Published in Bits and Pieces

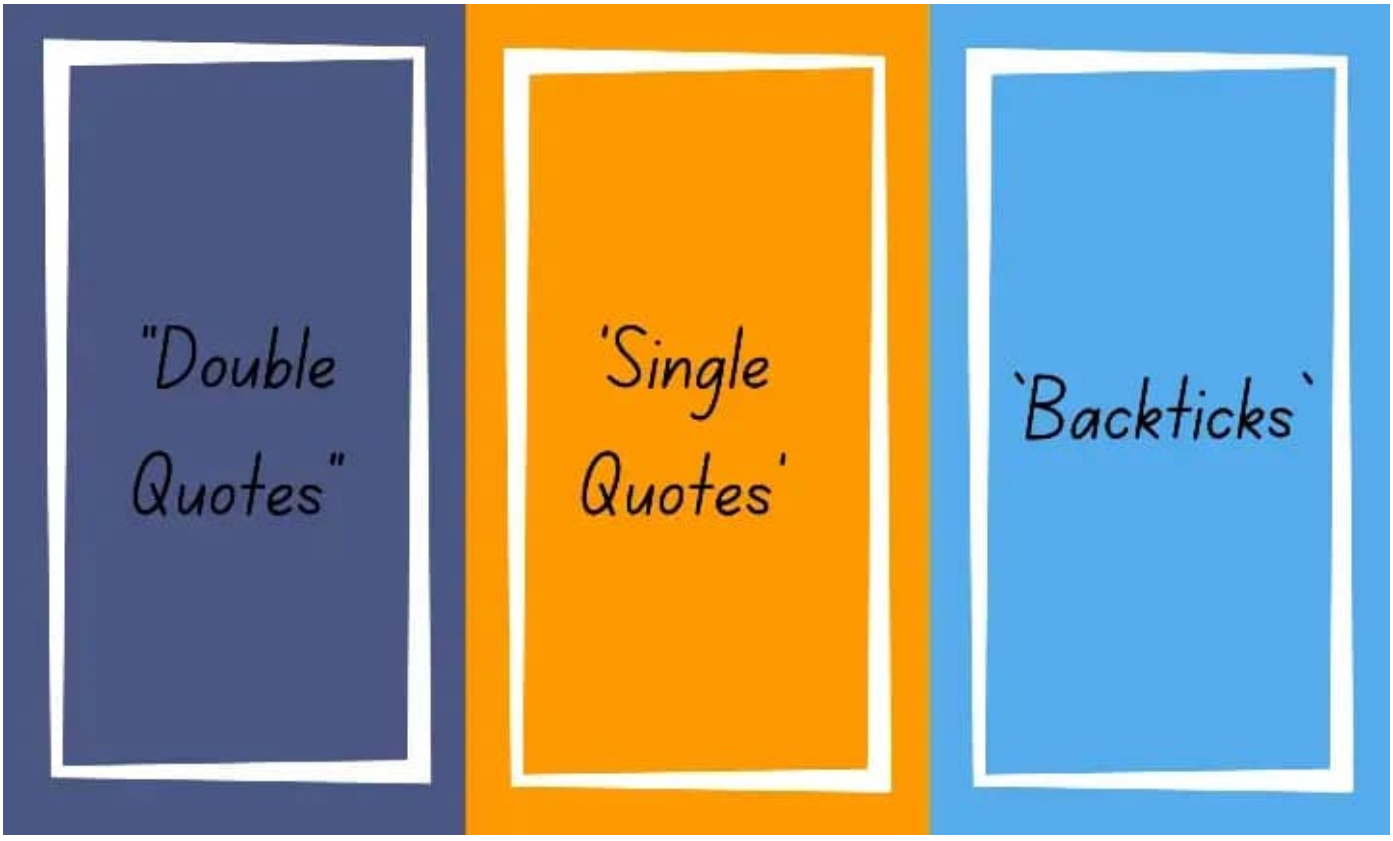
Chameera Dulanga Follow

Jun 15, 2021 · 5 min read · Listen



# "Double Quotes" vs 'Single Quotes' vs `Backticks` in JavaScript

Difference Between using 'Single Quotes', "Double Quotes", and `Backticks` in JavaScript Strings.



In JavaScript, single quotes (`''`) and double quotes (`""`) are used to create string literals. Most developers use single or double quotes as they prefer, and sometimes they let their code formatters decide what to use.

So, I wanted to find the real difference between these 2, and this article will discuss similarities, differences, and essential tips you should remember while using these quotes.

