

🗨️ We want your feedback on Python Packaging.

Take our survey today! [🔗](#)



Search projects



[Help](#)

[Sponsors](#)

[Log in](#)

[Register](#)

# django-widget-tweaks

## 1.4.12



[Latest version](#)

Released: Jan 13, 2022

```
pip install django-widget-tweaks
```



**Tweak the form field rendering in templates, not in python-level form definitions.**

### Navigation

[Project description](#)

[Release history](#)

[Download files](#)

### Project description

Tweak the form field rendering in templates, not in python-level form definitions. Altering CSS classes and HTML attributes is supported.

That should be enough for designers to customize field presentation (using CSS and unobtrusive javascript) without touching python code.

License is MIT.

### Project links

### Installation

🗨️ We want your feedback on Python Packaging.

Take our survey today! [🔗](#)

## Statistics

GitHub statistics:

★ Stars: 1,768

🔗 Forks: 126

! Open  
issues/PRs: 38

View statistics for  
this project via  
[Libraries.io](#), or by  
using our public  
dataset on [Google  
BigQuery](#)

## Meta

License: MIT License  
(MIT license)

Author: [Mikhail  
Korobov](#) [✉](#)

Requires: Python  
≥3.7

## Maintainers



[jazzband](#)



[kmike](#)

## Classifiers

Development Status

- 5 -  
[Production/Stable](#)

```
$ pip install django-widget-tweaks
```

To enable *widget\_tweaks* in your project you need to add it to *INSTALLED\_APPS* in your projects *settings.py* file:

```
INSTALLED_APPS += [  
    'widget_tweaks',  
]
```

## Usage

This app provides two sets of tools that may be used together or standalone:

1. a `render_field` template tag for customizing form fields by using an HTML-like syntax.
2. several template filters for customizing form field HTML attributes and CSS classes

The `render_field` tag should be easier to use and should make form field customizations much easier for designers and front-end developers.

The template filters are more powerful than the `render_field` tag, but they use a more complex and less HTML-like syntax.

### render\_field

This is a template tag that can be used as an alternative to aforementioned filters. This template tag renders a field using a syntax similar to plain HTML attributes.

Example:

```
{% load widget_tweaks %}  
  
<!-- change input type (e.g. to HTML5) -->
```

📢 We want your feedback on Python Packaging.

Take our survey today! [🔗](#)

- Django :: 3.2
- Django :: 4.0

## Intended Audience

- Developers

## License

- OSI Approved :: MIT License

## Programming

### Language

- Python
- Python :: 3
- Python :: 3.10
- Python :: 3.7
- Python :: 3.8
- Python :: 3.9
- Python :: Implementation :: CPython
- Python :: Implementation :: PyPy

### Topic

- Software Development :: Libraries :: Python Modules

```
{% render_field form.text rows="20" cols="20" title="<!-- append to an attribute -->
{% render_field form.title class+="css_class_1 css_c

<!-- template variables can be used as attribute val
{% render_field form.text placeholder=form.text.label

<!-- double colon -->
{% render_field form.search_query v-bind::class="{act
```

For fields rendered with `{% render_field %}` tag it is possible to set error class and required fields class by using `WIDGET_ERROR_CLASS` and `WIDGET_REQUIRED_CLASS` template variables:

```
{% with WIDGET_ERROR_CLASS='my_error' WIDGET_REQUIRED
    {% render_field form.field1 %}
    {% render_field form.field2 %}
    {% render_field form.field3 %}
{% endwith %}
```

You can be creative with these variables: e.g. a context processor could set a default CSS error class on all fields rendered by `{% render_field %}`.

## attr

Adds or replaces any single html attribute for the form field.

Examples:

```
{% load widget_tweaks %}

<!-- change input type (e.g. to HTML5) -->
{{ form.search_query|attr:"type:search" }}

<!-- add/change several attributes -->
{{ form.text|attr:"rows:20"|attr:"cols:20"|attr:"titl
```

 Meta [🔗](#)

Meta is a  
Visionary sponsor  
of the Python  
Software  
Foundation.

*PSF Sponsor · Served  
ethically*

🗨️ We want your feedback on Python Packaging.

Take our survey today! [🔗](#)

```
<!-- attributes with double colon Vuejs output: v-bind::class::active::Value  
{{ form.search_query|attr:"v-bind::class::active::Value" }}
```

## add\_class

Adds CSS class to field element. Split classes by whitespace in order to add several classes at once.

