

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS What's this?

Q&A for Work

How do I format numbers using JavaScript?

[Ask Question](#)

I want to format numbers using JavaScript.

91

For example:

```
10      => 10.00
100     => 100.00
1000    => 1,000.00
10000   => 10,000.00
100000  => 100,000.00
```

javascript

share improve this question

edited Oct 29 '17 at 13:38

Peter Mortensen
14.3k ● 19 ● 88 ● 115

asked Apr 20 '11 at 13:36

Kanak Vaghela
2,439 ● 6 ● 26 ● 36

10 Are you sure you mean "1,00,000"? Perhaps "100,000". – Lightness Races in Orbit Apr 20 '11 at 13:47

1 possible duplicate of [How to print a number with commas as thousands separators in JavaScript](#) – Johan Feb 19 '13 at 11:29

4 @LightnessRacesinOrbit, fun fact - in India, they group numbers like e.g. 12,34,567.89 – Nils Nov 14 '13 at 14:11

13 @Nils That's not fun :(– Lightness Races in Orbit Nov 14 '13 at 14:16

1 @Nils @BoundaryImposition: `(1234567.89).toLocaleString('hi-IN')` is "12,34,567.89" – Michael Mar 24 '17 at 18:39[show 1 more comment](#)

17 Answers

active oldest votes

If you want to use built-in code, you can use `toLocaleString()` with `minimumFractionDigits`, although browser compatibility for the extended options on `toLocaleString()` is limited.

135

```
var n = 100000;
var value = n.toLocaleString(
  undefined, // leave undefined to use the browser's locale,
  // or use a string like 'en-US' to override it.
  { minimumFractionDigits: 2 }
);
console.log(value);
// In en-US, logs '100,000.00'
// In de-DE, logs '100.000,00'
// In hi-IN, logs '1,00,000.00'
```

Run code snippet

Expand snippet

If you're using Node.js, you [will need to](#) `npm install` the `intl` package.

share improve this answer

edited Dec 18 '17 at 15:09

answered May 7 '15 at 16:05

asked 8 years, 3 months ago

viewed 202,533 times

active 15 days ago

Blog

Google got Looker. Salesforce bought Tableau. But open source tools are...

What a very bad day at work taught me about building Stack Overflow's community

Featured on Meta

Custom Filters release announcement

Unicorn Meta Zoo #5: Interview with Cesar M

Work from anywhere

Software Developer - Java/Spring/React - Latin America (Remote)

Dotsub No office location

REMOTE

java reactjs

Full-stack Java/JEE Engineer - REMOTE

SemanticBits No office location

REMOTE

java spring

Beyond GitHub? - App Template Developer(s)

Koji - Web App Development Platform No office location

\$120K - \$200K REMOTE

javascript reactjs

High response rate

Frontend Javascript Developer

Self Decode No office location

\$50K - \$100K REMOTE

css javascript



3,932 3 43 63

3 It does not work in nodejs. – Jian Weihang Aug 10 '17 at 8:18

1 I wanted to use built in code and this got me there. The locales options is a small issue I'm willing to live with. I think this should be the accepted answer because it takes into account the locale nuances of number rendition.
– Alexander Dixon Aug 10 '17 at 19:51

[add a comment](#)

[Short solution:](#)

33

```
var n = 1234567890;
String(n).replace(/(.)(?=(\d{3})+$)/g, '$1,')
// "1,234,567,890"
```

[share](#) [improve](#) [this answer](#)

answered Sep 2 '15 at 13:50



Adrian Panasiuk
5,453 4 26 50

[add a comment](#)

[Use](#)

30

```
num = num.toFixed(2);
```

Where 2 is the number of decimal places

[Edit](#):

Here's the function to format number as you want

```
function formatNumber(number)
{
    number = number.toFixed(2) + '';
    x = number.split('.');
    x1 = x[0];
    x2 = x.length > 1 ? '.' + x[1] : '';
    var rx = /(\d+)(\d{3})/;
    while (rx.test(x1)) {
        x1 = x1.replace(rx, '$1' + ',' + '$2');
    }
    return x1 + x2;
}
```

Sorce: www.mredkj.com

[share](#) [improve](#) [this answer](#)

edited Apr 20 '11 at 13:48

answered Apr 20 '11 at 13:39



Ortiga
6,408 5 27 58

2 But using .toFixed(2) i will not get the result for 10000 and 100000 as 10,000.00 and 1,00,000.00 –
Kanak Vaghela Apr 20 '11 at 13:40

[add a comment](#)

Due to the bugs found by JasperV — good points! — I have rewritten my old code. I guess I only ever used this for positive values with two decimal places.

