Filters

When we need to modify variable before displaying we can use filters. Pipe ‘|’ is used to apply filter.

Syntax:- {{variable | filter}}

Example:- {{name|upper}}

- Built-in Filters – upper, truncateword:40,
- Custom Filters

Custom Filters

Custom filters are Python functions.

- Create Filter’s Function
- Register Filter
- Use Filter in Template

Geeky Steps

- Create a Folder inside Application named:- **templatetags**
- Create __init__.py File inside templatetags Folder to ensure the Folder is treated as a Python package.
- Create Modules as many as you need, inside templatetags package and write code for custom filter

Note - After adding the templatetags module, you will need to restart your server before you can use the tags or filters in templates.

Custom Filters

Creating Filter’s Function:-

```
def function_name(value, arg):
    return value.replace(arg, ‘’)
```

Register Filter:-

```
from django import template
register = template.Library()
register.filter(“filter_name”, filter_function)
```

Creating Filter’s Function And Registering:-

```
@register.filter(name=“filter_name”)
def function_name(value, arg):
    return value.replace(arg, ‘’)
```

```
@register.filter
def function_name(value, arg):
    return value.replace(arg, ‘’)
```

Note- If you leave off the name argument, Django will use the function’s name as the filter name.

Custom Filters

How to use Custom Filter in Templates:-

- Load the filter module in Template using load Tag

```
{% load filter_module_name %}
```

- Use Filter

```
{{ variable|filter_name }}
{{ variable|filter_name:argument }}
```