attributes (/reference/attributes)

case (/reference/case)

code (/reference/code)

comments (/reference/comments)

conditionals (/reference/conditionals)

doctype (/reference/doctype)

extends (/reference/extends)

filters (/reference/filters)

includes (/reference/includes)

inheritance (/reference/inheritance)

iteration (/reference/iteration)

mixins (/reference/mixins)

plain text (/reference/plain-text)

tags (/reference/tags)

Code & Interpolation

Jade makes it possible to write inline JavaScript code in your templates. There are three types of code.

Unbuffered Code

Unbuffered code starts with `-` does not add any output directly, e.g.

```
- for (var x = 0; x < 3; x++)
  li item
```

```
<li>item</li>
<li>item</li>
<li>item</li>
```

Buffered Code

Buffered code starts with `=` and outputs the result of evaluating the JavaScript expression in the template. For security, it is first HTML escaped:

```
p
```

```
  = 'This code is <escaped>!'
```

```
<p>This code is &lt;escaped&gt;!
</p>
```

It can also be written inline with attributes, and supports the full range of JavaScript expressions:

```
p= 'This code is' + ' <escaped>!'
```

```
<p>This code is &lt;escaped&gt;!</p>
```

Unescaped Buffered Code

Unescaped buffered code starts with `!=` and outputs the result of evaluating the JavaScript expression in the template. This does not do any escaping, so is not safe for user input:

```
p
  != 'This code is <strong>not</strong> escaped!'
```

```
<p>This code is <strong>not</strong> escaped!
</p>
```

It can also be written inline with attributes, and supports the full range of JavaScript expressions:

```
p!= 'This code is <strong>not</strong> escaped!'
```

```
<p>This code is <strong>not</strong> escaped!</p>
```

Danger

Unescaped buffered code can be dangerous. You must be sure to sanatize any user inputs to avoid Cross Site Scripting (http://en.wikipedia.org/wiki/Cross-site_scripting)

Interpolation

Both plain-text and piped-text support interpolation. The following will output the user.name in the paragraph:

```
- var user = {name: 'Forbes Lindesay'}
p Welcome #{user.name}
```

```
<p>Welcome Forbes Lindesay</p>
```

It is generally recommended that you use `=` for buffering code unless it is a small amount of code in a large block.

Unescaped Interpolation

You can also use unescaped interpolation to output HTML:

```
- var user = {name: '<strong>Forbes Lindesay</strong>'}
p Welcome #{user.name}
p Welcome !{user.name}
```

```
<p>Welcome &lt;strong&gt;Forbes Lindesay&lt;/strong&gt;</p>
<p>Welcome <strong>Forbes Lindesay</strong></p>
```

Danger

Unescaped buffered code can be dangerous. You must be sure to sanatize any user inputs to avoid Cross Site Scripting (http://en.wikipedia.org/wiki/Cross-site_scripting)