

JS

عنوان مقاله: تبدیل رشته به عدد در جاوا اسکریپت

نویسنده مقاله: تیم فنی نیک آموز

تاریخ انتشار: ۲۶ خرداد ۱۴۰۳

منبع: <https://nikamooz.com/convert-string-to-number-in-javascript/>

تبدیل رشته به عدد در جاوا اسکریپت در دنیای برنامه‌نویسی، مثل تبدیل انواع دیگر داده‌ها به یکدیگر، اجتناب‌ناپذیر است. اغلب داده‌ها از منابع خارجی مانند فرم‌های کاربری، API ها و فایل‌های متنی، به صورت رشته وارد می‌شوند. گاهی این رشته‌ها باید به عدد تبدیل شوند تا محاسبات ریاضی، مقایسه‌ها و مرتب‌سازی قابل انجام باشد. این کار از روش‌های مختلفی انجام می‌شود که در این مطلب به تمام آن‌ها می‌پردازیم. اما پیش از آن، درباره ضرورت این کار، توابع مورد استفاده و همچنین روش‌های مدیریت خطا صحبت خواهیم کرد.

اهمیت تبدیل رشته به عدد در جاوا اسکریپت

اغلب داده‌ها به صورت رشته از منابع خارجی مانند فرم‌های کاربری، API ها و فایل‌های متنی وارد برنامه می‌شوند. در مواقعی که به انجام عملیات محاسباتی، مقایسه عددی، یا مرتب‌سازی نیاز داریم، تبدیل رشته به عدد در جاوا اسکریپت ضروری است. اگر رشته‌ها به اعداد تبدیل نشوند، عملیات محاسباتی به درستی انجام نخواهد شد و ممکن است نتایج نادرست یا غیرمنتظره‌ای حاصل شود.

کاربردهای تبدیل رشته به عدد در جاوا اسکریپت چیست؟

همانطور که در بخش قبل گفتیم، گاهی برای انجام محاسبات ریاضی نیاز به عددهایی دارید که در ورودی‌ها به درستی نمایش داده نشده‌اند. اگر بخواهیم کمی دقیق‌تر بحث را جلو ببریم، باید بگوییم که در موارد زیر می‌توانید از تبدیل رشته به عدد کمک بگیرید:

• ورود داده‌های رشته‌ای از منابع مختلف:

- داده‌ها اغلب به صورت رشته از فرم‌های کاربری، API ها و فایل‌های متنی وارد برنامه می‌شوند.

- مثال: دریافت مقدار "۴۲" از یک فرم HTML

• نیاز به انجام عملیات محاسباتی:

- برای انجام عملیات ریاضی و محاسباتی، داده‌ها باید به اعداد تبدیل شوند.

- مثال: تبدیل رشته "۴۲" به عدد ۴۲ برای انجام جمع یا ضرب.

- مقایسه و مرتب‌سازی صحیح:

- برای مقایسه و مرتب‌سازی عددی، داده‌ها باید به اعداد تبدیل شوند.

- مثال: مقایسه رشته "۱۰" و "۲" به صورت عددی به جای مقایسه لغوی.

- بهبود عملکرد برنامه:

- عملیات عددی، سریع‌تر از عملیات رشته‌ای هستند.

- مثال: اجرای حلقه‌ای که اعداد را جمع می‌کند، از خود رشته‌ها سریع‌تر خواهد بود.

تبدیل رشته به عدد در جاوا اسکریپت تضمین می‌کند که داده‌ها به درستی پردازش و نتایج قابل اطمینانی تولید می‌شود.

تابع های تبدیل رشته به عدد در جاوا اسکریپت

در جاوا اسکریپت هیچ عجیب نیست اگر یک روز ببینید مقدارهای عددی که برای عملیات ریاضی یا مقایسه به آن‌ها نیاز دارید، در رشته‌های عددی گیر افتاده‌اند. ممکن است زمانی که ورودی شما از طریق کاربر یا یک فایل یا حتی API شخص ثالث تأمین شود، با این مشکل روبه‌رو شوید. خوشبختانه جاوا اسکریپت چندین تابع برای تبدیل رشته به عدد ارائه می‌دهد که هرکدام کاربرد خاص خود را دارند:

- parseInt()
- parseFloat()
- Number()
- Unary Plus(+)
- Math.floor()

در این بخش، روش‌های مختلف را با هر یک از روش‌های بالا، به صورت مرحله‌به‌مرحله می‌آموزیم:

۱. تبدیل با استفاده از parseInt()

تابع parseInt() یک رشته را به یک عدد صحیح تبدیل می‌کند. این تابع می‌تواند یک پایه (radix) اختیاری را هم به عنوان آرگومان دوم بپذیرد. آرگومان دوم مشخص می‌کند که عدد باید به کدام سیستم عددی (مانند باینری، اکتال، ددهی و...) تبدیل شود. به مثال زیر توجه کنید:

```
let str = "123";
let num = parseInt(str);
console.log(num); // Output: 123

let strWithRadix = "1010";
let numWithRadix = parseInt(strWithRadix, 2);
console.log(numWithRadix); // Output: 10
```

۲. تبدیل با استفاده از parseFloat()

تابع `parseFloat()` یک رشته را به یک عدد اعشاری تبدیل می‌کند. از این تابع می‌توانید برای تبدیل رشته‌هایی که شامل نقطه اعشاری هستند، استفاده کنید.

```
let str = "123.45";
let num = parseFloat(str);
console.log(num); // Output: 123.45
```