High response rate

Work remotely - from home or wherever you choose.

[Browse remote jobs](#)

[Linked](#)

9 How do I add a thousand seperator to a number in JavaScript?

0 Adding comma-separated numbers to Javascript output

-1 How to add commas in number in javascript

-1 Insert comma(.) into a value using jQuery

0 Regular expression for currency formate

1603 How can I format numbers as currency string in JavaScript?

174 Using bitwise OR 0 to floor a number

14 Using .toLocaleString() in Node.js

5 Regex to separate thousands with comma and keep two decimals

-1 How to format a number on 'onblur' function in javascript?

[see more linked questions...](#)

[Related](#)

7648 How do JavaScript closures work?

7318 How do I check if an element is hidden in jQuery?

5616 How do I remove a property from a JavaScript object?

5670 Which equals operator (== vs ===) should be used in JavaScript comparisons?

7729 How do I redirect to another webpage?

1805 JavaScript equivalent to printf/String.Format

4712 How do I include a JavaScript file in another JavaScript file?

7196 What does "use strict" do in JavaScript, and what is the reasoning behind it?

7433 How to check whether a string contains a substring in JavaScript?

7279 How do I remove a particular element from an array in JavaScript?

[Hot Network Questions](#)

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS What's this?

Q&A for Work

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS What's this?

Q&A for Work

Depending on what you are trying to achieve, you may want rounding or not, so here are two versions split across that divide.

First up, with rounding.

I've introduced the `toFixed()` method as it better handles rounding to specific decimal places accurately and is well support. It does slow things down however.

This version still detaches the decimal, but using a different method than before. The `w|0` part removes the decimal. For more information on that, this is a [good answer](#). This then leaves the remaining integer, stores it in `k` and then subtracts it again from the original number, leaving the decimal by itself.

Also, if we're to take negative numbers into account, we need to while loop (skipping three digits) until we hit `b`. This has been calculated to be 1 when dealing with negative numbers to avoid putting something like `-,100.00`

The rest of the loop is the same as before.

```
function formatThousandsWithRounding(n, dp){
    var w = n.toFixed(dp), k = w|0, b = n < 0 ? 1 : 0,
        u = Math.abs(w-k), d = (''+u.toFixed(dp)).substr(2, dp),
        s = ''+k, i = s.length, r = '';
    while ( (i-3) > b ) { r = ',' + s.substr(i, 3) + r; }
    return s.substr(0, i + 3) + r + (d ? '.'+d: '');
};
```

In the snippet below you can edit the numbers to test yourself.

▶ [Show code snippet](#)

Now the other version, without rounding.

This takes a different route and attempts to avoid mathematical calculation (as this can introduce rounding, or rounding errors). If you don't want rounding, then you are only dealing with things as a string i.e. 1000.999 converted to two decimal places will only ever be 1000.99 and not 1001.00.

This method avoids using `.split()` and `RegExp()` however, both of which are very slow in comparison. And whilst I learned something new from Michael's answer about `toLocaleString`, I also was surprised to learn that it is — by quite a way — the slowest method out of them all (at least in Firefox and Chrome; Mac OSX).

Using `lastIndexOf()` we find the possibly existent decimal point, and from there everything else is pretty much the same. Save for the padding with extra 0s where needed. This code is limited to 5 decimal places. Out of my test this was the faster method.

```
var formatThousandsNoRounding = function(n, dp){
    var e = '', s = e+n, l = s.length, b = n < 0 ? 1 : 0,
        i = s.lastIndexOf('.'), j = i == -1 ? l : i,
        r = e, d = s.substr(j+1, dp);
    while ( (j-3) > b ) { r = ',' + s.substr(j, 3) + r; }
    return s.substr(0, j + 3) + r +
        (dp ? '.' + d + (''.length < dp ?
            ('0000').substr(0, dp - d.length):e):e);
};
```

▶ [Show code snippet](#)

I'll update with an in-page snippet demo shortly, but for now here is a fiddle:

<https://jsfiddle.net/bv2ort0a/2/>

- ⌚ Why is this guy handcuffed censored?
- 📊 Why do we need an estimator to be consistent?
- 🔫 Which modern firearm should a time traveler bring to be easily reproducible for a historic civilization?
- ⚡ Transistor power dissipation rating
- 🌐 Improving an O(N^2) function (all entities iterating over all other entities)
- ⌚ What is the simplest instruction set that has a C++/C compiler to write an emulator for?
- 🎹 Get Chord Name From A Given Set of Notes
- 🐧 Linux ext4 restore file and directory access rights after bad backup/restore
- 🗨 Why aren't there any women super Grandmasters (GMs)?

