

- **Template Filters:**

**How to create our own filters:**

-->Based on our requirement, we can create own filters, If predefined filters are not fullfill our requirement.

**Steps:**

- 1).Create a folder 'templatetags' inside our application.
- 2).Create a special file named with '\_init\_.py' inside this folder(templatetags), so that django will consider this folder as python package.
- 3).Create a python file inside this folder to define our filters cust\_filters.py

- **cust\_filters.py**

```
from django import template

register = template.Library()

def first_five_upper(value):
    result = value[:3].upper()
    return result

def first_n_upper(value,n):
    result = value[:n].upper()
    return result

register.filter('ffu',first_five_upper)
register.filter('fnu',first_n_upper)
```

- **html file**

```
<body>
```

```
{% load cust_filters %}
```

```
<li>Name:{{record.name | ffu}}</li>
```

```
<li>Name:{{record.name | fnu:6}}</li>
```

### **Session Management:**

-->Client and server can communicate with some common language which is nothing but HTTP.

-->The basic limitation of HTTP is it is stateless protocol. i.e it is unable to remember client information for future purpose across multiple requests. Every request to the server is treated as new request only.

-->To remember client information at server side, some special mechanism must be required which is nothing but session management mechanism.

Different session management mechanisms are:

1).Cookies

2).Session API

### **Session management by using cookies:**

Cookies is a very small amount of information created by server and maintained by client. Cookies are key-value pairs.

To add cookies to the response.

```
response.set_cookie(cname,cvalue)
```

To get cookies send by client:

```
request.COOKIES[cname]
```

```
request.COOKIES.get(cname)
```

```
request.COOKIES.get(cname,defaultvalue)
```

**Ex:**

```
>>> d={101:'sunny'}
>>> d[101] #'sunny'
>>> d[102] #KeyError: 102
>>> print(d.get(101)) #sunny
>>> print(d.get(102)) #None
>>> print(d.get(101,'pinny')) #sunny
>>> print(d.get(102,'pinny')) #pinny
```

**Page Count Application**

```
D:\Django_20MAR_7PM>django-admin startproject pagecountproject
D:\Django_20MAR_7PM>cd pagecountproject
D:\Django_20MAR_7PM\pagecountproject>py manage.py startapp testapp
D:\Django_20MAR_7PM\pagecountproject>py manage.py makemigrations
D:\Django_20MAR_7PM\pagecountproject>py manage.py migrate
```

- **views.py**

```
def page_count_view(request):
    print('Cookies from the client:',request.COOKIES)
    count = int(request.COOKIES.get('count',0))
    count += 1
    response = render(request,'testapp/counter.html',{'count':count})
    response.set_cookie('count',count)
    return response
```

- **counter.html**

```
<body>  
    <h1>The Page Count:{{count}}</h1>  
</body>
```

- **urls.py**

```
path('page/',views.page_count_view),
```