Django Template Language (DTL) Filtering Notes

1. Introduction to Filters in DTL

Filters in Django templates allow you to modify or format the data before rendering it to the frontend. You apply filters to variables using a pipe (`|`) symbol.

Basic Syntax:

```
{{ variable|filter_name:parameter }}
```

2. Commonly Used Filters

2.1. String Filters

- **lower**: Converts a string to lowercase.

  ```
  {{ name|lower }}
  ```

  Example: If `name = "John"`, the output will be `"john"`.

- **upper**: Converts a string to uppercase.

  ```
  {{ name|upper }}
  ```

  Example: If `name = "John"`, the output will be `"JOHN"`.

- **title**: Capitalizes the first letter of each word.

  ```
  {{ title|title }}
  ```

  Example: If `title = "hello world"`, the output will be `"Hello World"`.

- **truncatechars**: Truncates a string after a certain number of characters.

  ```
  {{ description|truncatechars:20 }}
  ```

  Example: If `description = "This is a long description"`, the output will be `"This is a long desc..."`.

- **default**: Provides a default value if the variable is empty or undefined.

```
{{ variable|default:"N/A" }}
```

Example: If `variable` is not set, the output will be `"N/A"`.

- **length**: Returns the length of a string or list.

```
{{ items|length }}
```

Example: If `items = ['apple', 'banana', 'cherry']`, the output will be `3`.

2.2. Numerical and Float Filters

- **floatformat**: Formats a float to a specified number of decimal places.

```
{{ price|floatformat:2 }}
```

Example: If `price = 23.456`, the output will be `"23.46"`.

- **add**: Adds a given number to the variable.

```
{{ count|add:5 }}
```

Example: If `count = 10`, the output will be `15`.

- **divisibleby**: Checks if the variable is divisible by a given number.

```
{% if value|divisibleby:3 %}
  <p>The value is divisible by 3.</p>
{% endif %}
```

2.3. Date and Time Filters

- **date**: Formats a datetime object.

```
{{ post.published_at|date:"F d, Y" }}
```

Example: If `post.published_at = datetime(2023, 10, 15)`, the output will be `"October 15, 2023"`.

- **time**: Formats a time object.

```
{{ event.start_time|time:"H:i" }}
```

Example: If `event.start_time = time(14, 30)`, the output will be `"14:30"`.

- **timesince**: Returns the time difference from now.

```
{{ post.created_at|timesince }}
```

Example: If `post.created_at` was 3 days ago, the output will be `"3 days"`.

2.4. Boolean and Conditional Filters

- **yesno**: Converts a boolean to a string.

```
{{ is_active|yesno:"Yes,No,Maybe" }}
```

Example: If `is_active = True`, the output will be `"Yes"`.

- **default_if_none**: Returns the default value if the variable is `None`.

```
{{ user.email|default_if_none:"No email provided" }}
```

Example: If `user.email` is `None`, the output will be `"No email provided"`.

2.5. List and Dictionary Filters

- **first**: Returns the first item of a list.

```
{{ items|first }}
```

Example: If `items = ['apple', 'banana', 'cherry']`, the output will be `"apple"`.

- **last**: Returns the last item of a list.

```
{{ items|last }}
```

Example: If `items = ['apple', 'banana', 'cherry']`, the output will be `"cherry"`.

- **slice**: Returns a slice of the list.

```
{{ items|slice:":2" }}
```

Example: If `items = ['apple', 'banana', 'cherry']`, the output will be `['apple', 'banana']`.

2.6. Miscellaneous Filters

- **join**: Joins a list with a string.

```
{{ items|join:", " }}
```

Example: If `items = ['apple', 'banana', 'cherry']`, the output will be `"apple, banana, cherry"`.

- **safe**: Marks a string as safe for HTML rendering (disables auto-escaping).

```
{{ "<strong>bold</strong>"|safe }}
```

Example: Outputs: `<strong>bold</strong>`.

- **urlize**: Converts URLs in plain text into clickable links.

```
{{ text|urlize }}
```

Example: Converts `"http://example.com"` to `<a href="http://example.com">http://example.com</a>`.

3. Advanced Filtering Examples

3.1. Filtering by Date

You can format a datetime object into a specific format.

```
<p>Published on: {{ post.published_at|date:"d F Y" }}</p>
```

Example: If `post.published_at = datetime(2023, 10, 15)`, the output will be `"15 October 2023"`.

3.2. Filtering by Float

```
<p>Total Price: ${{ total_price|floatformat:2 }}</p>
```

Example: If `total_price = 100.456`, the output will be `"100.46"`.

3.3. Filtering Strings

- **default**: Provides a default value when a variable is empty or undefined.

```
<p>{{ user.email|default:"No email provided" }}</p>
```

Example: If `user.email` is empty, the output will be `"No email provided"`.

- **truncatechars**: Truncate a string after a certain number of characters.

```
<p>{{ long_description|truncatechars:50 }}</p>
```

Example: If `long_description = "This is a long description of the product."`, the output will be `"This is a long description of the pr..."`.

3.4. Filtering Lists

```
<ul>
  {% for item in items|slice:":2" %}
    <li>{{ item }}</li>
  {% endfor %}
</ul>
```

Example: If `items = ['apple', 'banana', 'cherry']`, the output will be:

```
<ul>
  <li>apple</li>
  <li>banana</li>
</ul>
```

3.5. Filtering for Boolean

```
<p>Active: {{ is_active|yesno:"Yes,No" }}</p>
```

Example: If `is_active = True`, the output will be `"Yes"`.

3.6. Filtering by Length

```
<p>Number of items: {{ items|length }}</p>
```

Example: If `items = ['apple', 'banana', 'cherry']`, the output will be `"3"`.

4. Custom Filters

You can also create custom filters to extend Django's functionality. To do this, create a `templatetags` directory in your app and define a filter.

Example:

```
# templatetags/custom_filters.py
from django import template

register = template.Library()

@register.filter
def multiply(value, arg):
    return value * arg
```

Usage in template:

```
<p>Value multiplied: {{ number|multiply:5 }}</p>
```

This covers the filtering options available in Django Template Language, including basic and advanced use cases.