# Filter Template Tag

* Syntax: In most template engines, the filter tag is represented by a specific syntax. In Django, for example, it's typically written as {{ variable | filter\_name }}, where variable is the variable or expression you want to modify, and filter\_name is the name of the filter you want to apply.
* Functionality: When the template engine encounters a filter tag, it takes the value of the variable or expression on the left-hand side of the pipe symbol (|) and passes it through the specified filter function. The result of the filter function is then used in the output HTML.
* Chaining Filters: You can chain multiple filters together by separating them with additional pipe symbols. For example: {{ variable | filter1 | filter2 | ... }}.

{{ variable\_name|filter\_name }}

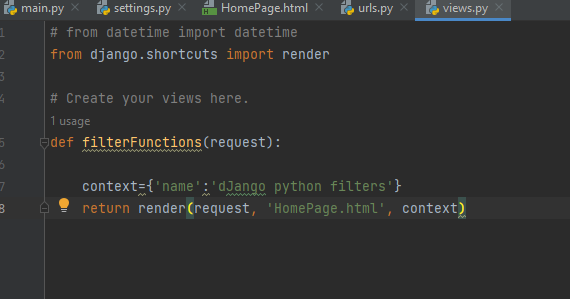
Where:

* variable\_name is the variable or expression you want to filter.
* filter\_name is the name of the filter you want to apply.

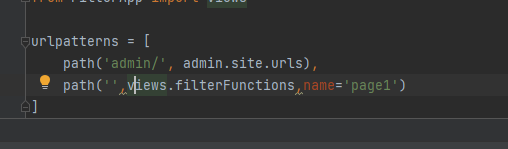
Here's an example of how you might use the filter template tag in a Django template:

**{{ name|title }}**

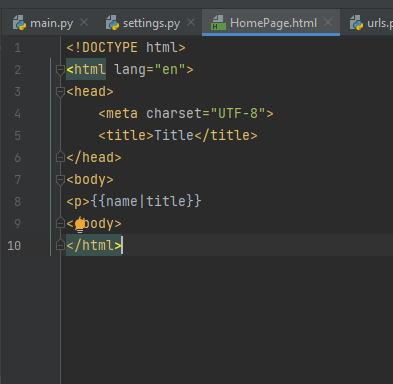
1){{ name|title }} takes whatever is stored in name, makes the first letter of each word uppercase, and then displays it.



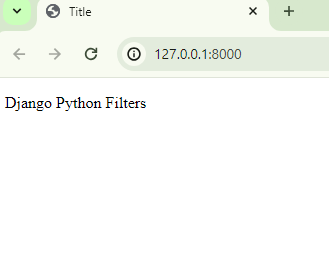
2)Set urls.py



3)



4)python manage.py runserver



**{{ name|upper }}**

<p>{{ name|upper }}</p> in a Django template essentially takes the value of name and displays it inside a paragraph element, with all letters converted to uppercase.

**{{ name|lower }}**

<p>{{ name|lower }}</p> in a Django template essentially takes the value of name and displays it inside a paragraph element, with all letters converted to lowercase.

**{{ name|capfirst }}**

{{ name|capfirst }} in a Django template takes the value of name and capitalizes the first letter, then displays it. This is particularly useful for ensuring consistent capitalization of names or titles.

