

# JS-중급 5

☰ 태그	
📅 날짜	@2023년 6월 1일

## toString()

10진수 → 2진수/16진수

```
let num = 10;

num.toString(); // "10"
num.toString(2); // "1010"

let num2 = 255;

num2.toString(16); // "ff" 색상에서 사용
```

**Math.PI**  
JS에는 수학과 관련된 프로퍼티와 메소드를 가지고 있는 Math 라는 내장 객체가 있다.

**Math.PI**  
3.14~~~~

**Math.ceil()** : 올림

```
let num1 = 5.1;
let num2 = 5.7;

Math.ceil(num1); //6
Math.ceil(num2); //6
```

-----

`Math.floor()` : 내림

```
let num1 = 5.1;
let num2 = 5.7;
```

```
Math.floor(num1) // 5
Math.floor(num2) // 5
```

-----

`Math.round()` : 반올림

```
let num1 = 5.1;
let num2 = 5.7;
```

```
Math.round(num1) // 5
Math.round(num2) // 6
```

소수점 자리수

```
let userRate = 30.1234;
```

요구 사항 : 소수점 둘째자리까지 표현(셋째 자리에서 반올림)

```
Math.round(userRate * 100)/100 // 30.12
```

-----

혹은 `toFixed()`

숫자를 인수로 받아서 그 숫자만큼 소숫점 이하 개수의 반응

```
userRate.toFixed(2);
> "30.12"
```

만약 소수점의 개수보다 크거나 작다면

```
userRate.toFixed(0);
> "30" 정수로 반환
```

```
userRate.toFixed(6);
> "30.123400" 0으로 채워짐
```

통계나 지표 작업할 때 많이 쓰인다.

주의사항

toFixed는 문자열을 반환한다는 점이다.  
반환 후 Number를 이용해 숫자로 변환

isNaN()

```
let x = Number('x'); //NaN
```

```
x == NaN // false
```

```
x === NaN // false
```

```
NaN == NaN // false
```

```
isNaN(x) // true
```

```
isNaN(3) // false
```

NaN은 자신과도 똑같이 **않**다고 한다.

isNaN만이 NaN이 아닌지 판단할 수 있다.

-----

parseInt() : 문자열을 숫자로 변환

```
let margin = '10px';
```

```
parseInt(margin); // 10
```

```
Number(margin); // NaN
```

Number와 다른점은 문자가 혼용되어 있어도 동작을 한다는점

parseInt는 읽을 수 있는 부분까지는 읽고 문자를 만나면

숫자를 반환한다.

```
let redColor = 'f3';
```

```
parseInt(redColor); // NaN
```

이렇게 숫자로 시작하지 않으면 NaN을 반환한다.

두번째 인수를 받아서 진수를 지정할 수 있다.

```
let redColor = 'f3';
```

```
parseInt(redColor, 16); // 243
```

```
parseInt('11', 2) // 3
```

parseFloat() : parseInt와 동일하게 동작하지만 부동 소수점을 반환한다.

parseInt는 소수점 이하는 무시하고 정수만 반환한다.

```
let padding = '18.5%';
```

```
parseInt(padding); // 18
```

```
parseFloat(padding); // 18.5
```

---

Math.random() : 0 ~ 1 사이 무작위 숫자 생성

```
Math.random() // 0.1654832
```

```
Math.random() // 0.84654654321
```

```
Math.random() // 0.4875456522
```

1 ~ 100까지 수를 뽑고 싶다면

```
Math.floor(Math.random()*100) + 1
```

+ 1을 하는 이유는 랜덤에서 0.0어쩌구가 나올수 있기 때문이다

---

Math.max() / Math.min()

```
Math.max(1,6,7,2,8); // 8
```

```
Math.min(1,6,-7,-3,10,5.51); // -7
```

---

Math.abs() : 절대값

```
Math.abs(-7) // 7
```

---

Math.pow(n,m) : n의 m승 값

```
Math.pow(2, 10); // 1024 2의 10승
```

---

Math.sqrt() : 제곱근

```
Math.sqrt(16); // 4
```