

한국어 ▼

# Intl.NumberFormat

**Intl.NumberFormat**은 언어에 맞는 숫자 서식을 지원하는 객체의 생성자입니다.

## JavaScript Demo: Intl.NumberFormat

```
1 const number = 123456.789;
2
3 console.log(new Intl.NumberFormat('de-DE', { style: 'currency'
4 // expected output: "123.456,79 €"
5
6 // the Japanese yen doesn't use a minor unit
7 console.log(new Intl.NumberFormat('ja-JP', { style: 'currency'
8 // expected output: "¥123,457"
9
10 // limit to three significant digits
11 console.log(new Intl.NumberFormat('en-IN', { maximumSignifi
12 // expected output: "1,23,000"
13
```

Run ›

Reset

## 구문

```
new Intl.NumberFormat([Locales[, options]])
Intl.NumberFormat.call(this[, locales[, options]])
```

## 매개 변수

### locales | Optional

BCP 47 언어 태그를 포함하는 문자열이나 문자열의 배열. 로케일 매개변수의 일반적인 형식 및 해석은 Intl 문서를 참고하세요. 다음의 Unicode 확장 키를 사용할 수 있습니다.

#### nu

사용할 기수법. "arab", "arabext", "bali", "beng", "deva", "fullwide", "gujr", "guru", "hanidec", "khmr", "knda", "lao", "latn", "limb", "mlym", "mong", "mymr", "orya", "tamldec", "telu", "thai", "tib" 등을 사용할 수 있습니다.

### options | Optional

다음 속성을 포함하는 객체.

#### localeMatcher

사용할 로케일 매칭 알고리즘. 가능한 값은 "lookup"과 "best fit"이고, 기본값은 "best fit"입니다. 자세한 정보는 Intl 문서를 참고하세요.

#### style

사용할 서식 스타일. 가능한 값은 평문 숫자 "decimal", 통화 서식 "currency", 백분율 서식 "percent"입니다. 기본값은 "decimal"입니다.

#### currency

통화 서식에 사용할 통화입니다. 가능한 값은 ISO 4217 통화 코드로 대한민국 원화 "KRW", 미국 달러화 "USD", 유럽 유로화 "EUR" 등을 포함합니다. 현재 통화 및 펀드 코드 목록을 참고하세요. 기본값은 없습니다. style 값을 "currency"로 지정했다면, currency 값도 반드시 지정해야 합니다.

#### currencyDisplay

통화 서식에서 통화를 표시하는 방법. 가능한 값은 €와 같이 현지 통화 기호를 사용하는 "symbol", ISO 통화 코드를 사용하는 "code", "dollar" 등 현지 통화 이름을 사용하는 "name"이 있습니다. 기본 값은 "symbol"입니다.

#### useGrouping

천 단위 구분자 혹은 천/라크/크로르 단위 구분자의 삽입 여부. 가능한 값은 true와 false입니다. 기본 값은 true입니다.

아래 속성은 두 가지 그룹으로 나뉩니다. minimumIntegerDigits, minimumFractionDigits, maximumFractionDigits가 첫 번째 그룹이고, minimumSignificantDigits와 maximumSignificantDigits가 두 번째 그룹으로, 두 번째 그룹 중 하나라도 지정하면 첫 번째 그룹은 모두 무시합니다.

**minimumIntegerDigits**

정수부의 최소 자릿수. 가능한 값은 1부터 21까지의 수로, 기본값은 1입니다.

**minimumFractionDigits**

소수부의 최소 자릿수. 가능한 값은 0부터 20까지의 수로, 평문 숫자와 백분율 서식의 기본값은 0입니다. 통화 서식의 기본값은 ISO 4217 통화 코드 목록이 제공하는 보조 단위의 자릿수에 따라 다릅니다. (목록에 정보가 없을 경우 2)

**maximumFractionDigits**

소수부의 최대 자릿수. 가능한 값은 0부터 20까지의 수로, 평문 숫자 서식의 기본값은 `minimumFractionDigits`와 3 중 더 큰 값입니다. 통화 서식의 기본값은 `minimumFractionDigits`와, ISO 4217 통화 코드 목록이 제공하는 보조 단위의 자릿수 (목록에 정보가 없을 경우 2) 중 더 큰 값입니다. 백분율 서식의 기본값은 `minimumFractionDigits`와 0 중 더 큰 값입니다.

