

# Code

Pug allows you to write inline JavaScript code in your templates. There are three types of code: Unbuffered, Buffered, and Unescaped Buffered.

## Unbuffered Code ¶

Unbuffered code starts with `-`. It does not directly add anything to the output.

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

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

Pug also supports block unbuffered code:

```
-  
  var list = ["Uno", "Dos", "Tres",  
             "Cuatro", "Cinco", "Seis"]  
each item in list  
  li= item
```

```
<li>Uno</li>  
<li>Dos</li>  
<li>Tres</li>  
<li>Cuatro</li>  
<li>Cinco</li>  
<li>Seis</li>
```

## Buffered Code ¶

Buffered code starts with `=`. It evaluates the JavaScript expression and outputs the result. For security, buffered code is first HTML escaped.

```
p
```

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

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

It can also be written inline (after any attributes), and supports the full range of JavaScript expressions:

```
p= 'This code is' + ' <escaped>!'
p(style="background: blue")= 'A message with a ' + 'blue' +
```

```
<p>This code is &lt;escaped&gt;!</p>
<p style="background: blue">A message with a blue background
```

## Unescaped Buffered Code ¶

Unescaped buffered code starts with `!=`. It evaluates the JavaScript expression and outputs the result. Unescaped buffered code does not perform any escaping, so is unsafe for user input:

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

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

Unescaped buffered code 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>
```

### Caution

**Unescaped buffered code can be dangerous.** You must be sure to sanitize any user inputs to avoid cross-site scripting (XSS).

## Get an update when we release new features

Name (optional):

Email:

Sign Up

We respect your [email privacy](#).

