

MarkupSafe


 Star 302

MarkupSafe implements a text object that escapes characters so it is safe to use in HTML and XML. Characters that have special meanings are replaced so that they display as the actual characters. This mitigates injection attacks, meaning untrusted user input can safely be displayed on a page. Escaping is implemented in C so it is as efficient as possible.

```
>>> from markupsafe import Markup, escape
>>> # escape replaces special characters and wraps in Markup
>>> escape('<script>alert(document.cookie);</script>')
Markup(u'&lt;script&gt;alert(document.cookie);&lt;/script&gt;')
>>> # wrap in Markup to mark text "safe" and prevent escaping
>>> Markup('<strong>Hello</strong>')
Markup('<strong>hello</strong>')
>>> escape(Markup('<strong>Hello</strong>'))
Markup('<strong>hello</strong>')
>>> # Markup is a text subclass (str on Python 3, unicode on Python 2)
>>> # methods and operators escape their arguments
>>> template = Markup("Hello <em>%s</em>")
>>> template % "'World'"
Markup('Hello <em>&#34;World&#34;</em>')
```

 [pallets/markupsafe \(https://github.com/pallets/markupsafe\)](https://github.com/pallets/markupsafe)

 [Releases on PyPI \(https://pypi.python.org/pypi/MarkupSafe\)](https://pypi.python.org/pypi/MarkupSafe)

 [Test status \(https://travis-ci.org/pallets/markupsafe\)](https://travis-ci.org/pallets/markupsafe)

 [Documentation \(https://markupsafe.palletsprojects.com/\)](https://markupsafe.palletsprojects.com/)

Latest Release

Version: 1.1.1

 (<https://github.com/pallets/>)  (<https://twitter.com/PalletsTeam/>) 
(<https://github.com/pallets/website/tree/master/content/projects/markupsafe/co>

[License & Copyright \(/license/\)](#) · [Contact \(/contact/\)](#) · Made with  and [Lektor \(https://www.getlektor.com/\)](https://www.getlektor.com/) all around the World