

🔢 JavaScript: Converting Between Numbers and Strings

JavaScript provides several ways to **convert strings to numbers** and **numbers to strings**. Let's explore them with examples! 🚀

1📦 Converting Strings to Numbers

We can convert strings representing numbers to actual numbers using **Number()**, **parseInt()**, and **parseFloat()**.

```
var integerString = "24";  
var num = Number(integerString); // ✔ Converts "24" to 24 (integer)  
console.log(num); // Output: 24
```

```
var floatingNumString = "24.9876";  
var numFloat = Number(floatingNumString); // ✔ Converts "24.9876" to 24.9876 (float)  
console.log(numFloat); // Output: 24.9876
```

🔑 Differences Between Methods:

Method	Converts?	Handles Decimals?	Removes Spaces?	Converts Empty "" to?	Converts null to?
Number()	✔ Integers & Floats	✔ Yes	✔ Yes	✔ 0	✔ 0
parseInt()	✔ Only Integers	✘ No (Truncates)	✘ No	✘ NaN	✘ NaN
parseFloat()	✔ Integers & Floats	✔ Yes	✘ No	✘ NaN	✘ NaN

2📦 Converting Numbers to Strings

When we need to format numbers for **display** (e.g., adding commas), we first convert them to strings using **.toString()**.

```
var numberAsNumber = 1234;  
var numberAsString = numberAsNumber.toString(); // ✔ Converts 1234 to "1234"  
console.log(numberAsString); // Output: "1234"
```

✦ Use Cases for String Conversion:

- Formatting numbers with commas
 - Displaying numbers in a specific way (e.g., `toFixed(2)`)
 - Working with user input in forms
-

3▣ Formatting Numbers with Commas

To **display large numbers** with commas (1,234,567 instead of 1234567), use `.toLocaleString()`:

```
var bigNumber = 1234567;  
console.log(bigNumber.toLocaleString()); // ✔ Output: "1,234,567"
```

You can also specify a **locale**:

```
console.log(bigNumber.toLocaleString("de-DE")); // ✔ Output: "1.234.567" (German format)
```

4▣ Rounding and Formatting Numbers

If you need to **control decimal places**, use `.toFixed()` OR `.toPrecision()`:

```
var num = 24.9876;  
console.log(num.toFixed(2)); // ✔ "24.99" (rounds to 2 decimal places)  
console.log(num.toPrecision(4)); // ✔ "24.99" (keeps 4 significant figures)
```

💡 Best Practices

- **Use `Number()`** for general number conversion (handles both integers and floats).
- **Use `parseInt()`** only when you need **whole numbers**.
- **Use `toString()`** to convert numbers back to strings.
- **Use `.toLocaleString()`** to format numbers **with commas** for readability.
- **Use `.toFixed()`** for **rounding decimals** when displaying prices or percentages.