

Expressions in a JavaScript Switch Case

Posted At : December 29, 2015 1:29 PM | Posted By : Jon Hartmann

Related Categories: Experiments, Javascript

Today I stumbled across something I'd never seen before, and at first thought had to be an error - a switch statement with expressions embedded in the case statement:

Start with a boilerplate:

- jQuery
- Vue
- React
- React + JSX
- Preact
- TypeScript
- CoffeeScript
- SCSS
- CSS Grid

Close

Wh

the heck is that thing?

Lets take a look at a normal example of a Switch statement:

Start with a boilerplate:

Close

- jQuery
- Vue
- React
- React + JSX
- Preact
- TypeScript
- CoffeeScript
- SCSS
- CSS Grid

Compared to a normal Case statement, this thing is pretty weird. So weird that my first instinct was that this code was wrong and would break, but it was located in a very active part of the codebase - there was no way that it was failing to do what it was trying to do... but how was it resolving the right value?

Normally I think of a Switch as a kind of stand in for a key-value lookup. Ideally, the run time is smart enough to evaluate it like that rather than checking each value sequentially (I'll let some other blog tell you which browsers do it best), but there is no way to actually do that optimization if the value of the Case is an expression - at best you would have to calculate the answers to each case until you found one that matched... and since a Switch matches Cases sequentially, you could in theory stop with the first match.

I created a test to see if that was true:

Start with a boilerplate:

Close

- jQuery
- Vue
- React
- React + JSX
- Preact
- TypeScript
- CoffeeScript
- SCSS

As you can see from the result, my expectation was correct - the browser has to evaluate each statement individually and sequentially in order to find a match... which makes this the exact same execution as a simple If-Else block:

Start with a boilerplate:

Close

- jQuery
- Vue
- React
- React + JSX
- Preact
- TypeScript
- CoffeeScript
- SCSS
- CSS Grid

Which in the end means that this kind of Expression-in-Case setup is no better than the If-Else block. With nothing to show as an upside, applying the KISS methodology means we're better off with this less clever version:

Start with a boilerplate:

Close

- jQuery
- Vue
- React
- React + JSX
- Preact
- TypeScript
- CoffeeScript
- SCSS
- CSS Grid