1. install module summernote
2. ```
   pip install django-summernote
   ```

   add django-summernote to INSTALLED APPS in **setting.py**

   add django-summernote.urls to **urls.py**

   be sure that you installed static and media in server .

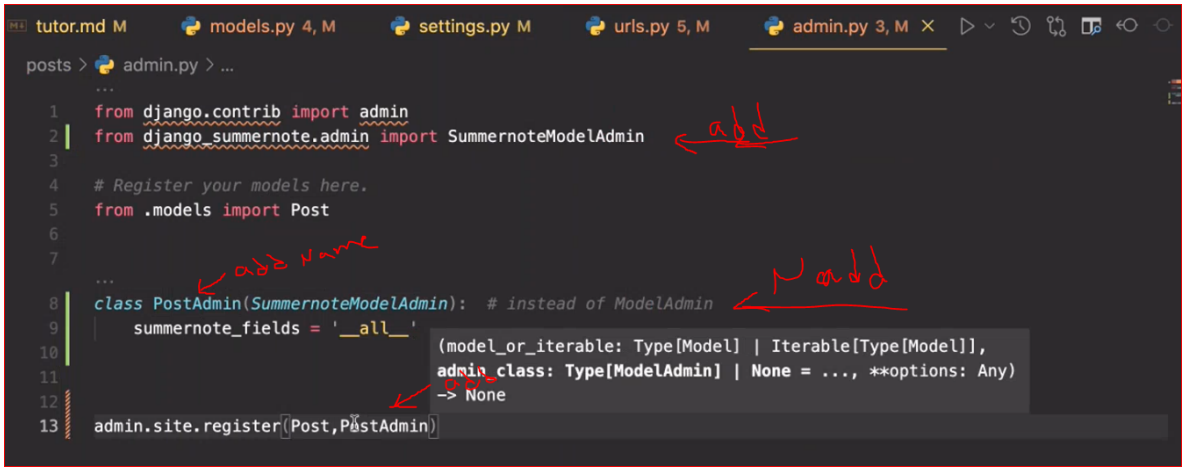
```
from django.urls import path,include -----> we add include to the command
urls.py

urlpatterns = [

    path('summernote/', include('django_summernote.urls')),

]
```

we add class to admin panel




```
class PostAdmin*(SummernoteModelAdmin):
    summernote_fields = '__all__'
```

after that

