

---

# **django-capture-tag Documentation**

*Release 1.0*

**Diederik van der Boor**

**Apr 12, 2018**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Syntax</b>	<b>5</b>
<b>3</b>	<b>Example usage</b>	<b>7</b>
3.1	Notice . . . . .	8



A micro-library to capture output in Django templates.

This can be useful for example to:

- Repeat page titles in web pages, e.g. for the `<title>` tag and breadcrumb.
- Repeat contents for Social Media tags.
- Reusing thumbnail output in multiple places.
- Fetch configuration data from extended templates.



# CHAPTER 1

---

## Installation

---

Install the module from PyPI:

```
pip install django-capture-tag
```

Add the package to INSTALLED\_APPS:

```
INSTALLED_APPS += (  
    'capture_tag',  
)
```

Load the tag in your template:

```
{% load capture_tags %}
```





The following options are available:

<code>{% capture %}...{% endcapture %}</code>	# output in <code>{{ capture }}</code>
<code>{% capture silent %}...{% endcapture %}</code>	# output in <code>{{ capture }}</code> only
<code>{% capture as varname %}...{% endcapture %}</code>	# output in <code>{{ varname }}</code>
<code>{% capture as varname silent %}...{% endcapture %}</code>	# output in <code>{{ varname }}</code> only



---

 Example usage
 

---

To capture Social Media tags:

```
{% load capture_tags %}

<head>
    ...

    {# Allow templates to override the page title/description #}
    <meta name="description" content="{% capture as meta_description %}{% block meta-
→description %}{% endblock %}{% endcapture %}" />
    <title>{% capture as meta_title %}{% block meta-title %}Untitled{% endblock %}{%_
→endcapture %}</title>

    {# display the same value as default, but allow templates to override it. #}
    <meta property="og:description" content="{% block og-description %}{{ meta_
→description }}{% endblock %}" />
    <meta name="twitter:title" content="{% block twitter-title %}{{ meta_title }}{%_
→endblock %}" />
</head>
```

Take configuration from extended templates:

```
# base.html

{% load capture_tags %}

# read once
{% capture as home_url silent %}{% block home_url %}{% url 'app:index' %}{% endblock
→%}{% endcapture %}

# reuse twice.
<a href="{{ home_url }}" class="btn page-top">Back to home</a>
<a href="{{ home_url }}" class="btn page-bottom">Back to home</a>

# child.html
```

```
{% extends "base.html" %}

{% block home_url %}{% url 'user:profile' %}{% endblock %}
```

## 3.1 Notice

When a value is used only once, this package is not needed. In such case, simply place the `{% block .. %}` at the proper location where contents is replaced. All common Django template tags support the `as variable` syntax, such as `{% url 'app:index' as home_url %}` or `{% trans "Foo" as foo_label %}`.