Example:

```
{% load widget_tweaks %}  
  
<!-- add 2 extra css classes to field element -->  
{{ form.title|add_class:"css_class_1 css_class_2" }}
```

## set\_data

Sets HTML5 data attribute ( <http://ejohn.org/blog/html-5-data-attributes/> ). Useful for unobtrusive javascript. It is just a shortcut for 'attr' filter that prepends attribute names with 'data-' string.

Example:

```
{% load widget_tweaks %}  
  
<!-- data-filters:"OverText" will be added to input field -->  
{{ form.title|set_data:"filters:OverText" }}
```

## append\_attr

Appends attribute value with extra data.

Example:

```
{% load widget_tweaks %}
```

🗨 We want your feedback on Python Packaging.

Take our survey today! [🔗](#)

‘**add\_class**’ filter is just a shortcut for ‘**append\_attr**’ filter that adds values to the ‘**class**’ attribute.

## remove\_attr

Removes any single html attribute for the form field.

Example:

```
{% load widget_tweaks %}

<!-- removes autofocus attribute from field element -->
{{ form.title|remove_attr:"autofocus" }}
```

## add\_label\_class

The same as *add\_class* but adds css class to form labels.

Example:

```
{% load widget_tweaks %}

<!-- add 2 extra css classes to field label element -->
{{ form.title|add_label_class:"label_class_1 label_class_2" }}
```

## add\_error\_class

The same as ‘**add\_class**’ but adds css class only if validation failed for the field (field.errors is not empty).

Example:

```
{% load widget_tweaks %}

<!-- add 'error-border' css class on field error -->
{{ form.title|add_error_class:"error-border" }}
```

🗨️ We want your feedback on Python Packaging.

Take our survey today! [🔗](#)

## add\_error\_attr

The same as 'attr' but sets an attribute only if validation failed for the field (field.errors is not empty). This can be useful when dealing with accessibility:

```
{% load widget_tweaks %}

<!-- add aria-invalid="true" attribute on field error -->
{{ form.title|add_error_attr:"aria-invalid:true" }}
```

## add\_required\_class

The same as 'add\_error\_class' adds css class only for required field.

Example:

```
{% load widget_tweaks %}

<!-- add 'is-required' css class on field required -->
{{ form.title|add_required_class:"is-required" }}
```

## field\_type and widget\_type

'field\_type' and 'widget\_type' are template filters that return field class name and field widget class name (in lower case).

Example:

```
{% load widget_tweaks %}

<div class=<s>"field </s>{{ field|field_type }}<s> </s>
    {{ field }}
</div>
```

🗨️ We want your feedback on Python Packaging.

Take our survey today! [🔗](#)

```
<div class=<s>"field charfield textinput name"</s>>
  <input id=<s>"id_name"</s> type=<s>"text"</s> name="id_name"</s>
</div>
```

## Mixing render\_field and filters

The `render_field` tag and filters mentioned above can be mixed.

Example:

```
{% render_field form.category|append_attr:"readonly:" %}
```

returns:

```
<input name=<s>"category"</s> placeholder=<s>"Professor" type="text"</s>
```

## Filter chaining

The order `django-widget-tweaks` filters apply may seem counter-intuitive (leftmost filter wins):

```
{{ form.simple|attr:"foo:bar"|attr:"foo:baz" }}
```

returns:

```
<input foo=<s>"bar"</s> type=<s>"text"</s> name=<s>"simple"</s>
```

It is **not** a bug, it is a feature that enables creating reusable templates with overridable defaults.

Reusable field template example:

```
{# inc/field.html #}
```

🗨️ We want your feedback on Python Packaging.

Take our survey today! [🔗](#)

Example usage:

```
{# my_template.html #}
{% load widget_tweaks %}
<form method=<s>'POST'</s> action=<s>' '</s>> {% csrf_token %}>
    {% include "inc/field.html" with field=form.title %}
    {% include "inc/field.html" with field=form.description %}
</form>
```

With ‘rightmost filter wins’ rule it wouldn’t be possible to override `|attr:"foo:default_foo"` in main template.

## Contributing

If you’ve found a bug, implemented a feature or have a suggestion, do not hesitate to contact me, fire an issue or send a pull request.

- Source code: <https://github.com/jazzband/django-widget-tweaks/>
- Bug tracker: <https://github.com/jazzband/django-widget-tweaks/issues>

## Testing

Make sure you have `tox` installed, then type

```
tox
```

from the source checkout.

## NOT SUPPORTED

MultiWidgets: `SplitDateTimeWidget`, `SplitHiddenDateTimeWidget`

## Changes



■ **We want your feedback on Python Packaging.**

Take our survey today! [🔗](#)

- Add **better documentation syntax highlighting**.
- Adjust **build settings and stop building deprecated universal Python 2 wheels**.