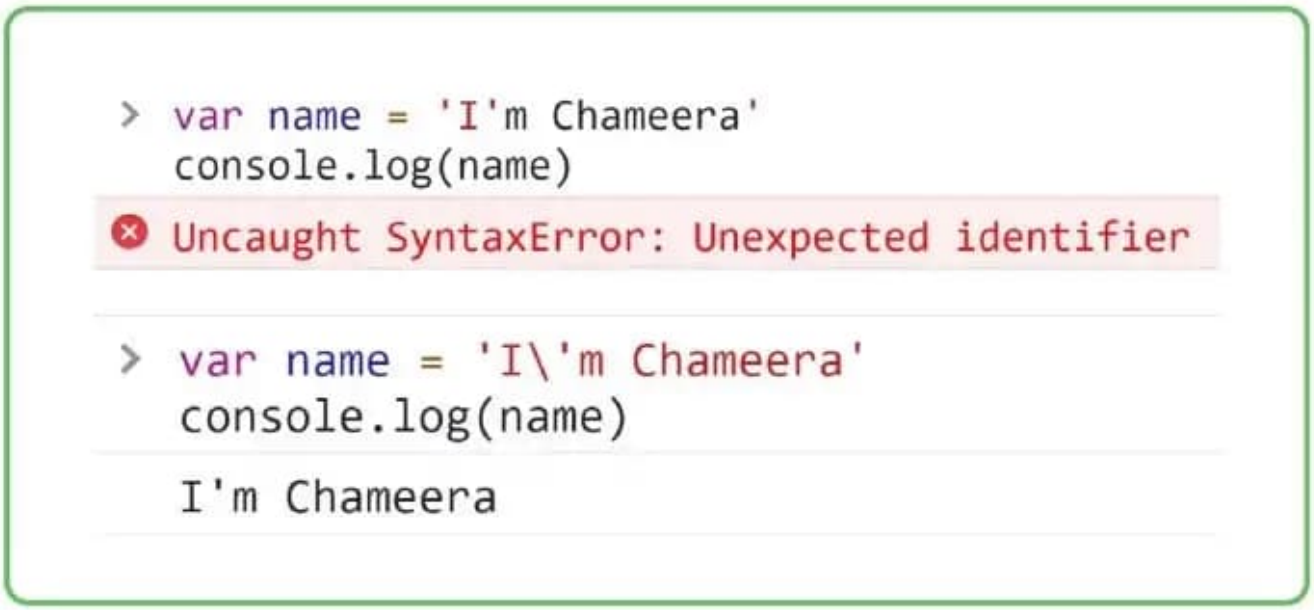
. . .

### 'Single Quotes' vs. "Double Quotes"

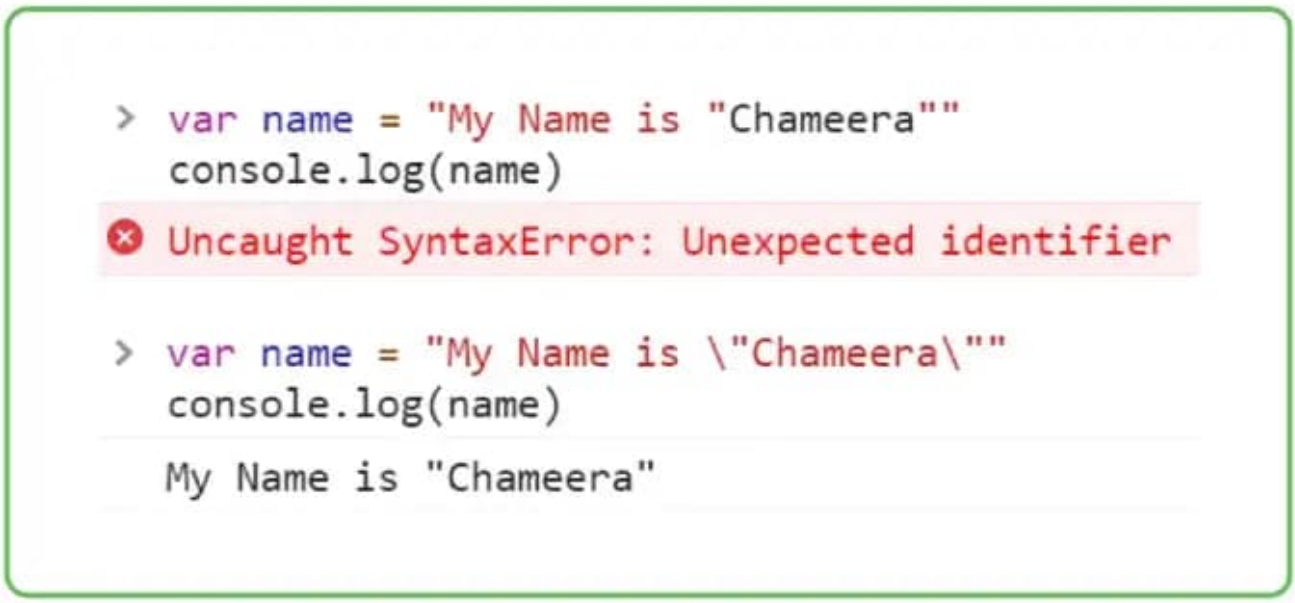
As I mentioned, both of these are used to create string literals (values) in JavaScript, and they act in exactly the same way.