```
 python manage.py migrate
```
```

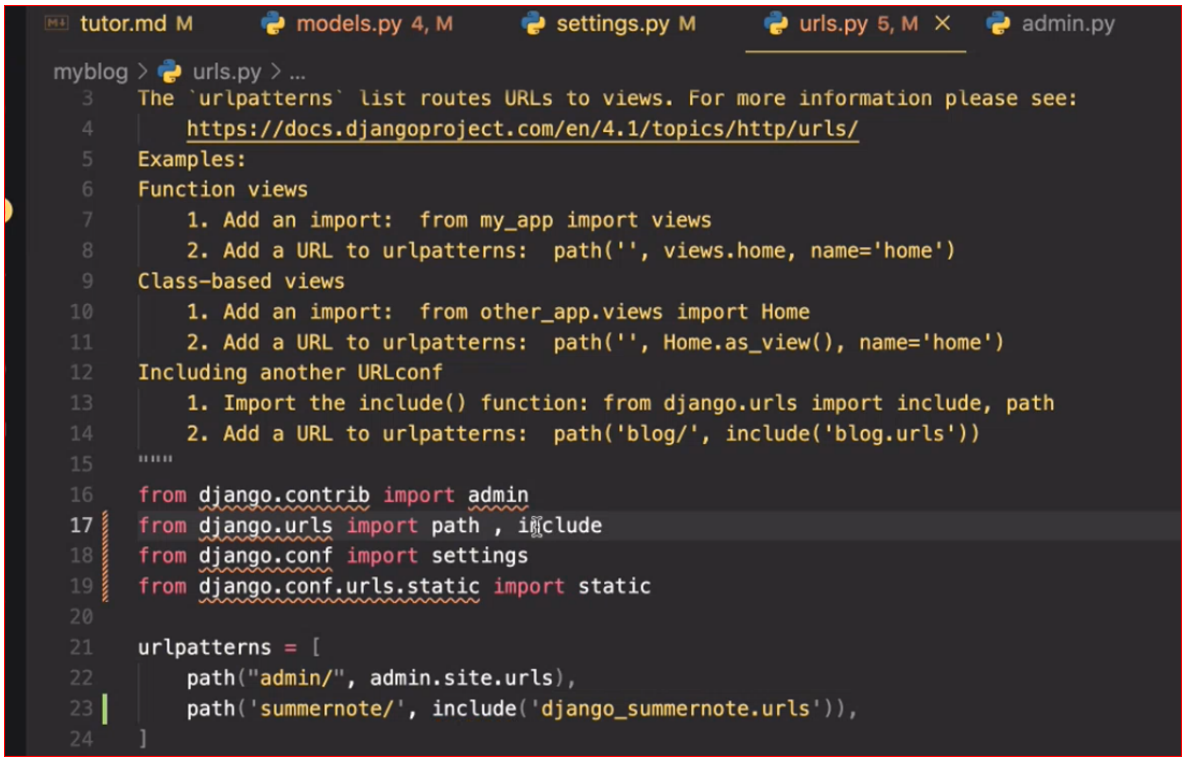
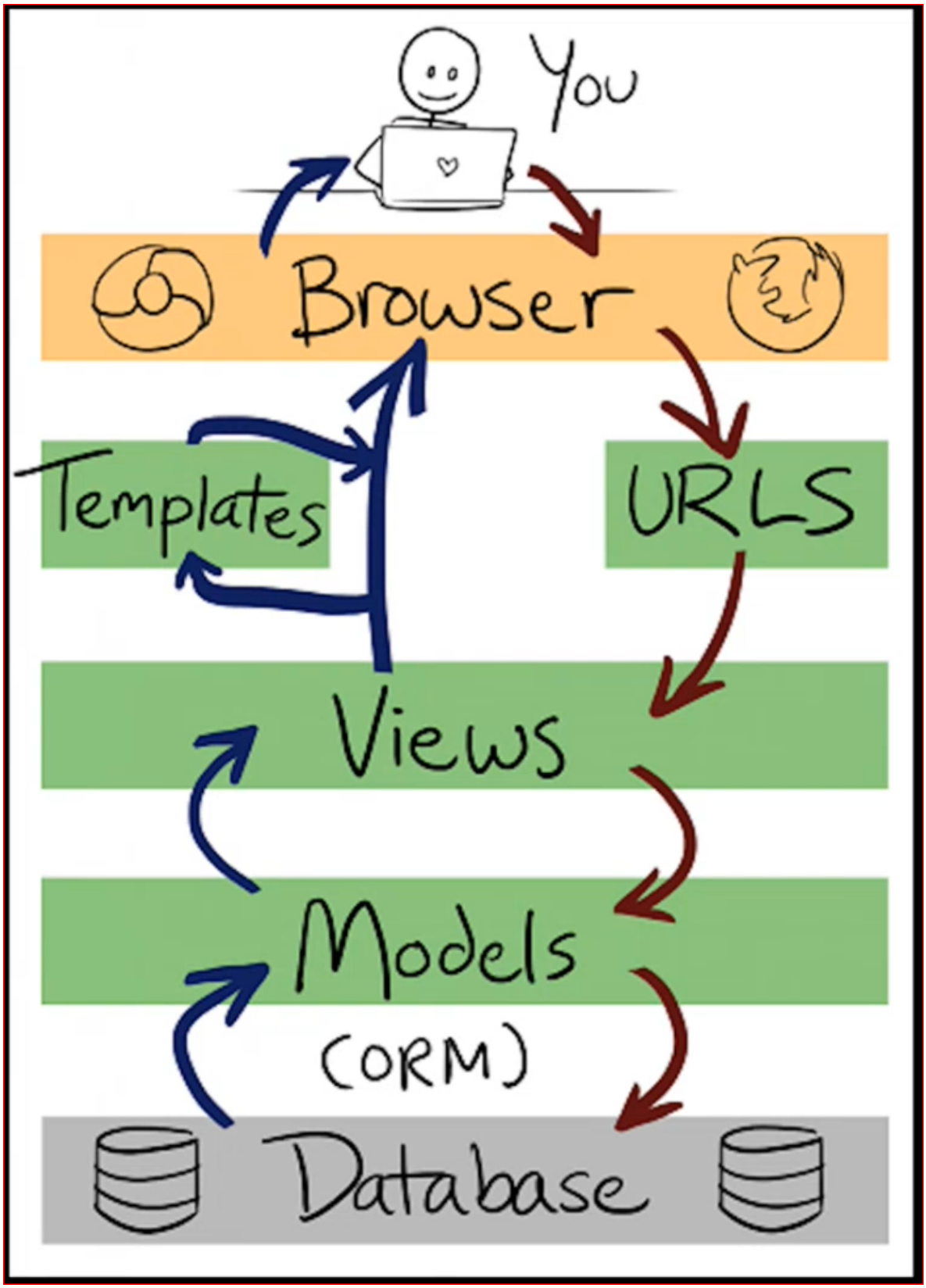
3. 