**minimumSignificantDigits**

유효숫자의 최소 수. 가능한 값은 1부터 21까지의 수로, 기본값은 1입니다.

**maximumSignificantDigits**

유효숫자의 최대 수. 가능한 값은 1부터 21까지의 수로, 기본값은 21입니다.

---

## 설명

### 속성

**Intl.NumberFormat.prototype**

모든 인스턴스에 속성을 추가할 수 있습니다.

### 메서드

**Intl.NumberFormat.supportedLocalesOf()**

주어진 로케일 목록 중, 런타임이 지원하는 항목을 배열로 반환합니다.

---

## NumberFormat 인스턴스

## 속성

NumberFormat 인스턴스는 프로토타입의 다음 속성을 상속합니다.

### Intl.NumberFormat.prototype.constructor

Intl.NumberFormat 에 대한 참조입니다.

## 메서드

NumberFormat 인스턴스는 프로토타입의 다음 메서드를 상속합니다.

### Intl.NumberFormat.prototype.format()

주어진 숫자에 NumberFormat 객체의 로케일과 서식 설정을 적용해 반환합니다.

### Intl.NumberFormat.prototype.formatToParts()

서식을 적용한 숫자의 각 부분을 객체로 표현해서 Array 로 반환합니다. 반환 값은 로케일별 커스텀 서식에 사용할 수 있습니다.

### Intl.NumberFormat.prototype.resolvedOptions()

객체 인스턴스 생성 때 주어진 로케일과 콜레이션 설정이 어떻게 반영되었나 나타내는 새로운 객체를 반환합니다.

---

## 예제

### 기본적인 사용 방법

로케일을 지정하지 않고 사용하면 기본 로케일 및 기본 옵션 서식을 적용한 문자열을 반환합니다.

```
1 | var number = 3500;
2 |
3 | console.log(new Intl.NumberFormat().format(number));
4 | // → 한국 로케일의 경우 '3,500' 표시
```

## locales 사용하기

다음 예제는 지역화된 숫자 서식의 예시를 보입니다. 어플리케이션의 사용자 인터페이스 언어에 맞는 서식을 적용하려면 `locales` 매개변수로 적절한 언어(와, 필요한 경우 대체 언어)를 제공하는걸 잊지 마세요.

```

1  var number = 123456.789;
2
3  // 독일은 소수점 구분자로 쉼표를 사용하고 천 단위 구분자로 마침표를 사용
4  console.log(new Intl.NumberFormat('de-DE').format(number));
5  // → 123.456,789
6
7  // 대부분의 아랍어 사용 국가에서는 실제 아라비아 숫자를 사용
8  console.log(new Intl.NumberFormat('ar-EG').format(number));
9  // → ١٢٣٤٥٦,٧٨٩
10
11 // 인도는 천, 라크(십만), 크로르(천만) 단위에 구분자 사용
12 console.log(new Intl.NumberFormat('en-IN').format(number));
13 // → 1,23,456.789
14
15 // nu 확장 키로 기수법 지정 (아래 예시는 중국식 숫자 표기)
16 console.log(new Intl.NumberFormat('zh-Hans-CN-u-nu-hanidec').format(number));
17 // → 一二三,四五六.七八九
18
19 // 발리어와 같이 지원되지 않을 수도 있는 언어를 지정할 때는
20 // 다음과 같이 대체 언어를 지정할 수 있음. 아래의 경우 대체 언어는 인도어
21 console.log(new Intl.NumberFormat(['ban', 'id']).format(number));
22 // → 123.456,789

```

## options 사용

`options` 매개변수를 지정해 결과를 원하는 형태로 바꿀 수 있습니다.

```

1  var number = 123456.789;
2
3  // 통화 서식
4  console.log(new Intl.NumberFormat('de-DE', { style: 'currency', currency: 'EUR' }).format(number));
5  // → 123.456,79 €
6

```

```
7 // 한국 원화는 보조 통화 단위를 사용하지 않음
8 console.log(new Intl.NumberFormat('ko-KR', { style: 'currency', curren
9 // → ₩123,457
10
11 // 유효숫자를 세 개로 제한
12 console.log(new Intl.NumberFormat('en-IN', { maximumSignificantDigits:
13 // → 1,23,000
```