> The only noticeable difference between single quotes and double quotes comes into play when we have to escape characters.

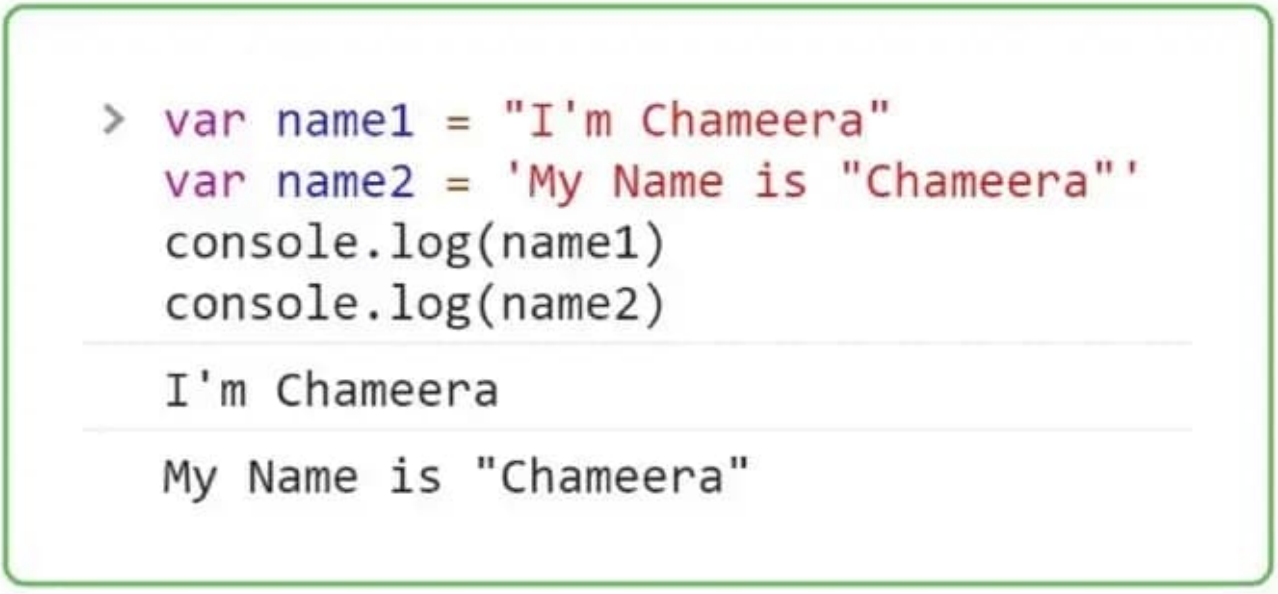
If you use single quotes to create a string, you can not use single quotes within that string without escaping them using a backslash (`\`).



The same theory applies to double quotes, and you have to use a backslash to escape any double quotes inside double quotes.




However you can use single quotes inside double quotes or double quotes inside single quotes without escaping.



I personally prefer to use double quotes to create strings in JavaScript. It makes strings more readable since we don't use backslashes.

. . .

