Week 1 - Fundamentals - Tues. Recap

Sigma

Implement function sigma(n) - given number n, return the sum of positive integers from 1 to n, inclusive. E.g.: sigma(3) == 1+2+3; sigma(5) == 1+2+3+4+5.

```
//  Num's fractional component is ignored.
//  Non-positive numbers return zero.
function iSigma(num)
{
  var sum = 0;
  for (var counter = 1; counter <= num;
                        counter++) {
    sum += counter;
  }
  return sum;
}
```

Factorial

Implement factorial(n) - given n, return the product (multiplication) of positive ints from 1 to n (inclusive). E.g.: factorial(3) == 1*2*3; factorial(5) == 1*2*3*4*5.

```
//  Num's fractional component is ignored.
//  Non-positive numbers return zero.
function iFactorial(num)
{
  if (num < 1) { return 0; }
  var factorial = 1;
  for (var counter = 2; counter < num;
                        counter++) {
    factorial *= counter;
  }
  return factorial;
}
```

Week 1 - Fundamentals - Wednesday

This week you will familiarize yourself with basic programming constructs.
Some or all of the below concepts will be used in this week's challenges.

for loops, while loops
Math.random, Math.floor, Math.ceil
if / else statements
console.log
% (called modulus)

Threes and Fives

Add all the values from 100 and 4000000 (inclusive) that are evenly divided by 3 or 5 but not both. Display this value in the console.

Second challenge: Create one where *start* and *end* values are customizable (defaults 100 and 4000000).

Answer:

Generate Coin Change

Implement a function generateCoinChange() that accepts a parameter for the number of cents, and computes how to represent that amount with the smallest number of coins. Console.log the result.

Answer:

Tomorrow: Dick and Jane visit Las Vegas, while Spot reduces large numbers to a pair of paws