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

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
```

~~ shorthand for Math.floor ~~ can be used in place of Math.floor to round down a number

```
randInt;
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

LAB 01.03

1. Generate a random integer from 10-19:

```
r = Math.floor(Math.random() * 10) + 10; // 10-19
console.log('r 10-19:', r);
// or:
r = ~~(Math.random() * 10) + 10; // 10-19
// or:
r = Math.ceil(Math.random() * 10) + 9; // 10-19
console.log('r 10-19:', r);
```

2. Generate a random integer from 26-50:

```
r = Math.floor(Math.random() * 25) + 26; // 26-50
console.log('r 26-50:', r);
// or:
r = ~~(Math.random() * 25) + 26; // 26-50
console.log('r 26-50:', r);
// or:
r = Math.ceil(Math.random() * 25) + 25; // 26-50
console.log('r 26-50:', r);
```

3. Find the maximum value from this array:

```
nums = [13, 35, 21, 57, 61, 39, 12, 16];
let numsMax = Math.max(...nums);
console.log('numsMax:', numsMax);
```

4. Find the square root of s.

```
let s = 144;
console.log('sqrt of 144:', Math.sqrt(144)); // 12
```

5. Round down x:

```
x = 9.9999;
console.log('Math.floor(9.9999):', Math.floor(x)); // 9
// or:
console.log('~~9.9999:', ~~x); // 9
```

6. Supply values for x, y and exp to match expected output.

```
x = 14;
y = 2;
let exp = x ** y;
console.log('exp:', exp); // exp: 196
expon = Math.pow(14,2);
console.log('exp:', exp); // exp: 196
```

7. The famous formula A = πr² tells us that the area of a circle equals PI times the radius squared. Given a circle of radius 4, find the area:

```
let radius = 4;
let area = Math.PI * radius ** 2;
console.log('area:', area); // 50.26548245743669
```

8. The hypotenuse (c) of a right triangle is obtained by the formula:

```
a² + b² = c², where a and b are the other two sides.
// Find the hypotenuse of a triangle, where side a is 5 and side b is 12.
let sideA = 5;
let sideB = 12;
let cSquared = sideA**2 + sideB**2; // c² = a² + b²
let sideC = Math.sqrt(cSquared);
console.log('sideC (hypotenuse):', sideC);
```

9. Generate two random floats in the 0-10 range. Round these numbers off to 3 decimal places. Find the sum of these two numbers, also rounded to 3 decimal places. HINT: Use Number() (or + shorthand instead of Number) and toFixed()

```
let r1 = Math.random() * 10;
let r2 = Math.random() * 10;
let sum = (Number(r1) + Number(r2)).toFixed(3);
console.log('Number(r1) + Number(r2):', sum); // sum r1 + r2: 13.87
// or:
sum = (+r1 + +r2).toFixed(3);
console.log('(+r1 + +r2).toFixed(3):', sum); // sum r1 + r2: 13.87
```

10. Given this baseball player and his statistics:

Player: Vladimir Guerrero Jr. Team: Toronto Blue Jays Year: 2021

Stats: PA AB R H 2B 3B HR RBI 698 604 123 188 29 1 48 111

A. Calculate the player's Batting Average (AB); Batting Average equals hits (H) divided by at bats (AB). It is customary for SLG to be rounded to 3 decimal places, and displayed without a leading zero, so .321 not 0.321.

B. Calculate the player's Total Bases (TB); Total Bases equals the sum of a player's hits (H), plus their doubles (2B), plus twice their triples (3B), plus three times their home runs (HR):

C. Calculate the player's Slugging Percentage (SLG). Slugging Percentage equals total bases (TB) divided by at bats (AB). It is customary for SLG to be rounded to 3 decimal places, and displayed without a leading zero, so .521 not 0.521. */

Solution:

Declare variables for the numbers needed by the TB and SLG formulas:

```
// for TB formula:
let H = 188;
let _2B = 29;
let _3B = 1;
let HR = 48;
// for SLG formula:
let AB = 604;

// A. Calculate Batting Average (AB):
let BA = (H / AB).toFixed(3);
console.log('BA:', BA); // 0.311

// B. Calculate Total Bases (TB):
let TB = H + _2B + (_3B * 2) + (HR * 3);
console.log('TB:', TB); // 363

// C. Calculate Slugging Percentage (SLG):
let SLG = (TB / AB).toFixed(3);
console.log('SLG:', SLG); // 0.601

/* RESEARCH BONUS:
```

```
   Google / research how to get rid of the first character of a string.
   This will allow us to get rid of the leading zero in BA and SLG,
   so that we have the desired format .311 and .601 (NOT 0.311, 0.601).

   Solution:
   str.slice(start_index, end_index) copies a portion of a string,
   from the start to end index. If only one argument is passed
   in, it slices (copies) all the way to the end of the string.
   We want all but the first character (leading zero), so do slice(1).
   BA and SLG are already strings, since they have been rounded to
   3 digits with toFixed(3).
    BA = BA.slice(1);
*/
console.log('BA, no leading 0:', BA); // 0.311

SLG = SLG.slice(1);
console.log('SLG, no leading 0:', SLG);
```

11. Solve the Pizza Lovers Dilemma /* Figure out which pizza is cheaper by the square inch:

- The medium pizza for $14.99 has a diameter of 12".
- The large pizza for $19.99 has a diameter of 16".
- Find the price per square inch of each pizza.
- Answer should be in cents to the nearest cent: 13¢
- A = πr² is the formula to use to find the area, as we may assume that both pizzas are perfect circles.
- The diameter of a circle is equal to twice the radius.
- declare medium pizza variables:

```
let mediumPrice = 14.99; // $14.99
let mediumRadius = 12 / 2; // 6 inches
// calculate the area in square inches of the medium pizza
let mediumArea = Math.PI * mediumRadius**2;
// divide the price by the area to get the price per square inch:
let mediumPricePerSqIn = mediumPrice / mediumArea;
console.log('mediumPricePerSqIn:', mediumPricePerSqIn); //
0.1325406998304173
// round off the answer to two decimal places, get rid of the
// leading zero and decimal place, and add the cent symbol:
mediumPricePerSqIn = mediumPricePerSqIn.toFixed(2) + '¢';
mediumPricePerSqIn = mediumPricePerSqIn.slice(2);
console.log('medium pizza price per sq in:', mediumPricePerSqIn);
// medium pizza price per sq in: 13¢

// repeat the medium pizza steps for the large pizza:
let largePrice = 19.99;
```

```
let largeRadius = 16 / 2;
let largeArea = Math.PI * largeRadius**2;
let largePricePerSqIn = largePrice / largeArea;
console.log('largePricePerSqIn:', largePricePerSqIn); //
0.09942210351271837
largePricePerSqIn = largePricePerSqIn.toFixed(2) + '¢';
largePricePerSqIn = largePricePerSqIn.slice(2);
console.log('large pizza price per sq in:', largePricePerSqIn);
// large pizza price per sq in: 10¢
```

END: Lab 01.03

NEXT: Lesson 02.01