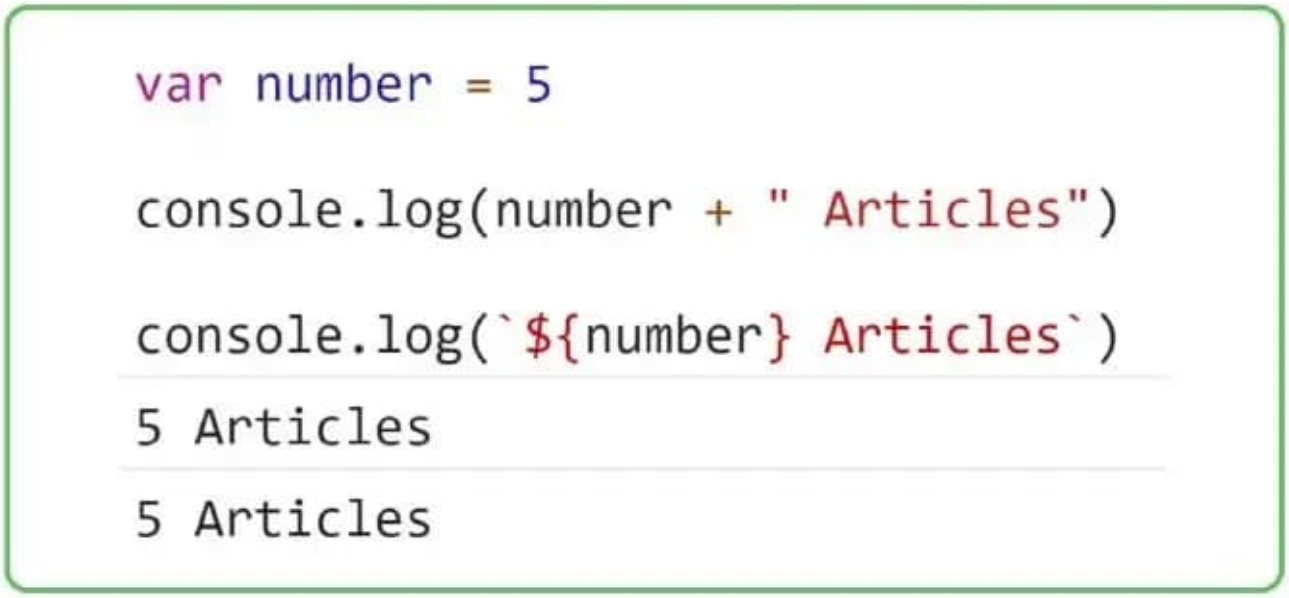
## Are There Any Alternatives?

Although single quotes and double quotes are the most popular, we have a 3rd option called Backticks (``).

> Backticks are an ES6 feature that allows you to create strings in JavaScript.

Although backticks are mostly used for HTML or code embedding purposes, they also act similar to single and double quotes. Besides, using backticks makes it easier for string operations.

### 1. Easy string concatenation.




### 2. You don't need to escape single or double quotes.


**3. Can write multiline without using the new line character.**


With all these advantages, using backticks seems to be a good alternative.

But is there any difference in performance when we use single quotes, double quotes, or backticks? Let's find out.

. . .

## Performance Between Different Quote Types

Since there was not much difference in these 3 methods, I wanted to see whether there is a significant performance gap between using single quotes, double quotes, and backticks.

So, I wrote a small code to create strings with all 3 methods, ran it in the Chrome browser console, and used `console.time` to measure the performance.

```
function testingDoubleQuotes(){
  console.time('doublequotes');
  for (let i = 0; i < 100000; i++) {
   const string1 = "String One";
  }
  console.timeEnd('doublequotes');
}

function testingSingleQuotes(){
  console.time('singlequotes');
  for (let i = 0; i < 100000; i++) {
   const string2 = 'String Two';
  }
  console.timeEnd('singlequotes');
}

function testingbackticks(){
  console.time('backticks');
  for (let i = 0; i < 100000; i++) {
   const string1 = `String Three`;
  }
  console.timeEnd('backticks');
}

testingDoubleQuotes();
testingSingleQuotes();
testingbackticks();
```

Initially, I started with small values, and it didn't show a significant difference. Then I increased the number of times the loops run, and the following results show the time durations with 100000 iterations.

```
doublequotes: 2.77001953125 ms
singlequotes: 2.491943359375 ms
backticks: 1.98095703125 ms
```

Based on the above results, backticks are the best, and double quotes are the worst. But, I don't believe that it will significantly impact the small strings that we use in our projects.

. . .

### Finally, It All Comes Down to Personal Preferences.

As you may have understood now, there is no real difference between using single quotes, double quotes, or backticks. You can choose one or multiple styles based on your preference. However, It is always good to stick to a single format throughout the project to keep it neat and consistent.

Also, you can use code formatters and style guides to take care of the styling for you. They also have a preferred type of quotes.

- Prettier uses "Double Quotes" by default.
- gjslint (Google Closure Linter) favors "Single Quotes".
- eslint uses "Double Quotes" by default.
- Airbnb style guide prefers "Single Quotes".

Using such a tool is the most recommended way to keep your code consistent, and you can configure these tools to your preferred choice as well.

> There are situations where you are forced to use one quoting style. For example, if you are working with JSON, you always need to use double quotes.

Overall, I prefer to use double quotes since I feel it provides more readability for me. Also, I use backticks whenever I need to take advantage of its features like interpolation and multiline strings.

These conventions can be different from person to person. So, don't forget to share your ideas with others in the comments section.

Thank you for Reading!!!



. . .

## Build applications differently

OSS Tools like Bit offer a new paradigm for building modern apps.

Instead of developing monolithic projects, you first build independent components. Then, you compose your components together to build as many applications as you like. This isn't just a faster way to build, it's also much more scalable and helps to standardize development.

It's fun, give it a try →