

28장 Number

28.1 Number 생성자 함수

- new Number()
- Number("10") => 숫자 10 반환
- Number(true) => 1 반환

28.2 Number 프로퍼티

- 1. Number.EPSILON — 부동 소수점 산술 연산
정확한 결과를 위해 사용 — Ex) $0.1 + 0.2 = 0.3000000000000004 \neq 0.3$
두 소수를 빼서 그 차이가
EPSILON보다 작으면 같은 수로
인정
- 2. Number.MAX_VALUE — JS에서 가장 큰 양수 반환 — Infinity보다는 작음
- 3. Number.MIN_VALUE — 가장 작은 양수 반환
- 4. Number.MAX_SAFE_INTEGER — Js에서 안전하게 표현할 수
있는 가장 큰 정수값 — 9007199254740991
- 5. Number.MIN_SAFE_INTEGER — 가장 작은 정수값 — -9007199254740991
- 6. Number.POSITIVE_INFINITY — Infinity와 같다.
- 7. Number.NEGATIVE_INFINITY — -Infinity와 같다.
- 8. Number.NaN — NaN 반환 — window.NaN와 같다.

28.3 Number 메서드

- 1. Number.isFinite — 인수가
Infinity인지
아닌지 확인하여 불리언 값 출력
- 2. Number.isInteger — 인수가 정수이면 true 반환
- 3. Number.isNaN — 인수가 NaN인지 확인
- 4. Number.isSafeInteger — 인수가 안전한 정수인지 확인 (안전한 정수는
-(253-1)과 253-1 사이의 정수값이다.)
- 5. Number.prototype.toExponential — 입력된 숫자를 지수 표기법으로
변환하여 문자열로 반환한다. — 인수의 숫자만큼 소수점
자리를 출력해준다.
 - (77.1234).toExponential(); => 7.71234e+1
 - (77.1234).toExponential(4); => 7.7123e+1
 - (77.1234).toExponential(2); => 7.71e+1
- 6. Number.prototype.toFixed — 숫자를 반올림하여 문자열로 반환한다. — 인수의 숫자만큼
소수점이 남게 된다.
 - (12345.6789).toFixed(); => "12346"
 - (12345.6789).toFixed(1); => "12345.7"
 - (12345.6789).toFixed(2); => "12345.68"
- 7. Number.prototype.toPrecision — 인수로 전달받은 전체 자리수까지 유효하도록 나머지
자릿수를 반올림하여 문자열로 반환
 - (12345.6789).toPrecision(); => "12345.6789"
 - (12345.6789).toPrecision(1); => "1e+4"
 - (12345.6789).toPrecision(2); => "1.2e+4"
 - (12345.6789).toPrecision(6); => "12345.7"
- 8. Number.prototype.toString — 숫자를 문자열로 변환하여 반환한다. 진법을 나타내는
2~36사이의 정수값을 인수로 전달한다. 인수 생략시
기본값 10진법이 지정된다.
 - (10).toString(); => "10"
 - (16).toString(2); => "10000"
 - (16).toString(8); => "20"
 - (16).toString(16); => "10"