- ↳ Why should fork() have been designed to return a file descriptor?

- 🇫🇷 "Je suis petite, moi?", purpose of the "moi"?
- gMaps What could make large expeditions ineffective for exploring territory full of dangers and valuable resources?
- Почем Doesn't the Schrödinger's cat inside the box cause the probability wave function to collapse long before a human opens the box?
- Почем Is there a standard frequency for wandering monster checks in a dungeon?
- Почем What is the intuition for higher homotopy groups not vanishing?
- Почем Last-minute canceled work-trip means I'll lose thousands of dollars on planned vacation
- Почем Old book about a young man marooned in space who is drawn to an alien shipwreck by a ring he is wearing

- Почем Found old paper shares of Motorola Inc that has since been broken up

- Почем Weight functions in graph algorithms

- Почем Why isn't a binary file shown as 0s and 1s?

- Почем Three Subway Escalators

- Почем Do pedestrians imitate automotive traffic?

- Почем How was Luke's prosthetic hand in Episode V filmed?

- Почем Function over a list that depends on the index

- RSS Question feed

[Tags](#)[Users](#)[Jobs](#)

TEAMS What's this?

Q&A for Work

Old Method

Why use RegExp for this? — don't use a hammer when a toothpick will do i.e. use string manipulation:

```
var formatThousands = function(n, dp){
  var s = ''+(Math.floor(n)), d = n % 1, i = s.length, r = '';
  while ( (i -= 3) > 0 ) { r = ',' + s.substr(i, 3) + r; }
  return s.substr(0, i + 3) + r +
    (d ? '.' + Math.round(d * Math.pow(10, dp || 2)) : '');
};
```

walk through

```
formatThousands( 1000000.42 );
```

First strip off decimal:

```
s = '1000000', d = ~ 0.42
```

Work backwards from the end of the string:

```
',' + '000'
',' + '000' + ',000'
```

Finalise by adding the leftover prefix and the decimal suffix (with rounding to `dp` no. decimal points):

```
'1' + ',000,000' + '.42'
```

fiddlesticks

<http://jsfiddle.net/XC3sS/>

share improve this answer

edited May 23 '17 at 12:34

answered Dec 12 '13 at 14:01



Community

1 • 1



Pebbl

25.1k 5 54 60

This deserves more upvotes! +1 for walkthrough instead of just some code stolen from a different place – [PieBie](#) Mar 24 '16 at 16:05

3 There are 2 bugs in this one: When using a negative number it fails with the decimals: -1000000.42 becomes -1,000,001.-42 and a one is added because the `floor` method. And it has another decimal error: when I process 1000000.999 with `dp=2` it becomes 1,000,000.100, so 3 decimals and my number is not rounded – [JasperV](#) Apr 7 '16 at 7:18

2 @JasperV, Very good points... cheers for the heads up. Not spotted those problems in any of my previous use-cases. Should be fixed now however. – [Pebbl](#) Apr 11 '16 at 22:11

[add a comment](#)

On browsers that support the [ECMAScript® 2016 Internationalization API Specification \(ECMA-402\)](#), you can use an `Intl.NumberFormat` instance:

```
var nf = Intl.NumberFormat();
var x = 4200000;
console.log(nf.format(x)); // 42,000,000 in many locales
// 42.000.000 in many other locales
```

21

▶ [Show code snippet](#)

share improve this answer

edited Aug 18 '16 at 7:42

answered Aug 18 '16 at 7:31

 **T.J. Crowder**
724k • 133 • 1306 •
1380

That's the shortest and easiest solution, without including any plugin. – [gael](#) Feb 11 at 22:55

[add a comment](#)

Use the Number function `toFixed` and this function to add the commas.

