Using constants as indices for Javascript Associative Arrays

Asked 8 years, 11 months ago Active 1 year, 1 month ago Viewed 10k times

19

3

I'm looking to create an associative array in JS, but use constants defined as part of the class as indices.

The reason I want this is so that users of the class can use the constants (which define events) to trigger actions.

Some code to illustrate:

```
STATE_NORMAL = 0;
STATE_NEW_TASK_ADDED = 0;
this.curr_state = STATE_NEW_TASK_ADDED;

this.state_machine = {
    /* Prototype:
    STATE_NAME: {
        EVENT_NAME: {
            "next_state": new_state_name,
            "action": func
        }
    }
    */

    STATE_NEW_TASK_ADDED : { // I'd like this to be a constant
        this.EVENT_NEW_TASK_ADDED_AJAX : {
            "next_state": STATE_NEW_TASK_ADDED,
            "action" : function() {console.log("new task added");},
        }
    }
}

// Public data members.
// These define the various events that can happen.
this.EVENT_NEW_TASK_ADDED_AJAX = 0;
this.EVENT_NEW_TASK_ADDED_AJAX = 1;
```

I'm having trouble getting this to work. I'm not too great with JS, but it looks like no matter what I do, the array gets defined with strings and not constants. Is there any way to force the array to use the constants?

javascript constants associative-array

asked Nov 7 '10 at 9:44

Edan Maor
6,290 14 56 83

2 Answers

37

The problem here, actually, is that you can't use a value for the key part when you're defining an object **literally**.

That is to say, this uses the constant values as expected:

```
var CONSTANT_A = 0, CONSTANT_B = 1;
var state_machine = {};
state_machine[CONSTANT_A] = "A";
state_machine[CONSTANT_B] = "B";
console.log(state_machine[0]); // => A
console.log(state_machine[1]); // => B
```

Run code snippet Expand snippet

But this won't work as expected, instead using the string `CONSTANT_A` as key:

```
var CONSTANT_A = 0, CONSTANT_B = 1;
var state_machine = {
    CONSTANT_A: "A",
    CONSTANT_B: "B",
};
console.log(state_machine[0]); // => undefined
console.log(state_machine["CONSTANT_A"]); // => A
console.log(state_machine.CONSTANT_A); // => A
```

JavaScript has a shorthand to define object literals where you can omit the double-quotes around keys. Expressions can't be used, so `CONSTANT_A` won't be evaluated.

See also @Kristian's answer below re: ES6/modern JS, essentially making what you want possible.

edited Aug 14 '18 at 5:35

answered Nov 7 '10 at 9:54

Ashe

14.9k 4 42 68

33

In ES6 you can use computed values for object keys.

```
var CONSTANT_A = 0, CONSTANT_B = 1
var state_machine = {
    [CONSTANT_A]: function () {
        return 'a'
    },
    [CONSTANT_B]: function () {
        return 'b'
    }
};

console.log(state_machine)
```

Run code snippet   Expand snippet

This does not work in IE 11 nor in safari browsers: https://kangax.github.io/compat-table/es6/#test-object_literal_extensions_computed_properties

edited Jun 19 '16 at 11:14

answered Mar 1 '16 at 14:20

Kristian

663 1 8 15

Thanks! You saved me a lot of time. – Mihir Jun 19 '16 at 8:40

Ah. ES6 is so great – Félix Gagnon-Grenier Sep 30 '16 at 15:28