noble desktop

UNIT 01

LESSON 01.04

JavaScript BOOTCAMP

Math Object

math operators: +, -, *=, /, **, %

Math.random() .round() .floor() .ceil()

.max() .min() .abs() .pow() .PI

.toFixed()

shorthand operators: +=, -=, *=, /=

Naturally, we can do math with numeric variables.

Mathematical operators include:

```
+   add
-   subtract
*   multiply
/   divide
**  exponent (power of)
%   modulo (remainder)
```

1. Declare two number variables, and use them for some simple calculations:

```
    let n1 = 10;
    let n2 = 3;

    console.log(n1 + n2); // 13
    console.log(n1 - n2); // 7
    console.log(n1 * n2); // 30
    console.log(n1 / n2); // 3.333333333
    console.log(n1 ** n2); // 1000
    console.log(n1 % n2); // 1
```

Results of mathematical calculations can be stored in variables.

2. Change the values of **n1** and **n2**, and do some more calculations, saving the results to variables:

```
    n1 = 6;
    n2 = 2;

    let sum = n1 + n2;
    console.log(sum); // 8

    let diff = n1 - n2;
    console.log(diff); // 4

    let prod = n1 * n2;
    console.log(prod); // 12

    let quot = n1 / n2;
    console.log(quot); // 3

    let expo = n1 ** n2;
    console.log(expo); // 36

    let modul = n1 % n2;
    console.log(modul); // 0
```

Order of Operations of Mathematical Expressions

- Do * and / before + and -
- Do * and / from left to right
- Do + and - from left to right
- Operations inside () are done first

3. Declare a third number variable, and do some math that shows how parentheses can affect the result:

```
    n1 = 4;
    n2 = 5;
    let n3 = 8;

    let tot = n1 + n2 * n3; // 4 + 40
    console.log(tot); // 44

    tot = (n1 + n2) * n3; // 9 * 8
    console.log(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 16 decimal points:

4. Generate a random number and log it:

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

Math.round() rounds off its argument:

5. Round off a number:

```
    let x = Math.round(2.5);
    console.log(x); // 3
```

Math.floor() rounds *down* its argument:

6. Round down a number:

```
    let y = Math.floor(2.999);
    console.log(y); // 2
```

Math.ceil() rounds *up* its argument:

7. Round up a number:

```
    let z = Math.ceil(2.001);
    console.log(z); // 3
```

getting a random number greater than 1

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

8. Generate a random number and multiply it by 100:

```
    let rando = Math.random() * 100;
    console.log(rando)  // some number between 0-100
```

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

9. Round down a random number multiplied by 100:

```
    let randInt = Math.floor(Math.random() * 100);
    console.log(randInt); // some integer between 0-99
```

getting a random integer in a range

To get a random integer in a range, multiply by the range span and then add the starting value.

10. Round down a random number multiplied by 50 and then add 50 to get a number in the 50-100 range:

```
let rand = Math.ceil(Math.random() * 50 + 50);
console.log(rand); // some value between 50-100
```

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

11. Find the maximum of a set of numbers:

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

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

12. Find the minimum of a set of numbers:

```
let mini = Math.min(3, 6, 8, 2, 12, 4, 10);
console.log(mini); // 2
```

Math.pow() takes two arguments: a number and a power to raise it to:

13. Raise a number to a power using the **Math.pow()** method:

```
let pwr = Math.pow(5, 4);
console.log(pwr); // 625 (5x5x5x5)
```

However, as we have seen, the ** operator does the same thing as Math.pow():

14. Raise a number to a power using the ** operator:

```
let powr = 5 ** 4;
console.log(powr); // 625 (5x5x5x5)
```

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

15. Use **Math.abs()** to get the absolute value:

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

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

16. Use **Math.sqrt()** to find a square root:

```
let sqRt = Math.sqrt(81);
console.log(sqRt); // 9
```

Math.PI returns the famous constant. If you save it, it should be to a **const**, uppercase:

17. Get **PI** to 16 digits:

```
const PI = Math.PI;
console.log(PI); // 3.141592653589793
// PI = 'apple'; // ERROR
```

toFixed() is a method called on a float. It returns a float with the number of decimal places in the argument:

18. Round PI to 2 digits. The rounded value is actually a string:

```
let pi2 = PI.toFixed(2);
console.log(pi2, typeof(pi2)); // 3.14 string
```

19. Try doing addition with **pi2**. The plus sign does concatenation, because it's working with a string:

```
console.log(pi2 + pi2); // 3.143.14
```

*math shorthand operators: += -= = /=

Math shorthand operators make math more concise by eliminating the need to repeat a variable:

20. Try these math shorthand operators:

```
x = 20;

x = x + 35; // add
console.log(x); // 55

x += 15; // add
console.log(x); // 70

x = x * 3; // multiply
console.log(x); // 210
```

```
    x *= 2; // multiply
    console.log(x); // 420

    x = x - 80; // subtract
    console.log(x); // 340

    x -= 100; // subtract
    console.log(x); // 240

    x = x / 4; // divide
    console.log(x); // 60

    x /= 3; // divide
    console.log(x); // 20
```

- **END Lesson 01.04**
- **NEXT: Lab 01.04**
- **NEXT: Lesson 02.01**