#### 1.4.11 (2022-01-08)

- Add support for **Django 4.0**
- Drop support for **Django 3.0 and 3.1**
- Add support for **Python 3.10**
- Drop support for **Python 3.6**

#### 1.4.9 (2021-09-02)

- Add support for **Django 3.2**
- Move to **GitHub Actions**.
- Drop support for **Django 1.11**.
- Add support for **Python 3.9**.

#### 1.4.8 (2020-03-12)

- Fix **Release version**

#### 1.4.7 (2020-03-10)

- Fix **Travis deployment to Jazzband site**

#### 1.4.6 (2020-03-09)

- Feature **remove attribute from field**
- Added **documentation for remove\_attr feature**
- Reformat code with **black for PEP8 compatibility**
- More **consistent tox configuration**
- adding a **new templatetag, unittest and documentation**
- Deprecate **Python 2.7 support**
- Use **automatic formatting for all files**

🗨 **We want your feedback on Python Packaging.**

Take our survey today! [🔗](#)

#### 1.4.4 (2019-06-05)

- Add support for type attr.
- Add Python 3.7 and drop Python 3.3 support.
- Add support for double colon syntax.

#### 1.4.3 (2018-09-6)

- Added `add_label_class` filter for CSS on form labels
- Removed compatibility code for unsupported Django versions
- Fixed support for non-value attributes in Django < 1.8
- Support non-value attributes in HTML5 by setting their value to True

#### 1.4.2 (2018-03-19)

- update readme to make installation more clear
- shallow copy field before updating attributes
- drop Python 2.6 and Python 3.2 support
- always cast the result of render to a string
- fix import for django >= 2.0
- moved to jazzband

#### 1.4.1 (2015-06-29)

- fixed a regression in django-widget-tweaks v1.4 (the field is no longer deep copied).

#### 1.4 (2015-06-27)

- Django 1.7, 1.8 and 1.9 support;
- setup.py is switched to setuptools;
- testing improvements;
- Python 3.4 support is added;
- Python 2.5 is not longer supported;

🗨️ We want your feedback on Python Packaging.

Take our survey today! [🔗](#)

## 1.3 (2013-04-05)

- added support for `WIDGET_ERROR_CLASS` and `WIDGET_REQUIRED_CLASS` template variables that affect `{% render_field %}`.

## 1.2 (2013-03-23)

- new `add_error_attr` template filter;
- testing improvements.

### 1.1.2 (2012-06-06)

- support for template variables is added to `render_field` tag;
- new `field_type` and `widget_type` filters.

### 1.1.1 (2012-03-22)

- some issues with `render_field` tag are fixed.

## 1.1 (2012-03-22)

- `render_field` template tag.

## 1.0 (2012-02-06)

- filters return empty strings instead of raising exceptions if field is missing;
- test running improvements;
- python 3 support;
- undocumented 'behave' filter is removed.

## 0.3 (2011-03-04)

- `add_error_class` filter.

🗨 **We want your feedback on Python Packaging.**

Take our survey today! [↗](#)

- no more extra whitespaces;
- tests;

## 0.1 (2011-01-12)

Initial release.



### Help

Installing packages [↗](#)  
Uploading packages [↗](#)  
User guide [↗](#)  
FAQs

### About PyPI

PyPI on Twitter [↗](#)  
Infrastructure dashboard [↗](#)  
Package index name retention [↗](#)  
Our sponsors

### Contributing to PyPI

Bugs and feedback  
Contribute on GitHub [↗](#)  
Translate PyPI [↗](#)  
Development credits [↗](#)

### Using PyPI

Code of conduct [↗](#)  
Report security issue  
Privacy policy [↗](#)  
Terms of use

Status: All Systems Operational [↗](#)

Developed and maintained by the Python community, for the Python community.  
Donate today!

© 2022 Python Software Foundation [↗](#)

Site map

🗨 We want your feedback on Python Packaging.

Take our survey today! [🔗](#)

› English   español   français   日本語   português (Brasil)   українська   Ελληνικά   Deutsch   中文 (简体)  
中文 (繁體)   русский   עברית   esperanto



**AWS**  
Cloud computing



**Datadog**  
Monitoring



**Facebook /**  
**Instagram**  
PSF Sponsor



**Fastly**  
CDN



**Google**  
Object Storage and  
Download Analytics



**Huawei**  
PSF Sponsor



**Microsoft**  
PSF Sponsor



**NVIDIA**  
PSF Sponsor



**Pingdom**  
Monitoring



**Salesforce**  
PSF Sponsor



**Sentry**  
Error logging



**StatusPage**  
Status page