bild explain operation

4. first we go to view.py
5.



6. this bild explain all django background work .

7.

```
models : class
views : class , function
admin : class
forms : class
```

we create 2 functions in **view.py** :

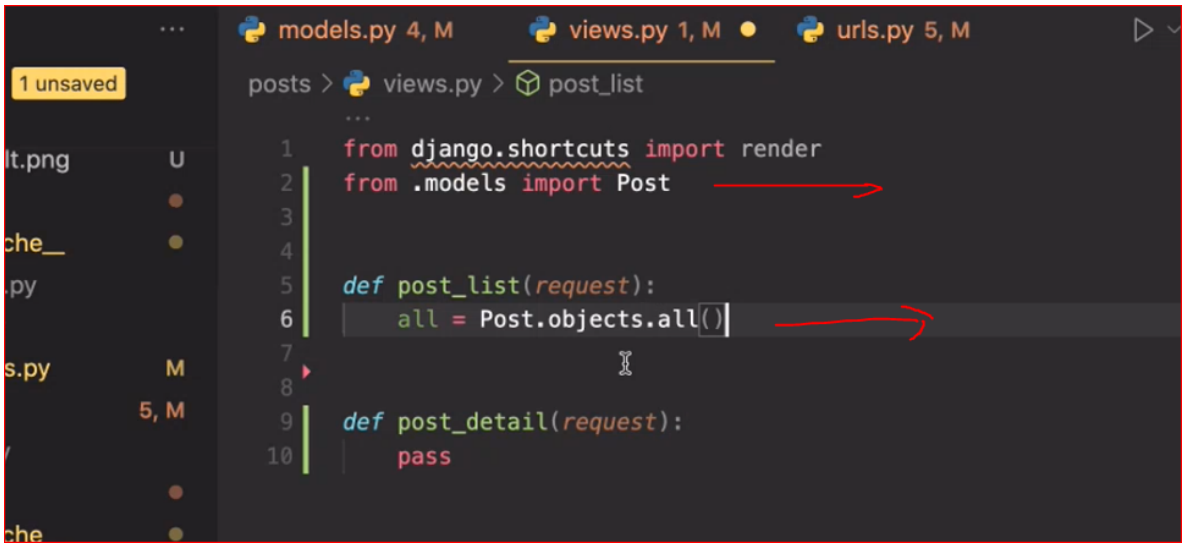
```
def post_list (request):   ----to view---  all post
  def post_detail(request):  -----to view---- 1 post
```

8. in url.py we write command :

```
from post.view import post_list , post_detail
```

9. in **url.py** we write command under :

