Template

What is template?

- Django provides a convenient way to generate dynamic HTML pages by using its template system
- A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted
- Django template engine is used to separate the design from the python code and allows us to build dynamic pages.

What we were doing so far?

- We were writing HTML code in python view functions.
- The view functions are not appropriate place to write HTML code but to write business logic
- HTML should be coded elsewhere for the separation of presentation and business logic, which leads to the better code management and maintenance
- templates are HTML files

Example Project

- To understand how to create and use templates, let’s see the following example project
- Outline for our task
  - Create project
  - Create application
  - Create templates folder
  - Update settings.py
  - Define views and templates
  - Set urls
  - Run server
  - Send requests

Create Project

- Open command prompt or windows shell

- Go to workspace directory
- Type following command

  Django-admin startproject firstproject

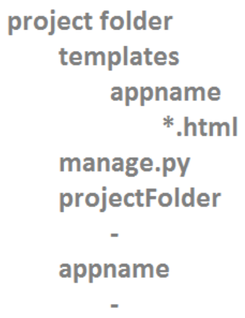
Create application

- Go to firstproject directory in workspace folder in the command prompt
- Type the following command

  Python manage.py startapp testapp

Creates templates folder

- Create templates folder inside firstproject (outer) folder
- Create folder with the same name as the name of app (testapp) inside templates folder
- Inside this testapp folder you have to create html files, known as templates



```
project folder
    templates
        appname
            *.html
    manage.py
    projectFolder
        -
    appname
        -
```

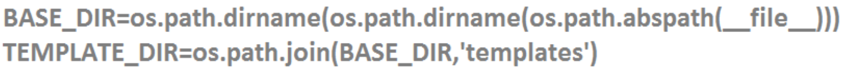
Update settings.py

- Add app in the Installed_apps list
- Add template folder path
  - In settings.py, create variable with path string

    TEMPLATE_DIR=**'f:/projects/firstproject/templates'**
  - Now add this variable to the list TEMPLATES. TEMPLATES list contains a dict element with one of the key DIRS. By default value of DIRS is empty list.
  - Write this TEMPLATE_DIR variable as list element for the key DIRS.

    'DIRS': [TEMPLATE_DIR],

- Hard coded paths are not recommended, because then it won’t be easy to move your project to some other directory. So the alternative is:
    - Create a variable with value as path of the templates folder



```
BASE_DIR=os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
TEMPLATE_DIR=os.path.join(BASE_DIR,'templates')
```

    - Now add this variable to the list TEMPLATES. TEMPLATES list contains a dict element with one of the key DIRS. By default value of DIRS is empty list.
    - Write this TEMPLATE_DIR variable as list element for the key DIRS.

```
'DIRS': [TEMPLATE_DIR],
```

Define Views and templates

- Inside view function, just mention the template file to render it.

```
res=render(request,’appname/filename.html’)
return res
```

- **showTest() function in exam/views.py**

```
def showTest(request):
   res=render(request,'exam/test.html')
   return res
```

- **Code of templates/exam/test.html**

```
<!DOCTYPE html>
<html>
 <head>
   <title>Test</title>
 </head>
 <body>
   <h1>Test Paper</h1>
   <div>
     <h2>Who developed C Language?</h2>
     <input type="radio" name="answer">Ken Thompson<br/>
     <input type="radio" name="answer">Dennis Ritchie<br/>
     <input type="radio" name="answer">Bjarne Stroustrup<br/>
     <input type="radio" name="answer">Saurabh Shukla<br/>
   </div>
   <input type="submit" >
 </body>
</html>
```

Set urls

- In exam/urls.py

```
urlpatterns =[
   url('test',views.showTest),
   url('result',views.showResult),
]
```

- In firstproject/urls.py

```
urlpatterns = [
   url('testapp/',include('testapp.urls')),
   url('exam/',include('exam.urls')),
]
```

Run Server

- Python manage.py runserver

Send requests

- http://localhost:8000/exam/test