

## A recursive django template tag

admin · Friday, June 5th, 2009

Here is my attempt to create a "silver bullet" tag for printing tree structures with the Django templating language. It's far from a silver bullet, tho, but it can do basic stuff:

It's a modification of the standard "for" tag and i have kept the counter, counter0, first and last variables, only this time they are not attributes to forloop, but rather to recurseloop. Check the docstring for more info.

For example, if you need to print comments that have other comments as replies, you would want the comments to appear one below the other, but the replies to be indented a bit. Let's say 20 pixels per level. So the code would be:

```
{% load file_that_contains_recurse_tag %}
{% recurse comment in comments children="replies" indent=(0,20) %}
    {{ comment.text }}
{% endrecurse %}
```

This tag will expect a list of comments (top-level) that have a property named 'replies' which contain other comments.

indent is an argument that will start with the float value of 0 and get increased by 20 on each depth level of the recursion. You can pass as many variables like indent as you like. They must be in the form

```
name=(float,float)
```

or strings will also work but must be enclosed in quotes

```
name=("string","string")
```

Caveat: You must not leave blank spaces between the equal signs when assigning children and additional incremented arguments. I will fix this later.

A second scenario is when you need the parent element to **contain** the children, like in unordered/ordered lists. In that case you can use the {% yield %} tag inside the recurse block. This tag will output the HTML between the recurse and endrecurse tags if there are any children in the current iteration item, or it will output nothing if there are no children.

```
{% recurse comment in comments children="replies" indent=(0,20) %}
    {{ comment.text }}
    {% yield %}
{% endrecurse %}
```

The yield tag (as for now) can **only** be used directly inside the recurse block, much like the {% else %} tag can only be used directly inside the if-endif block. This means that you can't make code like

```
{% recurse comment in comments children="replies" indent=(0,20) %}
    ({{ comment.text }})
    {% if cond %}
        {% yield %}
    {% endif %}
{% endrecurse %}
```

This will fail with an invalid block tag exception. You can check the docstring of the do\_recurse function inside the code for more info. Also, you may want to check this code for errors since I just wrote it today, and haven't had much time to test it.

[Download and comment on any errors you find](#) :)

---

This entry was posted on Friday, June 5th, 2009 at 7:37 pm and is filed under [Django](#), [Programming](#), [Python](#). You can follow any responses to this entry through the [Comments \(RSS\)](#) feed. You can leave a response, or [trackback](#) from your own site.

---

4 response to "A recursive django template tag"

1.) JamesD says:

June 11th, 2009 at 3:03 pm

Thanks for the useful info. It's so interesting

2.) Benjamin Bach says:

June 19th, 2009 at 5:10 pm

Thanks for this, and the nice description. I had it working pretty quickly. The indention part isn't needed for me, though. You can easily obtain this with CSS. But it's a nice feature, anyways... put it on django-snippets. The one I found there doesn't appear to work in 1.1.

3.) [admin](#) says:

June 19th, 2009 at 5:52 pm

I want to add a {% loop %} functionality first, so you can print nested lists recursively :), you can see how that works here: <http://www.undefinedfire.com/lab/recursion-django-templates/>

4.) [Nanna Hanchett](#) says:

March 6th, 2010 at 8:36 pm

Thanks for the article. I am new at django and this will be a big help.

---

**The comment form is not available in PDF.  
But you can open the [post in a browser window](#) and comment there.**