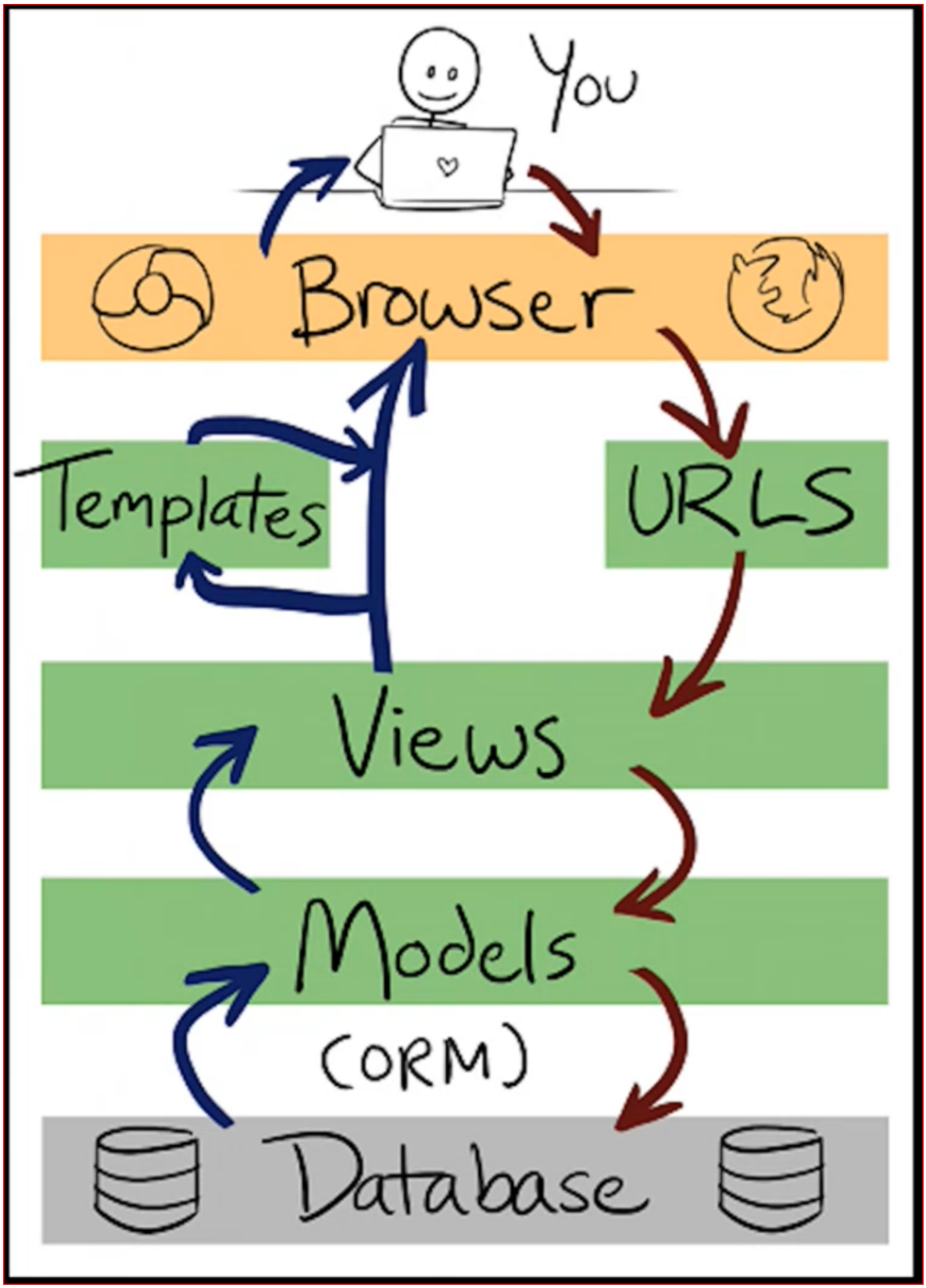
```
path('blog/', post_list) -----> we choose thise path to write in browser'blog/''
path('blog/', post_detail)
```

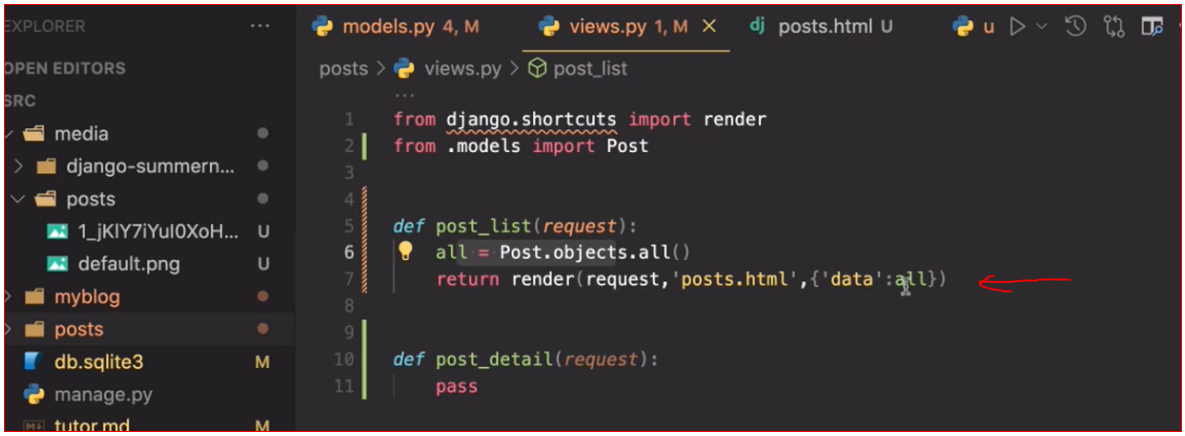
10. in view.py we call Post from models :
11. 


```
from .models import Post
```