## 명세

명세	상태	Comment
ECMAScript Internationalization API 1.0 (ECMA-402) The definition of 'Intl.NumberFormat' in that specification.	ST Standard	초기 정의.
ECMAScript Internationalization API 2.0 (ECMA-402) The definition of 'Intl.NumberFormat' in that specification.	ST Standard	
ECMAScript Internationalization API (ECMA-402) The definition of 'Intl.NumberFormat' in that specification.	LS Living Standard	

## 브라우저 호환성

[Update compatibility data on GitHub](#)

NumberFormat	
Chrome	24
Edge	12
Firefox	29
IE	11
Opera	15
Safari	10

WebView Android	4.4
Chrome Android	25
Firefox Android	56
Opera Android	14
Safari iOS	10
Samsung Internet Android	1.5
nodejs	0.12

NumberFormat() constructor

Chrome	24
Edge	12
Firefox	29
IE	11
Opera	15
Safari	10
WebView Android	4.4
Chrome Android	25
Firefox Android	56
Opera Android	14
Safari iOS	10
Samsung Internet Android	1.5
nodejs	13.0.0

format

Chrome	24
Edge	12
Firefox	29
IE	11
Opera	15
Safari	10

WebView Android	4.4
Chrome Android	25
Firefox Android	56
Opera Android	14
Safari iOS	10
Samsung Internet Android	1.5
nodejs	0.12

## formatToParts

Chrome	64
Edge	12
Firefox	58
IE	No
Opera	51
Safari	13
WebView Android	64
Chrome Android	64
Firefox Android	58
Opera Android	47
Safari iOS	13
Samsung Internet Android	9.0
nodejs	10.0.0

## resolvedOptions

Chrome	24
Edge	12
Firefox	29
IE	11
Opera	15
Safari	10



WebView Android	4.4
Chrome Android	25
Firefox Android	56
Opera Android	14
Safari iOS	10
Samsung Internet Android	1.5
nodejs	0.12

## supportedLocalesOf

Chrome	24
Edge	12
Firefox	29
IE	11
Opera	15
Safari	10
WebView Android	4.4
Chrome Android	25
Firefox Android	56
Opera Android	14
Safari iOS	10
Samsung Internet Android	1.5
nodejs	13.0.0

What are we missing?



Full support



No support

Experimental. Expect behavior to change in the future.

See implementation notes.

## 같이 보기

- 소개글: The ECMAScript Internationalization API
- 생성자
  - Intl.Collator
  - Intl.DateTimeFormat
  - Intl.ListFormat
  - Intl.NumberFormat
  - Intl.PluralRules
  - Intl.RelativeTimeFormat
- 메서드
  - String.prototype.localeCompare()
  - Number.prototype.toLocaleString()
  - Date.prototype.toLocaleString()
  - Date.prototype.toLocaleDateString()
  - Date.prototype.toLocaleTimeString()

---

마지막 변경일: 2020년 4월 21일, by MDN contributors

## 관련 주제

### Standard built-in objects

Intl

### Methods

Intl.getCanonicalLocales() [\[Translate\]](#)

### Related pages:

Collator

DateTimeFormat

ListFormat

NumberFormat

## NumberFormat

### PluralRules

### RelativeTimeFormat

### Inheritance:

### Object

### Properties

Object.prototype.\_\_proto\_\_ [Translate]

Object.prototype.constructor

### Methods

Object.prototype.\_\_defineGetter\_\_()

Object.prototype.\_\_defineSetter\_\_() [Translate]

Object.prototype.\_\_lookupGetter\_\_() [Translate]

Object.prototype.\_\_lookupSetter\_\_() [Translate]

Object.prototype.hasOwnProperty()

Object.prototype.isPrototypeOf()

Object.prototype.propertyIsEnumerable()

Object.prototype.toLocaleString()

Object.prototype.toSource() [Translate]

Object.prototype.toString()

Object.prototype.valueOf()

Object.setPrototypeOf()

---

×

# 웹 개발 배우기

받은 편지함으로 바로 배달되는 MDN의 최신 뉴스와 좋은 글을 받아보세요.

뉴스레터는 아직 영어로만 제공됩니다.

지금 가입하기