6

```
function addCommas(nStr) {
    nStr += '';
    var x = nStr.split('.');
    var x1 = x[0];
    var x2 = x.length > 1 ? '.' + x[1] : '';
    var rgx = /(\d{1})(\d{3})/;
    while (rgx.test(x1)) {
        x1 = x1.replace(rgx, '$1' + ',' + '$2');
    }
    return x1 + x2;
}
n = 10000;
r = n.toFixed(2); //10000.00

addCommas(r); // 10,000.00
```

<http://www.mredkj.com/javascript/numberFormat.html>

share improve this answer

edited Apr 20 '11 at 13:48

answered Apr 20 '11 at 13:39

 **John Giotta**
12.7k • 4 • 38 • 71

But using `.toFixed(2)` i will not get the result for 10000 and 100000 as 10,000.00 and 1,00,000.00 – [Kanak Vaghela](#) Apr 20 '11 at 13:40

Oh! I've updated it – [John Giotta](#) Apr 20 '11 at 13:43

2 if you're copying code from somewhere else, giving reference-link would be nice:

mredkj.com/javascript/numberFormat.html – [oezi](#) Apr 20 '11 at 13:46

@oezi - Sure. @Chris - It was updated to show comma formatting. – [John Giotta](#) Apr 20 '11 at 13:49

thanks for this answer :) – [dada](#) Jul 12 '17 at 0:16

[add a comment](#)

▶ I think with this [jQuery-numberformatter](#) you could solve your problem.

Of course, this is assuming that you don't have problem with using jQuery in your project. Please notice that the functionality is tied to the `blur` event.

6

```
$("#salary").blur(function(){
    $(this).parseNumber({format:"#,###.00", locale:"us"});
    $(this).formatNumber({format:"#,###.00", locale:"us"});
});
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
<script src="https://cdn.jsdelivr.net/gh/timdown/jshashable/hashtable.js"></script>
<script src="https://cdn.jsdelivr.net/gh/hardhub/jquery-numberformatter/src/jquery.numberf
<input type="text" id="salary">
```

Run code snippet Expand snippet

share improve this answer

edited Feb 25 at 0:01

answered Apr 20 '11 at 13:43



- 1 Google Code [was shut down on 2016-01-15](#). You may want to update the broken link, "**404. That's an error.**" (many projects on Google Code have been moved to [GitHub](#)). – Peter Mortensen Oct 29 '17 at 13:53 ↗

thank you @PeterMortensen! updated the link! – Cacho Santa Feb 6 '18 at 4:55

why the downvote? care to explain? – Cacho Santa May 18 '18 at 16:57

updated with the code snippet to make it simpler to view the results and the script on the works. – Cacho Santa Feb 25 at 0:03

add a comment

4

```
function numberWithCommas(x) {
  x=String(x).toString();
  var afterPoint = '';
  if(x.indexOf('.') > 0)
    afterPoint = x.substring(x.indexOf('.'),x.length);
  x = Math.floor(x);
  x=x.toString();
  var lastThree = x.substring(x.length-3);
  var otherNumbers = x.substring(0,x.length-3);
  if(otherNumbers != '')
    lastThree = ',' + lastThree;
  return otherNumbers.replace(/\B(?=(\d{3})+(?!\d))/g, ",") + lastThree + afterPoint;
}

console.log(numberWithCommas(100000));
console.log(numberWithCommas(1000000));
```

Run code snippet Expand snippet

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS

What's this?

Q&A for Work

Output

1,00,000
1,00,00,000

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS

What's this?

Q&A for Work

▲ This is an article about your problem. Adding a thousands-separator is not built in to JavaScript, so

you'll have to write your own function like this (example taken from the linked page):

3

```
function addSeparator(nStr){  
    nStr += '';  
    x = nStr.split('.');  
    x1 = x[0];  
    x2 = x.length > 1 ? '.' + x[1] : '';  
    var rx = /(\d{1})(\d{3})/;  
    while (rx.test(x1)) {  
        x1 = x1.replace(rx, '$1' + ',' + '$2');  
    }  
    return x1 + x2;  
}
```

[share](#) [improve this answer](#)

edited Oct 29 '17 at 13:39

 Peter Mortensen
14.3k • 19 • 88 • 115

answered Apr 20 '11 at 13:42

 oezi
43.5k • 9 • 86 • 111

[add a comment](#)

Or you could use the [sugar.js](#) library, and the [format](#) method:

2

format(place = 0 , thousands = ',' , decimal = '.') Formats the number to a readable string. If place is undefined, will automatically determine the place. thousands is the character used for the thousands separator. decimal is the character used for the decimal point.