12. in **view.py** in----

```
def post_list (request):
            all = Post.object.all() ---its mean view all post -->
            go to Post and make object and view all Post(what inside Post)--
after models goes to database bring data then go to views as (all= ) and make
render for data in Template then show that in Browser.
```



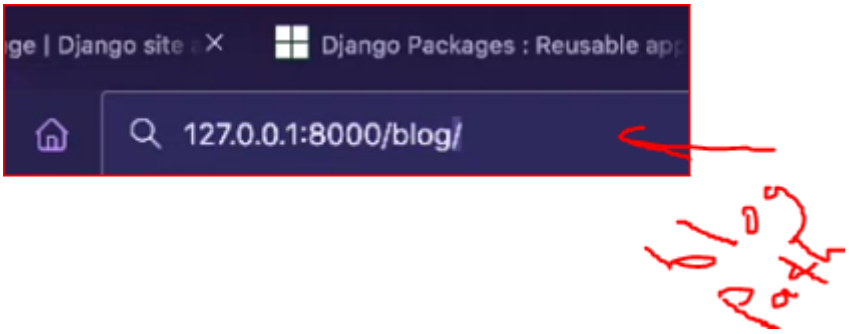
13. we create a file in Post named: templates/posts.html ---we create a posts.html inside templates
14. in **Template** Files that we created we need for every operation (**likes, add post**) we need to make **HTML page** .
15. after that we render : in view.py --
16. 


```
def post_list (request):
        all = Post.object.all() ---its mean view all post
        return render(request,'posts.html',{'data':all})

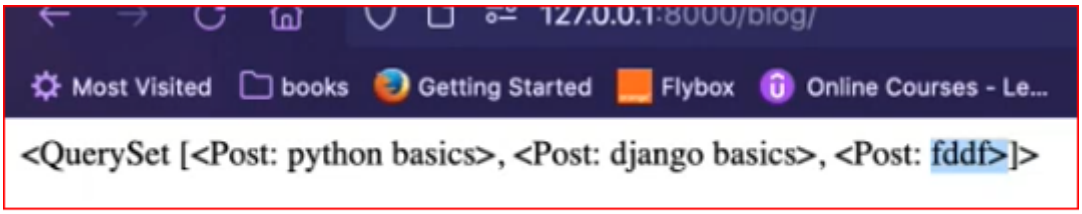
-here when we render data we 3 things (request,html page,data that comeback from
database  (all) but in dictionary{})
```

17. in posts.html -- that we created we write

```
{{data}}---it mean to view data in html
```

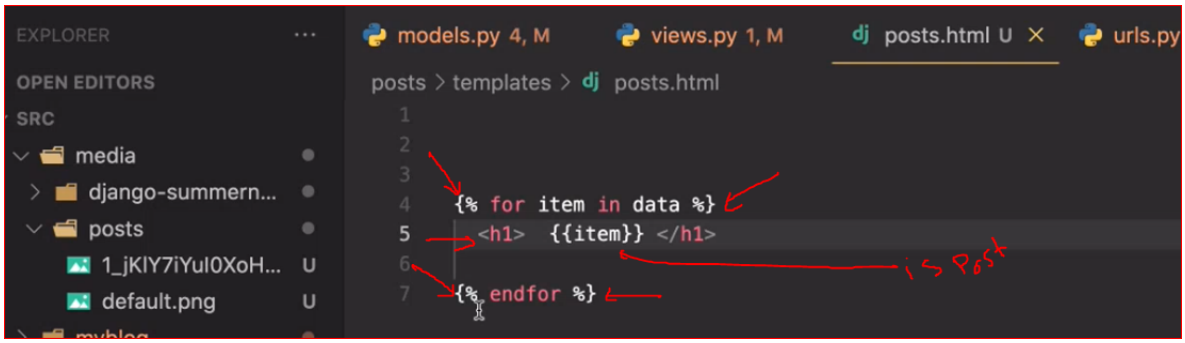


-- write in **Browser** seite http://127.0.1:8000/blog/ and it will view the data



