



## Exercise 20: Creating Custom Tags

🕒 20 to 30 minutes

In this exercise, you will add a custom tag to the `djangojokes.com` project. The custom tag will make it easy to put a random joke on any page on the site using this simple template tag:

```
{% random_joke %}
```

1. Open the **djangojokes.com** workspace in Visual Studio Code.
2. With `djangojokes.com` open in the terminal and the virtual environment activated, run:

```
(.venv) ~/projects/djangojokes.com> python manage.py startapp common
```

3. Add the new app to `INSTALLED_APPS`:

```
INSTALLED_APPS = [  
    ...  
  
    # Local apps  
    'common.apps.CommonConfig',  
    'jokes.apps.JokesConfig',  
    'pages.apps.PagesConfig',  
]
```

4. Within the `common` folder, create a `templatetags` folder, and create a `common_tags.py` file within that.
5. Add the following code to `common_tags.py`:<sup>35</sup>

---

35. **Don't want to type?** Copy from `starter-code/template-filters-and-tags/common_tags.py`.

## Exercise Code 20.1: common/templatetags/common\_tags.py

---

```
1. import random
2. from django import template
3.
4. from jokes.models import Joke
5.
6. register = template.Library()
7.
8. @register.inclusion_tag('common/joke.html')
9. def random_joke():
10.     count = Joke.objects.count()
11.     if count > 0: # In case we haven't added any jokes yet
12.         i = random.randint(0, count-1)
13.         joke = Joke.objects.all()[i]
14.         return {'joke': joke}
15.     else:
16.         return {
17.             'joke': {
18.                 'question': 'You know what is funny?',
19.                 'answer': 'There are no jokes in the database.'
20.             }
21.         }
```

---

### Things to notice:

- A. Just as with custom filters, you need to import `template` from `django` and create the `register` variable.
- B. You import `Joke` from `jokes.models`.
- C. The template function is decorated with:

```
@register.inclusion_tag('common/joke.html')
```

`inclusion_tag` functions must return a dictionary, which provides context for the associated template (in this case, `common/joke.html`), which Django will search for in the same places it searches for other templates.

- D. The function itself gets a random integer between 0 and one less than the number of jokes in the database. It uses that integer to get a random joke, which it returns in the dictionary. If there are no jokes, it returns a default joke.
6. Within the project's `templates` folder, create a new `common` folder, and within that, create a `joke.html` template with the following content:<sup>36</sup>

---

36. **Don't want to type?** Copy from `starter-code/template-filters-and-tags/joke.html`.

## Exercise Code 20.2: templates/common/joke.html

---

```
1. <div class="card border-primary m-auto mb-3 text-center"
2.     style="max-width: 30rem">
3.     <div class="card-header">Random Joke</div>
4.     <div class="card-body">
5.         <h5 class="card-title">{{ joke.question }}</h5>
6.         <p class="card-text">{{ joke.answer }}</p>
7.     </div>
8. </div>
```

---

This template works like any other template and will be included within the template that uses the custom `{% random_joke %}` tag.

7. Open `templates/pages/home.html`:

A. Immediately below the `{% extends "_base.html" %}` tag, add:

```
{% load common_tags %}
```

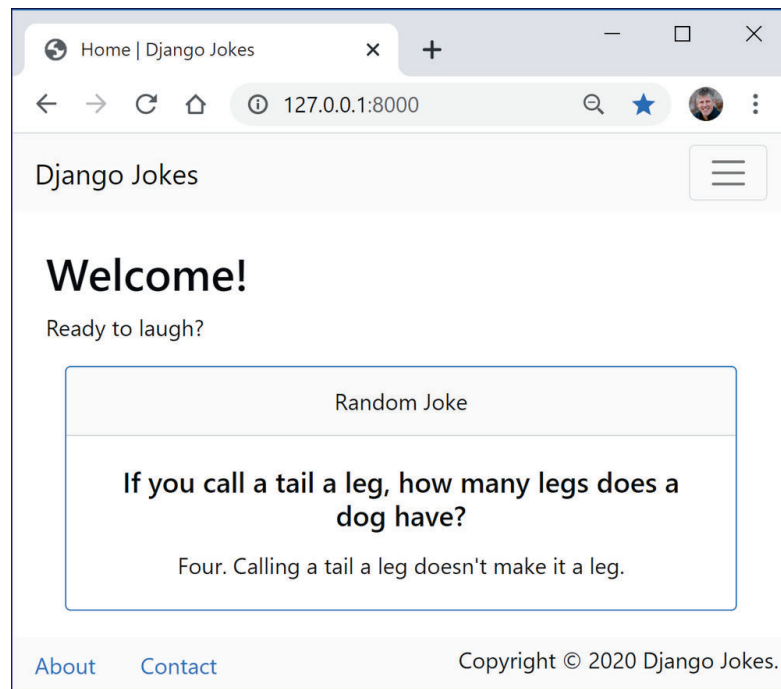
B. Immediately above the final `{% endblock %}`, add:

```
{% random_joke %}
```

8. Start up the server:

```
(.venv) ~/projects/djangojokes.com> python manage.py runserver
```

9. In your browser, navigate to `http://127.0.0.1:8000`. The site should look like this:



This custom template tag can be used on any page throughout the project.

## Git Commit

Commit your code to Git.

## Conclusion

In this lesson, you have learned to use Django's built-in template filters and tags to create your own custom filters and tags.