

# Data types in JavaScript

Sunday, February 16, 2025 10:12 PM

There are two types of data types in JavaScript

1. Primitive data type
2. Non- primitive data type

## 1. Primitive data type

We can classify primitive data type in seven categories

- a. String
- b. Number
- c. Boolean
- d. Undefined
- e. Null
- f. BigInt
- g. Symbol

1. **\*\*String\*\***: Represents textual data. Strings are enclosed in single quotes (') or double quotes (").

```
```JavaScript
let greeting = "Hello, world!";
```
```

2. **\*\*Number\*\***: Represents numeric values, including integers and floating-point numbers.

```
```JavaScript
let age = 25;
let pi = 3.14;
```
```

3. **\*\*Boolean\*\***: Represents logical values. It can be either 'true' or 'false'.

```
```JavaScript
let isJavaScriptFun = true;
```
```

4. **\*\*Undefined\*\***: A variable that has been declared but not assigned a value yet.

```
```JavaScript
let name;
console.log(name); // Output: undefined
```
```

5. **\*\*Null\*\***: Represents the absence of any object value.

```
```JavaScript
let emptyValue = null;
```
```

```
...
```

6. **BigInt**: it Represents big number

```
``// Using the 'n' notation
```

```
const bigintLiteral = 1234567890123456789012345678901234567890n;
```

```
// Using the BigInt() function
```

```
const bigintFunction = BigInt("1234567890123456789012345678901234567890");
```

```
...
```

7. **Symbol**: A unique and immutable primitive value used as the key of an object property.

```
``javascript
```

```
let uniqueId = Symbol("id");
```

```
...
```

## • Convert string to number in JavaScript

In JavaScript, you can convert a string to a number using several methods. Here are a few common ones:

1. **Using `parseInt()` or `parseFloat()`**:

```
``javascript
```

```
let str = "123";
```

```
let num = parseInt(str); // converts to integer
```

```
let floatNum = parseFloat(str); // converts to floating-point number
```

```
...
```

2. **Using the `Number` constructor**:

```
``javascript
```

```
let str = "123";
```

```
let num = Number(str); // converts to number
```

```
...
```

3. **Using the unary plus operator**:

```
``javascript
```

```
let str = "123";
```

```
let num = +str; // converts to number
```

```
...
```

4. **Using `parseInt()` with a radix for hexadecimal or other bases**:

```
``javascript
```

```
let hexStr = "1A";
```

```
let num = parseInt(hexStr, 16); // converts hexadecimal string to number
```

...

5. **\*\*Using `parseFloat()` for strings with decimal points\*\*:**

```
```\javascript
let str = "123.45";
let num = parseFloat(str); // converts to floating-point number
```
```

Choose the method that best fits your needs. If you have any other questions or need further assistance, feel free to ask!

## • Convert number to string

To convert a number into a string in JavaScript, you have several options. Here are a few common methods:

1. Using the `String` function:

```
```\javascript
let num = 123;
let str = String(num); // "123"
```
```

2. Using the `toString` method:

```
```\javascript
let num = 123;
let str = num.toString(); // "123"
```
```

3. Using template literals:

```
```\javascript
let num = 123;
let str = `${num}`; // "123"
```
```

4. Using the `+` operator:

```
```\javascript
let num = 123;
let str = num + ""; // "123"
```
```

Each of these methods will convert a number to a string. Pick the one that best fits your coding style. Let me know if you have any other questions or need further assistance!

## • Convert Null to Number

To convert `null` into a number in JavaScript, you can use the `Number` function. Here's how you can do it:

```
``javascript
let nullValue = null;
let numberValue = Number(nullValue);

console.log(numberValue); // Output will be 0
...`
```

When you convert `null` to a number using the `Number` function, it results in `0`. This is because JavaScript treats `null` as `0` when converting it to a numeric value.

If you have any specific scenarios or additional questions related to this, feel free to ask!

- **convert undefined into number**

In JavaScript, you can convert undefined into a Number using different methods. Here are a few approaches:

- 1. Using Number() Constructor**

The Number() function can be used to convert undefined into a number:

```
let result = Number(undefined);
console.log(result); // NaN
```

This will result in NaN (Not-a-Number), as undefined is not a valid numeric value.

- 2. Using + (Unary Plus) Operator**

You can also use the unary plus (+) operator to convert undefined into a number:

```
let result = +undefined;
console.log(result); // NaN
```

Similar to Number(), it will return NaN.

- 3. Using parseInt() or parseFloat()**

These functions are designed to parse a string and convert it into an integer or a floating-point number. They will also return NaN if they cannot parse a valid number from the input:

```
let result = parseInt(undefined);
console.log(result); // NaN
let resultFloat = parseFloat(undefined);
console.log(resultFloat); // NaN
```

- **Convert undefined into string**

In JavaScript, you can convert undefined into a string using different methods. Here are a few approaches:

- 1. Using String() Constructor**

The String() function can be used to explicitly convert undefined to a string:

```
let result = String(undefined);
console.log(result); // "undefined"
```

This will convert undefined into the string "undefined".

- 2. Using .toString() Method**

If undefined is not explicitly passed as an object, calling `.toString()` on undefined will throw an error. However, if you are dealing with values that might be undefined, you can handle it more safely with a check:

```
let result = (undefined).toString(); // This will throw an error
```

To avoid errors, it's better to use:

```
let result = (undefined !== undefined) ? undefined.toString() : "undefined";  
console.log(result); // "undefined"
```

### **3. Using Template Literals**

You can also use template literals (backticks) to convert undefined to a string:

```
let result = `${undefined}`;  
console.log(result); // "undefined"
```

### **4. Using Concatenation**

By concatenating undefined with an empty string (`""`), it will automatically convert undefined into a string:

```
let result = undefined + "";  
console.log(result); // "undefined"
```