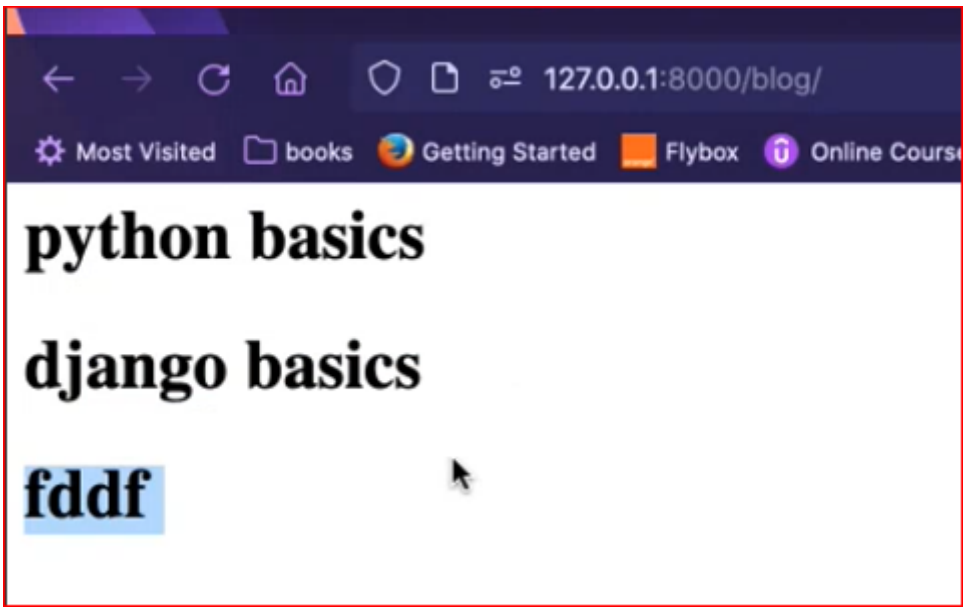
look to that data : that all post we created with tags with title names. what we do after that is : how that look in HTML tools (,

)..how data looks like in Browser



to make django understand that **For loop** not text is . we should add to it : **{%%}** and every loop has end **{%end%}**.

```
  django commmand :
  {% for item in data %}
  {{item}}  ----to defin variable in django and we need to end loop for in
django :
  {%endfor%} -- we view all thise operation in posts.html
```



---

18. in url.py we write other post ----- when call one post --- post_detail

```
path ('blog/<int:id>', post_detail)
```

19.
```
   in view.py     def post_detail(request,id):
            post = Post.objects.get(id=id) --(id=id---mean when id commes )
```

20. we create first templates/single.html ---- another webpage then we add return command

```
def post_detail(request,id):
            post = Post.objects.get(id=id)
            return render(request,'single.html',{'data':post})
```

21. in single.html we write command :{{data}}--- to view in page

22. in browser http://127.0.1.1:8000/blog/1 --------it will show you post with id :1
    in browser http://127.0.1.1:8000/blog/2 --------it will show you post with id :2
    in browser http://127.0.1.1:8000/blog/3 --------it will show you post with id :3

23.
```
    {% for item in data %}
```

```
            <h1><a  href="/blog/{{item.id}}">  {{item}}  </a></h1>--- make item in
link
 •

 ```
 {%endfor%}
 ```

 -- we view all thise operation in posts.html
```

19. in single.html we view data command :

```
{{data,author}}
```

20.
```
 we write html tag--- <img src="{{data.image.url}}" alt=""> -- to view post foto
```

21. django dont show you some data for security reason so if say django this file is secure:

```
we write this command : <p> {{data.content|safe}}</p>   ---- show you secure
data
```

22.