۳. تبدیل با استفاده از Number()

تابع `Number()` یک رشته را به عدد تبدیل می‌کند. این تابع می‌تواند هم اعداد صحیح و هم اعشاری را تبدیل کند. در صورتی که رشته قابل تبدیل نباشد، مقدار `NaN` (Not-a-Number) را برمی‌گرداند.

```
let strInt = "123";
let numInt = Number(strInt);
console.log(numInt); // Output: 123

let strFloat = "123.45";
let numFloat = Number(strFloat);
console.log(numFloat); // Output: 123.45

let strInvalid = "abc";
let numInvalid = Number(strInvalid);
console.log(numInvalid); // Output: NaN
```

۴. تبدیل با استفاده از Unary Plus(+)

عملگر یوناری مثبت (+) نیز می‌تواند برای تبدیل رشته به عدد در جاوا اسکریپت استفاده شود. این روش ساده و کوتاه است اما مانند `Number()` در صورت نادرست بودن رشته، `NaN` برمی‌گرداند.

```
let strInt = "123";
let numInt = +strInt;
console.log(numInt); // Output: 123

let strFloat = "123.45";
let numFloat = +strFloat;
console.log(numFloat); // Output: 123.45

let strInvalid = "abc";
let numInvalid = +strInvalid;
console.log(numInvalid); // Output: NaN
```

۵. تبدیل با استفاده از Math.floor()

تبدیل رشته به عدد در جاوا اسکریپت با استفاده از Math.floor() معمولاً زمانی انجام می‌شود که نیاز به تبدیل رشته به یک عدد صحیح و سپس گردکردن آن به پایین‌ترین عدد صحیح داریم. Math.floor() یک تابع ریاضی است که عدد اعشاری را به پایین‌ترین عدد صحیح نزدیک به آن عدد گرد می‌کند. برای استفاده از Math.floor() ابتدا باید رشته را به یک عدد اعشاری تبدیل و سپس آن را گرد کنیم. این تبدیل معمولاً با استفاده از parseFloat() انجام می‌شود. در نهایت، نتیجه را با Math.floor() گرد می‌کنیم.

مثال

فرض کنید رشته‌ای داریم که نمایانگر یک عدد اعشاری است و می‌خواهیم آن را به پایین‌ترین عدد صحیح نزدیک گرد کنیم. در این مثال برای تبدیل رشته به عدد به روش زیر عمل می‌کنیم:

```
let str = "42.78";
let num = parseFloat(str); // تبدیل رشته به عدد اعشاری
let flooredNum = Math.floor(num); // گرد کردن به پایین‌ترین عدد صحیح نزدیک
console.log(flooredNum); // نتیجه: ۴۲
```

روش مدیریت خطاها در تبدیل رشته به عدد در جاوا اسکریپت

برای مدیریت خطا لازم است مقادیر null ، NaN (Not-a-Number) و متغیرهای تعریف نشده در جاوا اسکریپت برای نوشتن کدی پایدار و قابل اطمینان ضروری را کنترل کنید. برخی از روش‌هایی که می‌توانید به کار ببرید، عبارتند از:

۱. بررسی null یا undefined

برای بررسی اینکه آیا یک متغیر null یا undefined است، می‌توانید از الگوی مختصر زیر استفاده کنید:

```
if (some_variable == null) {
  // Handle null or undefined value
}
```

این روش، هر دو مقدار null و undefined را پوشش می‌دهد. توجه داشته باشید که در این روش، باید متغیر تعریف شده باشد؛ در غیر این صورت، یک ReferenceError رخ خواهد داد.

۲. عملگر Nullish Coalescing

عملگر Nullish Coalescing (??) یک روش مدرن برای مدیریت مقادیر null یا undefined است:

```
const result = some_variable ?? defaultValue;
```

اگر some_variable null یا undefined باشد، result به defaultValue برگردانده می‌شود.

۳. بررسی NaN

برای مدیریت NaN (که نمایانگر یک عدد نامعتبر است)، به روش زیر از `isNaN()` استفاده کنید:

```
if (isNaN(some_variable)) {  
    // Handle NaN value  
}
```

۴. متغیرهای تعریف نشده

هنگام کار با متغیرهای سراسری تعریف نشده، استفاده از `typeof`، یکی از امن‌ترین روش‌ها است:

```
if (typeof some_global_variable === 'undefined') {  
    // Handle undeclared variable  
}
```

با وجود این، برای بهبود سازماندهی کد، توصیه می‌کنیم استفاده از متغیرهای تعریف نشده را به حداقل برسانید.

جمع بندی: تبدیل رشته به عدد در جاوا اسکریپت

تبدیل رشته به عدد در جاوا اسکریپت یک مهارت کلیدی برای هر برنامه‌نویس است و حالا شما به خوبی با آن آشنا هستید. شما می‌دانید که چطور با استفاده از توابع مختلف در جاوا اسکریپت، رشته را به عدد تبدیل کنید. در صورتیکه به ارور برخورد کنید نیز مشکلی نخواهید داشت؛ چراکه روش‌های مدیریت خطا را می‌شناسید. یادتان باشد که انتخاب روش و تابع مناسب، تأثیری مستقیم در پاسخی که دریافت می‌کنید، خواهد گذاشت. اگر هنوز درباره مطالبی که عنوان کردیم، سؤال دارید؛ از بخش نظرات زیر همین پست از ما بپرسید.