What is the meaning of an Underscore in javascript function parameter?

Asked 4 years, 8 months ago   Active 1 month ago   Viewed 8k times

▲

18

▼

★

2

I was going through the code of one of the chart library written in javascript, wherein I've seen passing underscore(_) as a function parameter. What does that mean?

e.g.

```
chart.x = function(_) {
    if (!arguments.length) return lines.x;
    lines.x(_);
    lines2.x(_);
    return chart;
};
```

| Run code snippet | Copy snippet to answer | Expand snippet |

Can someone please update on this...Thanks.

javascript

edited Aug 9 at 9:30   asked Dec 24 '14 at 12:33

[⊔⌐] Flip   [🔲] Azhar
       2,501 2   21   47   **200** 1   10

so @GabyakaG.Petrioli if you call that function with any parameter irrespective of type or no. of parameters..it will call that function? – Azhar   Dec 24 '14 at 12:42

In the specific function if you pass no parameters it will return lines.x right away, if you pass more than one, then it will use the first only and pass it to lines.x and lines2.x and finally return the chart . – Gabriele Petrioli Dec 24 '14 at 12:44

1 Answer

In this case `_` is just a function parameter - a single underscore is a convention used by some programmers to indicate "ignore this binding/parameter".

30

Since JavaScript doesn't do parameter-count checking the parameter could have been omitted entirely. Such a "throw-away" identifier is found more commonly in other languages, but consider a case like `arr.forEach(function (_, i) {..})` where `_` indicates the first parameter is not to be used.

edited Nov 28 '18 at 12:05   answered Dec 24 '14 at 12:35

Billal Begueradj   sagar43

6,546   14   39   64   **1,825**   2   19   39

---

2   If it was omitted entirely, then he would have nothing to pass to `lines.x()` and `lines2.x(_)` . – user663031 Dec 24 '14 at 12:59

---

1   To confirm my understanding, passing `_` in this case is equivalent to passing `null` , and `_` is used just for aesthetics/readability? – Mark Jackson Dec 10 '18 at 0:12

---

1   @MarkJackson The `_` will still hold a value. You can test this by declaring a function with a parameter `_` that logs that parameter, and you will see it logs whatever was passed in – Felipe May 20 at 17:01

@MarkJackson: No, `_` isn't special; it's a valid identifier in JS, just like `x` or `foo` . These definitions are equivalent: a) `function f(x) { return x; }` , b) `function f(_) { return _; }` . I'd say @sagar43's answer is incorrect, because `_` *is* in fact used in the body of `chart.x` , so omitting the parameter (`chart.x = function() { ... }`) would change the function's semantics. – Simon Alling Aug 26 at 13:55 ✎

Got a question that you can't ask on public Stack Overflow? Learn more about sharing private information with Stack Overflow for Teams.   ✕