삼육대학교 SW융합교육원 김진호

자바스크립트 기초교육

<class>
</body>
<head>
<title>

11-2. Number

11-2-1. Number

11-2-2. Number property

11-2-3. Number method

11-2. Number

> 표준 빌트인 객체인 Number는 원시 타입인 숫자를 다룰 때 유용한 프로퍼티와 메서드를 제공한다.

11-2-1. Number

```
/* Number 생성자 함수 */
const obj = new Number();          // Number 인스턴스 생성
console.log(obj);                  // 인수 전달하지 않을 경우 0을 할당

const obj2 = new Number(1);
console.log(obj2);                 // 인수로 전달 받은 숫자 할당

const obj3 = new Number('1');
console.log(obj3);                 // 인수로 전달 받은 문자 숫자로 형변환

const obj4 = new Number('number');
console.log(obj4);                 // 숫자 형변환 불가 시 NaN
```

11-2. Number

11-2-2. Number property

Number.MAX_VALUE

- 자바스크립트에서 표현할 수 있는 가장 큰 양수 값

```
console.log(Number.MAX_VALUE);              // 1.7976931348623157e+308
console.log(Infinity > Number.MAX_VALUE);   // 무한대가 크다
```

Number.MIN_VALUE

- 자바스크립트에서 표현할 수 있는 가장 작은 양수 값

```
console.log(Number.MIN_VALUE);              // 5e-324
console.log(Number.MIN_VALUE > 0);          // 0보다 크다
```

11-2. Number

11-2-2. Number property

Number.MAX_SAFE_INTEGER

- 자바스크립트에서 안전하게 표현할 수 있는 가장 큰 정수 값

Number.MIN_SAFE_INTEGER

- 자바스크립트에서 안전하게 표현할 수 있는 가장 작은 정수 값

```
console.log(Number.MAX_SAFE_INTEGER);    // 9007199254740991
console.log(Number.MIN_SAFE_INTEGER);    // -9007199254740991
```

11-2. Number

11-2-2. Number property

Number.POSITIVE_INFINITY

- 양의 무한대를 나타내는 숫자값 Infinity와 같다

Number.NEGATIVE_INFINITY

- 음의 무한대를 나타내는 숫자값 -Infinity와 같다

```
console.log(Number.POSITIVE_INFINITY);    // Infinity
console.log(Number.NEGATIVE_INFINITY);    // -Infinity
```

11-2. Number

11-2-2. Number property

Number.NaN

- 숫자가 아님을 나타내는 숫자값

```
console.log(Number.NaN);        // NaN
```

11-2. Number

11-2-2. Number property

Number.EPSILON

- 부동 소수점으로 인해 발생하는 오차를 해결하기 위해 사용한다.

```
console.log(Number.EPSILON);     // 1과 1보다 큰 숫자 중에서 가장 작은 숫자와의 차이와 같다
console.log(0.1 + 0.2);          // 부동소수점 표현은 2진법으로 변환했을 때 무한소수가 되어 미세한 오차가 발생할 수 밖에 없다
미세한 오차가 발생할 수 밖에 없다
console.log(0.1 + 0.2 === 0.3);              // false
console.log(isEqual(0.1 + 0.2, 0.3));        // true

function isEqual(a, b) {
    // a - b의 절대값이 Number.EPSILON 보다 작으면 같은 수로 인정한다
    return Math.abs(a - b) < Number.EPSILON;
}
```

11-2. Number

11-2-3. Number method

Number.isFinite

- 인수로 전달 된 숫자값이 정상적인 유한수인지 검사하여 결과를 불리언으로 반환

```
console.log(Number.isFinite(10));           // true
console.log(Number.isFinite(-10));          // true
console.log(Number.isFinite(Infinity));     // false
console.log(Number.isFinite(-Infinity));    // false
console.log(Number.isFinite(NaN));          // false

console.log(Number.isFinite(null));

// 빌트인 전역함수 isFinite는 암묵적 타입변환을 한다.
console.log(isFinite(null));
```

11-2. Number

11-2-3. Number method

Number.isInteger

- 인수로 전달 된 숫자값이 정수인지 검사하여 결과를 불리언으로 반환

```
console.log(Number.isInteger(10));          // true
console.log(Number.isInteger(-10));         // true
console.log(Number.isInteger(10.10));       // false
console.log(Number.isInteger(-10.10));      // false
console.log(Number.isInteger('10'));        // false
console.log(Number.isInteger(false));       // false
console.log(Number.isInteger(Infinity));    // false
console.log(Number.isInteger(-Infinity));   // false;
```

11-2. Number

11-2-3. Number method

Number.isNaN

- 인수로 전달 된 숫자값이 NaN인지 검사하여 결과를 불리언으로 반환

```
console.log(Number.isNaN(NaN));             // true
console.log(Number.isNaN(undefined));       // false

// 빌트인 전역함수 isNaN은 암묵적 타입변환을 한다
console.log(isNaN(undefined));
```

11-2. Number

11-2-3. Number method

Number.isSafeInteger

- 인수로 전달 된 숫자값이 안전한 정수인지 검사하여 결과를 불리언으로 반환

```
console.log(Number.isSafeInteger(10));          // true
console.log(Number.isSafeInteger(10000000000000000000));      // false
console.log(Number.isSafeInteger(10.10));       // false
console.log(Number.isSafeInteger('10'));        // false
console.log(Number.isSafeInteger(false));       // false
console.log(Number.isSafeInteger(Infinity));    // false
```

11-2. Number

11-2-3. Number method

Number.prototype.toExponential

- 숫자를 지수 표기법으로 변환하여 문자열로 반환

```
// e 앞에 있는 숫자에 10의 n승을 곱하는 형식으로 수를 나타낸다
console.log((1.23456).toExponential());          // 1.23456e+0
// 소수점 이하로 표현할 자리수 전달한다.
console.log((1.23456).toExponential(3));         // 1.235e+0
console.log((1.23456).toExponential(1));         // 1.2e+0
```

11-2. Number

11-2-3. Number method

Number.prototype.toFixed

- 숫자를 반올림하여 문자열로 반환

```
// 반올림 하는 소수점 이하 자리수를 나타내는 0~20 사이의 정수값을 인수로 전달할 수 있다
console.log((1.23456).toFixed());          // 1
console.log((1.23456).toFixed(3));         // 1.235
console.log((1.23456).toFixed(1));         // 1.2
```

11-2. Number

11-2-3. Number method

Number.prototype.toPrecision

- 인수로 전달 받은 전체 자릿수까지 유효하도록 나머지 자릿수를 반올림하여 문자열로 반환

```
// 0~21사이의 정수 값을 인수로 전달할 수 있으며 생략하면 기본값 0이 지정된다
console.log((123.456).toPrecision());           // 123.456
console.log((123.456).toPrecision(5));          // 123.46
console.log((123.456).toPrecision(3));          // 123
console.log((123.456).toPrecision(1));          // 1e+2
```

11-2. Number

11-2-3. Number method

Number.prototype.toString

- 숫자를 문자열로 변환하여 반환

```
// 진법을 나타내는 2~36 사이의 정수값을 인수로 전달할 수 있다
console.log((100).toString());                  // 100
console.log((100).toString(2));                 // 1100100
console.log((100).toString(8));                 // 144
console.log((100).toString(16));                // 64
```