Examples:

```
(56782).format() > "56,782"  
(56782).format(2) > "56,782.00"  
(4388.43).format(2, ',') > "4 388.43"  
(4388.43).format(3, '.', ',') > "4.388,430"
```

[share](#) [improve this answer](#)

answered Jun 30 '14 at 8:32

 andersh
5,213 • 3 • 30 • 27

[add a comment](#)

If you are looking for a formatting with limit to three significant digits, for example:

2

1,23,45,67,890.123

Use:

```
number.toLocaleString('en-IN');
```

Working Example:

```
let number = 1234567890.123;  
  
document.write(number.toLocaleString('en-IN'));
```

[Run code snippet](#)

[Expand snippet](#)

Home

PUBLIC

 Stack Overflow

Tags

Users

Jobs

TEAMS

What's this?

 Q&A for Work

Home

PUBLIC

 Stack Overflow

Tested in Chrome v60.0.3112.113

[Tags](#)[Users](#)[Jobs](#)

TEAMS What's this?

Q&A for Work

Source: [Number.prototype.toLocaleString\(\)](#) | MDN[share](#) [improve this answer](#)

answered Sep 11 '17 at 18:13

 Rahul Desai
11.2k ● 10 ● 65 ● 107[add a comment](#)

This will get you your comma seperated values as well as add the fixed notation to the end.

1

```
nStr="1000";
nStr += '.';
x = nStr.split('.');
x1 = x[0];
x2 = x.length > 1 ? '.' + x[1] : '';
var regex = /(\d+)(\d{3})/;
while (regex.test(x1)) {
    x1 = x1.replace(regex, '$1' + ',' + '$2');
}
commaSeparated = x1 + x2 + ".00";
alert(commaSeparated);
```

[Source](#)[share](#) [improve this answer](#)

edited Apr 20 '11 at 14:20

answered Apr 20 '11 at 14:00

 clamchoda
2,072 ● 2 ● 25 ● 58[add a comment](#)

Home

PUBLIC

Stack Overflow

[Tags](#)[Users](#)[Jobs](#)

TEAMS What's this?

Q&A for Work

Let me also throw my solution in here. I've commented each line for ease of reading and also provided some examples, so it may look big.

1

```
function format(number) {
    var decimalSeparator = ".";
    var thousandSeparator = ",";

    // make sure we have a string
    var result = String(number);

    // split the number in the integer and decimals, if any
    var parts = result.split(decimalSeparator);

    // if we don't have decimals, add .00
    if (!parts[1]) {
        parts[1] = "00";
    }

    // reverse the string (1719 becomes 9171)
    result = parts[0].split("").reverse().join("");

    // add thousand separator each 3 characters, except at the end of the string
    result = result.replace(/(\d{3})(?!\$)/g, "$1" + thousandSeparator);

    // reverse back the integer and replace the original integer
    parts[0] = result.split("").reverse().join("");

    // recombine integer with decimals
    return parts.join(decimalSeparator);
}

document.write("10 => " + format(10) + "<br/>");
document.write("100 => " + format(100) + "<br/>");
```

```
document.write("1000 => " + format(1000) + "<br/>");  
document.write("10000 => " + format(10000) + "<br/>");  
document.write("100000 => " + format(100000) + "<br/>");  
document.write("100000.22 => " + format(100000.22) + "<br/>");
```

[Run code snippet](#)[Expand snippet](#)

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS

What's this?

Q&A for Work

share improve this answer

edited Oct 28 '15 at 14:32

answered Apr 24 '15 at 8:40

 Nicu Surdu
4,933 ● 8 ● 58 ● 86

add a comment

If you're using jQuery, you could use the [format](#) or [number format](#) plugins.

0

share improve this answer

answered Apr 20 '11 at 13:43

 Ferruccio
82.2k ● 35 ● 202 ● 286

add a comment

0

```
function formatNumber1(number) {  
    var comma = ',',  
        string = Math.max(0, number).toFixed(0),  
        length = string.length,  
        end = /^[^d{4}]*$/.test(string) ? length % 3 : 0;  
    return (end ? string.slice(0, end) + comma : '') + string.slice(end).replace(/(\d{3})(?=\d),/g, '$1' );  
  
    function formatNumber2(number) {  
        return Math.max(0, number).toFixed(0).replace(/(?:\d{3})+$)(?!^)/g, ',');  
    }  
}
```

Source: <http://jsperf.com/number-format>

share improve this answer

edited Apr 20 '11 at 13:48

answered Apr 20 '11 at 13:37

 Lightness Races in Orbit
308k ● 57 ● 510 ● 853 Adnan
19.2k ● 17 ● 68 ● 104

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS

What's this?

