

Wednesday, 12 March 2014

Search...



Programming »

[Home](#) / [Programming](#) / [JavaScript](#) / [Paths and More Expressions – Handlebars.js Tutorial](#)

PATHS & MORE EXPRESSIONS



HANDLEBARS.JS TUTORIAL

Paths and More Expressions

PATHS AND MORE EXPRESSIONS – HANDLEBARS.JS TUTORIAL

In this last tutorial, we'll talk about paths and more expressions. So far, we've been dealing with plain text, but what if we had HTML inside our strings? Handlebars will naturally escape these values to prevent XSS injections. However, you can stop Handlebars from doing so simply by using 3 staches. Update your **menuData** variable to this.

```
1 var menuData = {  
2   info: '<strong>Bold!</strong>'  
3 };
```

Let's update our template to this.

```
1 <script id="menu-template" type="text/x-handlebars-template">  
2   {{info}}  
3 </script>
```

If you refresh your page, then you will see our **info** property escaped and displayed the same way we wrote it. We don't want this. We want the **** tags to bold our message. Update the expression to this.

```
1 | {{{info}}}
```

By using 3 staches on each side, then Handlebars will not escape the value and use it as is!

Paths

Handlebars supports paths. Let's say our **menuData** variable had a nested object like so.

```
1 var menuData = {  
2   author: {  
3     name: "Jack"  
4   }  
5 };
```

You would simply use dot notation inside our expression like so.

```
1 | {{{author.name}}}
```

SUBSCRIBE
To RSS Feed**12**
Fans**2**
Subscribers

ABOUT ME



Hey! I'm Jack and this is my personal blog. I love playing guitar and programming. I like to share my knowledge for free.

Popular

Recent

Comments

Tags



External JSON File – JSON Tutorial
July 24, 2013



Shortcode Attributes – WordPress Plugin Tutorial
March 19, 2013



WordPress Admin Forms – WordPress Plugin Tutorial
March 26, 2013



Getting Started – PHP OOP Tutorial
May 23, 2013



Conclusion – Advanced JavaScript Tutorial
September 6, 2013

You can also go back one by using the `../` path before the property name. Update your `menuData` variable to this.

```
1 var menuData = {
2   author: [
3     { name: "Jack" }
4   ],
5   lastName: " Brians"
6 };
```

Next, let's update the template to this.

```
1 <script id="menu-template" type="text/x-handlebars-template">
2   {{#each author}}
3     {{name}} {{../lastName}}
4   {{/each}}
5 </script>
```

Refresh your page and you should see the name **Jack Brians** outputted. Even though we're in a loop, we can still access the properties outside of this object by using the `../` path.

Comments

To create comments for your code, then you just insert a `!` before your expression like so.

```
1 {{!This is a comment }}
```

You can write whatever you want in here. It will not be outputted into your document. You can use regular HTML comments, but handlebars supplies you with their comment system just in case you need it.

Conclusion

Handlebars is a pretty powerful JavaScript templating system. It's actually the most popular templating system out there. To learn more about handlebars, then check out the main website here.

<http://handlebarsjs.com/>

[Download This Free Handlebars.js Example Source Code](#)

SHARE !

Tweet

0

8+1

0

Share

Pin it

ABOUT JASKO KOYN



Previous:
Custom Helpers – Handlebars.js
Tutorial

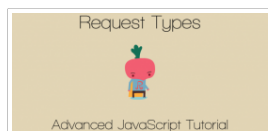
Next:
History of JavaScript – Advanced
JavaScript Tutorial



RELATED ARTICLES

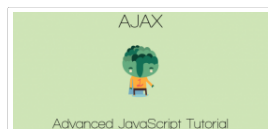


Conclusion – Advanced
JavaScript Tutorial
September 6, 2013

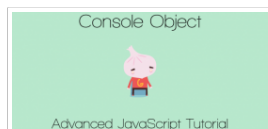


Request Types – Advanced
JavaScript Tutorial
September 6, 2013

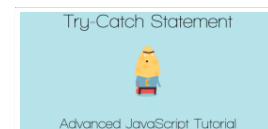
HTTP Headers – Advanced
JavaScript Tutorial
September 6, 2013



AJAX – Advanced JavaScript



Console Object – Advanced



Try-catch Statement –

Tutorial

September 6, 2013

JavaScript Tutorial

September 5, 2013

Advanced JavaScript Tutorial

September 5, 2013

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

Name *

Email *

Website