Lesson 01.03 - PROG

In this lesson:

- Math Shorthand Operators: ++, --, +=, -=, *=, /=
- Math.random(), Math.round()
- mutable vs immutable
- Math.floor(), Math.ceil()
- Math.max(), Math.min(), Math.abs()
- Math.pow(), Math.sqrt(), Math.PI
- Spread Operator: ...
- toFixed() for rounding a number to n digits
- Order of Operations of Mathematical Expressions
    - Do * and / before + and -
    - Do * and / from left to right
    - Do + and - from left to right
    - Do exponents ** before * or /
    - Do anything inside () first

1. Compare math operators to shorthand operators:

```
let x = 40;

// addition: + vs. +=
x = x + 13;
console.log('x:', x); // 53
x += 17;
console.log('x:', x); // 70

// multiplication: * vs. *=
x = x * 3;
console.log('x:', x); // 210
x *= 2;
console.log('x:', x); // 420

// subtraction: - vs. -=
x = x - 80;
console.log('x:', x); // 340
x -= 100;
console.log('x:', x); // 240

// division: / vs. /=
x = x / 2;
console.log('x:', x); // 120
x /= 3;
console.log('x:', x); // 40
```

2. Declare 3 number variables, and then do some calculations showing the order of operations of mathematical expessions:

```
let n1 = 4;
let n2 = 5;
let n3 = 8;
let tot = n1 + n2 * n3; // 4 + 40
console.log('tot:', tot); // 44
tot = (n1 + n2) * n3; // 9 * 8
console.log('tot:', tot); // 72
```

Math Object
JS has a built-in Math Object, which comes with many useful methods.

Math.random() generates a random float from 0-1, to 17 decimal points

3. Generate a random number and log it:

```
let r = Math.random();
console.log('r:', r); // 0.7492906781140873
```

Math.round() rounds off its argument to the nearest integer

4. Round off some numbers:

```
console.log(Math.round(2.4)); // 2
console.log(Math.round(2.5)); // 3
```

Math.floor() rounds down, while Math.ceil() rounds up

5. Round down y and round up z:

```
let y = 2.99;
console.log(Math.floor(y)); // 2
let z = 2.01;
console.log(Math.ceil(z)); // 3
```

Strings and numbers are immuatable (cannot be changed).
Math.floor(y) does not change y, actually.

6. Log y and z to see that they have not been changed.

```
console.log(y); // 2.99
console.log(z); // 2.01
```

7. Floor y again, but this time save the result back to y:

```
y = Math.floor(y);
console.log(y); // 2
```

random numbers

Math.random() generates a random float from 0-1, so to get a larger number, just multiply by some value:

8. Generate a random float from 0-100:

```
r = Math.random() * 100;
console.log('r:', r); // 74.92906781140873
```

To get an integer, round, floor or ceil the random value:

9. Floor a random number multiplied by 100 to get an integer from 0-99:

```
r = Math.floor(Math.random() * 100);
console.log('r:', r); // some integer from 0-99
```

** getting a random integer in a range**

To get a random integer in a range, multiply by the range, and add the minimum:

10. Generate a number integer in the 50-100 range:

```
randInt = Math.floor(Math.random() * 51 + 50);
console.log('randInt:', randInt); // some value between 50-100
```

max(), min(), pow(), abs(), sqrt(), PI

Math.max() returns the greatest of the multiple values passed to it

11. Find the maximum of a set of numbers. Save the result to a variable and log it:

```
let maxi = Math.max(3, 6, 8, 2, 12, 4, 10);
console.log('maxi:', maxi); // 12
```

12. Pass nums to the Math.max() method, saving the result to a variable, numsMax.
    Then log numsMax, which we expect to be 12:

```
let nums = [3, 6, 8, 2, 12, 4, 10];
maxi = Math.max(nums);
console.log('maxi:', maxi); // NaN
```

We get NaN because Math.max() expects comma-separated numbers, but not as an array.

Spread Operator ...
The spread operator is three dots that "destructures" (gets rid of) the array,
while leaving the individual array items.

13. Find the max value of nums by destructuring the array:

```
maxi = Math.max(...nums);
console.log('maxi:', maxi); // 12
```

Math.min() returns the smallest of the multiple values passed to it

14. Get the min value of the nums array. Use the spread operator:

```
let mini = Math.min(...nums);
mini = Math.min(mini); // 2
console.log('mini:', mini); // 2 (without ... we get NaN)
```

15. Raise 5 to the 4th power using the Math.pow() method:

```
let pwr = 5 ** 4;
console.log('5 to 4th power:', pwr);
pwr = Math.pow(5,3);
console.log('5 to 3rd power:', pwr);
```

Math.abs() returns the absolute value of its argument, meaning it makes it positive

16. Use Math.abs() to get the absolute value of a negative number:

```
console.log(Math.abs(-7)); // 7
```

Math.sqrt() returns the square root of its argument

17. Find the square root of a sumber:

```
console.log(Math.sqrt(36)); // 6
```

18. Math.PI returns pi to 17 digits:

```
console.log(Math.PI); // 3.141592653589793
```

toFixed() is a method called on a float
It returns a float to the specified number of decimal places, but as a string.

19. Round PI to 5 digits, and log its data type, which is 'string':

```
let pi5 = Math.PI.toFixed(5);
console.log("pi5:", pi5, typeof(pi5)); // pi5 3.14159 string
```

20. Add pi5 to itself; pi5 is a string, so the plus sign concatenates:

```
let twoPI = pi5 + pi5;
console.log("twoPI:", twoPI); // 3.141593.14159
```

21. Pass pi5 to the Number() method to convert the string to a number.
    Now, the value can be used for addition:

```
twoPI = Number(pi5) + Number(pi5);
console.log("twoPI:", twoPI, typeof(twoPI)); // 6.28318 string
```

parseFloat() takes a number-like string, and returns a floored integer

22. Declare a stringy float and then pass it to the parseFloat method:

```
let price = "7.89";
price = parseFloat(price);
console.log("price:", price, typeof(price)); // price: 7 number
```

The += shorthand operator also works for string concatenation:

23. Concatenate a greeting, line by line with += :

```
let greeting = "Good";
greeting += " Morning!";
console.log(greeting); // Good Morning
```

operators ++ and -- increment and decrement a value by 1

24. Declare a number variable, and increment and decrement it by 1.
    The changes "stick" without having to save the variable to itself:

```
let score = 10;
score++;
console.log('score:', score); // score: 11
score--;
console.log('score:', score); // score: 10
```

// DONE: Lesson 01.03 // NEXT: Lab 01.03 // THEN: Lesson 02.01