Q&A for Work

But using .toFixed(2) i will not get the result for 10000 and 100000. – [Kanak Vaghela](#) Apr 20 '11 at 13:38 ↗I just updated my answer – [Adnan](#) Apr 20 '11 at 13:42This doesn't really answer the question. These functions round the numbers, they don't add two decimals. – [Albert MN](#). Dec 10 '17 at 20:21None of these solutions work for negative numbers – [Vamshi Gudipati](#) Apr 27 '18 at 17:06

add a comment

0

This is about 3 times faster version of the accepted answer. It doesn't create array and avoids object creation and string concatenation for whole numbers at the end. This might be useful if you render lots of values e.g. in a table.

```
function addThousandsSeparators(number) {  
    var whole, fraction;  
    var decIndex = number.lastIndexOf('.');  
    if (decIndex > 0) {  
        whole = number.substr(0, decIndex)  
        fraction = number.substr(decIndex + 1);  
    } else {  
        whole = number;  
        fraction = '';  
    }  
    var result = whole.replace(/\B(?=(\d{3})+(?!\d))/g, ',') + fraction;  
    return result;  
}
```

```
        traction = number.substr(decindex)
    } else {
        whole = number
    }
var regex = /(\d+)(\d{3})/
while (regex.test(whole)) {
    whole = whole.replace(regex, '$1' + ',' + '$2')
}
return fraction ? whole + fraction : whole
}
```

share improve this answer

answered Mar 9 '15 at 10:58

 petrsyn
3,668 ● 1 ● 32 ● 45

Home

PUBLIC

 Stack Overflow

Tags

Users

Jobs

TEAMS

What's this?

 Q&A for Work



0

```
function formatThousands(n,dp,f) {
    // dp - decimal places
    // f - format >> 'us', 'eu'
    if (n == 0) {
        if(f == 'eu') {
            return "0," + "0".repeat(dp);
        }
        return "0." + "0".repeat(dp);
    }

    /* round to 2 decimal places */
    //n = Math.round( n * 100 ) / 100;
    var s = ''+(Math.floor(n)), d = n % 1, i = s.length, r = '';
    while ( (i -= 3) > 0 ) { r = ',' + s.substr(i, 3) + r; }
    var a = s.substr(0, i + 3) + r + (d ? '.' + Math.round((d+1) * Math.pow(10,dp)) : '');
    /* change format from 20,000.00 to 20.000,00 */
    if (f == 'eu') {
        var b = a.toString().replace(".", "#");
        b = b.replace(",",".");
        return b.replace("#", ",");
    }
    return a;
}
```

share improve this answer

answered Jul 5 at 23:02

 The Geremy
1 ● 1

What's different about this answer compared to the other answers on the question? Can you elaborate the meaning of the code only answers? – Akin Okegbile Jul 5 at 23:27

add a comment

Home

PUBLIC

 Stack Overflow

Tags

Users

Jobs

TEAMS

What's this?

 Q&A for Work

Your Answer



Home

PUBLIC

 Stack Overflow

Tags

Users

Jobs

TEAMS

What's this?

 Q&A for Work

Sign up or [log in](#)

 Sign up using Google

 Sign up using Facebook

 Sign up using Email and Password

Post as a guest

Name

Email

Required, but never shown

[Post Your Answer](#)

By clicking "Post Your Answer", you agree to our [terms of service](#), [privacy policy](#) and [cookie policy](#)

Not the answer you're looking for? Browse other questions tagged [javascript](#) or [ask your own question](#).



STACK OVERFLOW

[Questions](#)
[Jobs](#)
[Developer Jobs Directory](#)
[Salary Calculator](#)
[Help](#)
[Mobile](#)

PRODUCTS

[Teams](#)
[Talent](#)
[Advertising](#)
[Enterprise](#)

COMPANY

[About](#)
[Press](#)
[Work Here](#)
[Legal](#)
[Privacy Policy](#)
[Contact Us](#)

STACK EXCHANGE NETWORK

[Blog](#) [Facebook](#) [Twitter](#) [LinkedIn](#)

[Technology](#) 
[Life / Arts](#) 
[Culture / Recreation](#) 
[Science](#) 
[Other](#) 

site design / logo © 2019 Stack Exchange Inc; user contributions licensed under [cc by-sa 3.0](#) with attribution required.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#). 