Django is a framework, which tries to do "the right" thing by default. This means when you do the most simple thing, you're propably doing the right thing.

Now let's look at some template in php and python:

PHP:

<? echo $foo ?>

May give:

<script src="evil">

Django:

{{ foo }}

Gives with the same input:

&gt;script src="evil"&lt;

Now assume, you want to place a link <a href="link">text</a>. Then django will render it as text using &lt;&gt; again. If you know what you're doing, you can now use mark\_safe to indicate that the text is trusted (i.e. not coming from userinput).

You use {{ foo|safe }} or {% autoescape off %}{{foo}}{% endautoescape %} in your templates as django programmer, which is more clear when the string is declared as being safe.

So, where is mark\_safe used? When you write own templatetags or filters, then you need to mark the string as safe from python, because the developer will assume, that {{ foo|mylinkifyfunction }} does the right thing (i.e. it escapes the url foo, but does not escape the <a href=""></a> around the url).