This is politics information</h1>  
{% endblock %}
```

- **views.py**

```
def politics_view(request):  
    return render(request, 'testapp/politics.html')
```

- **urls.py**

```
path('politics/', views.politics_view),
```

Note:

- 1).In the parent template, we can define any number of blocks. But child template is responsible to implement these blocks.
- 2).It is not mandatory to implement every parent block in child. Based on our requirement, child template can decide which blocks are required to implement.
- 3).While implementing blocks in child template, it is not required to follow order.

Advantages of Template Inheritance:

- 1).What ever code available in base template is by default available to child template and we are not required to write again. Hence it promotes code re-usability.
- 2).It reduces length of the code and improves readability.
- 3).It reduce development time.
- 4).It provides unique and same look and feel for total web application.

Template Filters:

-->In the template file, the injected data can be displayed by using template tags.

{{name}}
{{emp.eno}}

-->Before displaying to the end user, if we want to perform some modifications to the injected text, like add something or cut something, case conversions etc then we should go for filters.

Syn for Template Filter:

{{value | filtername}}
{{value | filtername:'argument'}}

-->Arguments to the filter are optional.

Ex:

D:\Django_20MAR_7PM>django-admin startproject filterproject

D:\Django_20MAR_7PM>cd filterproject

D:\Django_20MAR_7PM\filterproject>python manage.py startapp testapp

-->Add app in settings.py

- **models.py**

```
class FilterModel(models.Model):
    name = models.CharField(max_length=20)
    subject = models.CharField(max_length=20)
    dept = models.CharField(max_length=20)
    date = models.DateField()
```

-->makemigrations and migrate

- **admin.py**

```
from testapp.models import FilterModel
class FilterModelAdmin(admin.ModelAdmin):
    list_display = ['name','subject','dept','date']
admin.site.register(FilterModel,FilterModelAdmin)
```

- **views.py**

```
from testapp.models import FilterModel
def upper_data_view(request):
    records = FilterModel.objects.all()
    return render(request,'testapp/upperdata.html',{'records':records})
```

- **upperdata.html**

```
<body>
    {% for record in records%}
    <h1>{{record.name}} Information</h1>
    <ul>
        <li>Name:{{record.name | upper}}</li>
        <li>Subject:{{record.subject | lower}}</li>
        <li>Dept:{{record.dept | title}}</li>
        <li>Date:{{record.date | date:"d-m-Y"}}</li>
    </ul><hr>
    {% endfor %}
</body>
```

- **urls.py**

```
path('upperview/', views.upper_data_view)
```

- **date filters**

```
<li>Date:{{record.date | date:"d-m-Y"}}</li>
<li>Date:{{record.date | date:"m-d-Y"}}</li>
```

Date:{{record.date | date:"m/d/Y"}}
Date:{{record.date | timesince}}
Date:{{record.date | date:'d-b-Y'}}
Date:{{record.date | date:'l,F j, Y'}}