

# Customizing Emails

Table of Contents

- 1 Conditional Statements
- 2 Loops
- 3 Filters
  - 3.1 Supported Filters
- 4 Debugging



Blesta 3.0 uses the [H2o Template system](#) for parsing Email Templates. The following is an overview of its basic usage. We've overridden the default {{ var\_name }} with { var\_name } for simplicity, but that's the only way in which the following documentation differ from any [official H2o documentation](#).

## Conditional Statements

```
{% if id > 3 %}  
This will print out if $id is > 3  
{% endif %}
```

## Loops

```
{% for user in users %}  
Name: {user.first_name} {user.last_name}  
{% endfor %}
```

Assuming:

```
$users = array(  
    array(  
        'first_name'=>"First name",  
        'last_name'=>"Last name"  
    ),  
    array(  
        'first_name'=>"First name 2",  
        'last_name'=>"Last name 2"  
    )  
);
```

Loops can also take advantage of limits:

```
{% for user in users limit:1 %}
```

The above would only cycle through the loop one time.

## Filters

Filters are applied to individual variables and are preceded by a pipe symbol (|). Parameters are passed to filters using a comma (,) separated list.

```
{past_due | set_default "Not past due"} // Produces "Not past due" if past_due is false
```

Multiple filters can be applied in succession:

```
{past_due | set_default "not past due" | capitalize} // Produces NOT PAST DUE if past_due is false
```

## Supported Filters

Filter	Description	Example Usage
--------	-------------	---------------

currency_format	<p>Formats the value into a currency string.</p> <p><b>Parameters</b></p> <ol style="list-style-type: none"> <li>1. currency code</li> </ol>	<pre>{ "1.2345"   currency_format "USD" }</pre> <p>For Invoice Delivery Email templates, you can use this example which includes the currency code and invoice total tags.</p> <pre>{ invoice.total   currency_format invoice.currency }</pre>
md5	<p>Hashes the value using MD5.</p>	<pre>{ "some string"   md5 }</pre>
sha1	<p>Hashes the value using SHA1.</p>	<pre>{ "some string"   sha1 }</pre>
numberformat	<p>Formats the value into a number.</p> <p><b>Parameters</b></p> <ol style="list-style-type: none"> <li>1. number of decimal places</li> <li>2. decimal point character</li> <li>3. thousands separator character</li> </ol>	<pre>{ "12345.6789"   numberformat 2, ".", "," }</pre>
wordwrap	<p>Wraps a string into multiple lines.</p> <p><b>Parameters</b></p> <ol style="list-style-type: none"> <li>1. number of characters per line</li> <li>2. break string</li> </ol>	<pre>{ "some string"   wordwrap 5, "\n&lt;br /&gt;" }</pre>
trim	<p>Removes white space around a string.</p>	<pre>{ " some string "   trim }</pre>
upper	<p>Converts a string to uppercase.</p>	<pre>{ "some string"   upper }</pre>
lower	<p>Converts a string to lowercase.</p>	<pre>{ "SOME STRING"   lower }</pre>
first	<p>Returns the first element of an array.</p>	<pre>{ array_of_items   first }</pre>
last	<p>Returns the last element of an array.</p>	<pre>{ array_of_items   last }</pre>
join	<p>Combines an array of elements into a delimited string.</p> <p><b>Parameters</b></p> <ol style="list-style-type: none"> <li>1. delimiter (default is ',')</li> </ol>	<pre>{ array_of_items   join ', ' }</pre>
length	<p>Returns the length of an array.</p>	<pre>{ array_of_items   length }</pre>

urlencode	URL encodes a string or array of parameters.	<pre>{array_of_items   urlencode}</pre>
hyphenize	Replaces white space with hyphens.	<pre>{"some string"   hyphenize}</pre>
urlize	Converts a URL into an HTML link.	<pre>{"http://domain.com/"   urlize}</pre>
set_default	Sets a default value when no other value exists.  <b>Parameters</b>  1. default value	<pre>{not_set   set_default "default value"}</pre>
humanize		<input type="text"/>
capitalize		<input type="text"/>
capfirst		<input type="text"/>
tighten_space		<input type="text"/>
escape		<input type="text"/>
escapejson		<input type="text"/>
force_escape		<input type="text"/>
truncate		<input type="text"/>
limitwords		<input type="text"/>
filesize		<input type="text"/>
image_tag		<input type="text"/>
css_tag		<input type="text"/>
script_tag		<input type="text"/>
links_to		<input type="text"/>
links_with		<input type="text"/>
strip_tags		<input type="text"/>
linebreaks		<input type="text"/>
nl2br		<input type="text"/>

nl2pbr		<input type="text"/>
htmlAttribute		<input type="text"/>
extract		<input type="text"/>
date		<pre>{"some date string"   date "Y M d"} eg. {invoice.date_due   date "Y M d"}</pre>
relative_time		<input type="text"/>
relative_date		<input type="text"/>
relative_datetime		<input type="text"/>

## Debugging

Adding the following tag to any email template will enable debugging, and display data key value pairs:

```
{% debug %}
```