

Math Extensions

in-built mathematical functions in ES6

The `Number` and the `Math` objects have been extended with many useful methods. There are many new mathematical functions in ES6.

In my opinion, the most useful and semantic addition is `Math.trunc`, which gives you the truncated integer value of a number.

```
console.log(Math.trunc( 1.99 ));  
console.log(Math.trunc( -1.99 ));
```



Another extension is `Math.sign` which gives you the sign of a number. Even though the `>` and `<` operators can still form proper boolean expressions, when using calculations, sometimes the `sign` function is still useful.

```
console.log(Math.sign( 5 ));  
//> 1  
  
console.log(Math.sign( '5' ));  
//> 1  
  
console.log(Math.sign( 0 ));  
//> 0  
  
console.log(Math.sign( -0 ));  
//> -0  
  
console.log(Math.sign( '' ));  
//> 0  
  
console.log(Math.sign( 'twenty' ));  
//> NaN  
  
console.log(Math.sign( NaN ));  
//> NaN
```



Other new **Math** functions include:

Function	Usage
<code>Math.cbrt</code>	cube root
<code>Math.clz32</code>	count leading zeros of 32bit integers
<code>Math.exp(num)</code>	Returns e^{num} ($e=2.718...$)
<code>Math.expm1(num)</code>	<code>Math.exp(num) - 1</code>
<code>Math.fround</code>	round to the nearest 32 bit floating point value
<code>Math.sinh</code> , <code>Math.cosh</code> , <code>Math.tanh</code> , <code>Math.asinh</code> , <code>Math.acosh</code> , <code>Math.atanh</code> , <code>Math.hypot</code>	Hyperbolic functions
<code>Math.imul</code>	32 bit integer multiplication
<code>Math.log2</code>	base 2 logarithm
<code>Math.log10</code>	base 10 logarithm
<code>Math.log(num)</code>	natural logarithm (ln) of num
<code>Math.log1p(num)</code>	<code>Math.log(num + 1)</code>

You can read about these in more detail [here](#).

In the next lesson, we will talk about the Number extensions in ES6.

