



Chap28 Number

▼ Number 생성자 함수

표준 빌트인 객체로서 생성자 함수 객체는 new 연산자와 함께 Number 인스턴스를 생성할 수 있다.

```
const numObj = new Number();  
console.log(numObj); // Number {[[PrimitiveValue]]: 0}
```

- 인수를 전달하지 않고 호출하면 `[[NumberData]]` 내부 슬롯에 0을 할당한 Number 래퍼 객체를 생성.
- 인수로 숫자를 전달하면 `[[NumberData]]` 내부 슬롯에 인수를 할당한 Number 래퍼 객체를 생성
- 인수로 숫자가 아닌 값을 전달하면 숫자로 강제 변환 후 전달한다.

▼ Number 프로퍼티

▼ Number.ERPSILON

- 1과 1보다 큰 숫자 중에서 가장 작은 숫자와의 차이
- `Number.EPSILON` : `2.2204460492503130808472633361816E-16`.
- 부동소수점 연산할 때 유용

```
//원래는 이런식으로 무한소수가 되어 비교가 불가능  
0.1 + 0.2; // -> 0.30000000000000004  
0.1 + 0.2 === 0.3; // -> false
```

```
//이런식으로 대체가능  
function isEqual(a, b){  
  // a와 b를 뺀 값의 절대값이 Number.EPSILON보다 작으면 같은  
  return Math.abs(a - b) < Number.EPSILON;  
}
```

```
isEqual(0.1 + 0.2, 0.3); // -> true
```

▼ Number.MAX_VALUE

- 자바스크립트에서 표현할 수 있는 가장 큰 양수 값
- Number.MAX_VALUE보다 큰 숫자는 Infinity이다.

▼ Number.MIN_VALUE

- 자바스크립트에서 표현할 수 있는 가장 작은 양수 값
- Number.MIN_VALUE보다 작은 숫자는 0이다.

▼ Number.MAX_SAFE_INTEGER

- 자바스크립트에서 안전하게 표현할 수 있는 가장 큰 정수값

▼ Number.MIN_SAFE_INTEGER

- 자바스크립트에서 안전하게 표현할 수 있는 가장 작은 정수값

▼ Number.POSITIVE_INFINITY

양의 무한대를 나타내는 숫자값 Infinity와 같다.

▼ Number.NEGATIVE_INFINITY

음의 무한대를 나타내는 숫자값 -Infinity와 같다.

▼ Number.NaN

숫자가 아님을 나타내는 숫자값. window.NaN과 같다.

▼ Number 메서드

▼ Number.isFinite

- 인수로 전달된 숫자값이 Infinity 또는 -Infinity가 아닌지 검사하여 그 결과를 불리언 값으로 반환
- 만약 인수가 NaN이면 언제나 false
- 인수가 숫자가 아닌 인수가 주어졌을 때 반환값은 언제나 false. 즉 암묵적 타입 변환 하지 않는다.

▼ Number.isInteger

- 인수로 전달된 숫자값이 정수인지 검사하여 그 결과를 불리언으로 반환

- 검사하기 전에 인수를 숫자로 암묵적 타입변환하지 않는다.

▼ Number.isNaN

- 빌트인 isNaN과 차이가 있다. 빌트인 전역 함수 isNaN은 전달받은 인수를 숫자로 암묵적타입 변환하지만 Number.isNaN메서드는 암묵적 타입 변환하지 않는다.
- 숫자가 아닌 인수가 주어졌을 때 언제나 false 반환

▼ Number.isSafeInteger

- 인수로 전달된 숫자값이 안전한 정수인지 검사하여 그 결과를 불리언 값으로 반환
- 안전한 정수값 : $-(2^{53} - 1)$ 부터 $2^{53} - 1$ 사이의 모든 정수값
- 암묵적 타입 변환 하지 않음

▼ Number.prototype.toExponential

- 숫자를 지수 표기법으로 변환하여 문자열로 반환
- 지수 표기법 : 매우 크거나 작은 숫자를 표기할 때 주로 사용하며 e앞에 있는 숫자에 10의 n승을 곱하는 형식으로 수를 나타내는 방식
- 인수로 소수점 이하로 표현할 자릿수를 전달할 수 있다.
- Number프로토타입 메서드를 사용할 경우 에러가 발생

```
77.toExponential();//SyntaxError
```

▼ Number.prototype.toFixed

- 숫자를 반올림하여 문자열로 반환
- 소수점 이하 자릿수를 나타내는 0~20사이의 정수값을 인수로 전달할 수 있다.
- 인수를 생략하면 기본값은 0

▼ Number.prototype.toPrecision

- 인수로 전달받은 전체 자릿수까지 유효하도록 나머지 자릿수를 반올림하여 문자열로 반환
- 인수로 전달받는 전체 자릿수를 표현할 수 없는 경우 지수 표기법으로 결과를 반환한다.

▼ Number.prototype.toString

- 숫자를 문자열로 변환하여 반환.
- 진법을 나타내는 2~36사이의 정수값을 인수로 전달할 수 있다.
- 인수를 생략하면 10진법이 기본값으로 지정된다.