

```
hello > JS array.js
1  myNumber = 100
2  console.log(myNumber.toString())
3
4  str = '1222'
5  console.log(Math.floor(str))
6
```



Sanchithasr
Posted on 16 дек. 2020 г. • Updated on 19 авг.

7 ways to convert a String to Number in JavaScript

1. Using parseInt()

`parseInt()` parses a string and returns a whole number. Spaces are allowed. Only the first number is returned.

This method has a limitation though. If you parse the decimal number, it will be rounded off to the nearest integer value and that value is converted to *string*. One might need to use `parseFloat()` method for literal conversion.

```
myString = '129'
console.log(parseInt(myString)) // expected result: 129

a = 12.22
console.log(parseInt(a)) // expected result: 12
```

2. Using Number()

`Number()` can be used to convert JavaScript variables to numbers. We can use it to convert the string too number.

If the value cannot be converted to a number, `NaN` is returned.

```
Number("10");           // returns 10
Number(" 10 ");         // returns 10
Number("10.33");        // returns 10.33
```

3. Using Unary Operator (+)

The unary plus operator (`+`) precedes its operand and evaluates to its operand but attempts to convert it into a number, if it isn't already.

```
const x = 25;
const y = -25;
console.log(+x); // expected output: 25
console.log(+y); // expected output: -25
console.log(+''); // expected output: 0
```

4. Using parseFloat()

`parseFloat()` parses a string and returns a number. Spaces are allowed. Only the first number is returned.

```
parseFloat("10");           // returns 10
parseFloat("10.33");        // returns 10.33
parseFloat("10 20 30");     // returns 10
parseFloat("10 years");     // returns 10
parseFloat("years 10");     // returns NaN
```

5. Using Math.floor()

The `Math.floor()` function returns the largest integer less than or equal to a given number. This can be little tricky with decimal numbers since it will return the value of the nearest integer as *Number*.

```
str = '1222'
console.log(Math.floor(str)) // returns 1222

a = 12.22
Math.floor(a) // expected result: 12
```

6. Multiply with number

Multiplying the string value with the `1` which won't change the value and also it will be converted to number by default.

```
str = '2344'
console.log(str * 1) // expected result: 2344
```

7. Double tilde (~~) Operator

We can use the *double tilde* operator to convert the string to number.

```
str = '1234'
console.log(~~str) // expected result: 1234
negStr = '-234'
console.log(~~negStr) // expected result: -234
```

[Here](#) is the comparison of the ways mentioned performance wise. Comment below if you know more methods.

Thank You

Sanchithasr

Follow

Web developer

WORK
Software Engineer

JOINED
26 нояб. 2020 г.

More from Sanchithasr

How to Communicate between Components in Vue.js
#vue #webdev #javascript #intermediate

How to Overlay Icons on Top of Images with CSS
#css #html #webdev #beginners

3 ways to convert HTML text to plain text
#javascript #html #intermediate #webdev

Discussion (5)

Subscribe

- Add to the discussion
- RVxLab

Dec 16 '20 • Edited on Dec 16

...

×

The double tilde method is actually something I never thought about, but is worth explaining.

The double tilde "operator" is not as much as an operator as it's a double **bitwise negation**.

Let's cast `'64'` to `64` using this method, so we do `~~'64'`. First we will evaluate `~'64'`. As bitwise operations work on binary, `'64'` is cast to a number. So `~64`.

`64` in binary is `01000000`. `~` will negate all the bits so it becomes `10111111`, which is `-65` since numbers in JavaScript are signed. Now we negate it again, which becomes `01000000`, which is `64` in decimal.

Correction:

I previously stated that `10111111` is `-63`, which is incorrect. It's actually `-65`. Sorry about that.

5 likes

Reply
- raddevus

Dec 16 '20

...

×

Thanks for explaining this. When I got to that one, I was wondering what it was. Great explanation.

2 likes

Reply
- Sanchithasr

Dec 16 '20 • Edited on Dec 16

...

×

Thank you. This was informative. I didn't pay attention about how it works while writing. Will note it down.

2 likes

Reply
- Jean-Guillaume Buret

Dec 18 '20

...

×

You should always pass the base as second argument to `parseInt`.

1 like

Reply
- Pawan Pawar

Dec 16 '20

...

×

thanks for the info.

2 likes

Reply

Code of Conduct • Report abuse

Read next

- Backend Developer Learning Path 2021**
Coder - Oct 18
- How fast is your code?**
Shuvo - Oct 18
- You are reading environment variables the wrong way in Next.js**
Austin Shelby - Oct 18
- How to create a fancy Github profile with README?**
Creative Tim - Oct 18

