Week 1 - Fundamentals - Wed. Recap

Threes and Fives

Implement a function threesAndFives() that adds all values from an optional parameter (default: 100) up to another optional parameter (default: 4000000) that are divisible by 3 or 5 but not both.. Console.log the result.

```
function threeAndFive(startVal, endVal)
{
  var result = 0;
  if (startVal === undefined) {
    startVal = 100;
  }
  if (endVal === undefined) {
    endVal = 4000000;
  }
  for (var count = startVal;
       count <= endVal; count++)
  {
    if (  (count % 3 == 0)
       || (count % 5 == 0))
    {
      if (  (count % 3 == 0)
         && (count % 5 == 0))
      {
        continue;
      }
      result += count;
    }
  }

  console.log("The 3 & 5 sum [%d-%d] is %d",
              startVal, endVal, result);
}
```

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.

```
// Greedy: Start with largest, work down.
// dollars quarters dimes nickels pennies
// A more elegant solution uses % and /
function generateCoinChange(amount)
{
  console.log("%d cents is:", amount);
  var numDollars = 0;
  var numQuarters = 0;
  var numDimes = 0;
  var numNickels = 0;
  var numPennies = 0;

  while (amount >= 100) {
    numDollars++;
    amount -= 100;
  }
  while (amount >= 25) {
    numQuarters++;
    amount -= 25;
  }
  while (amount >= 10) {
    numDimes++;
    amount -= 10;
  }
  while (amount >= 5) {
    numNickels++;
    amount -= 5;
  }
  while (amount >= 1) {
    numPennies++;
    amount -= 1;
  }

  console.log("%d dollars",  numDollars);
  console.log("%d quarters", numQuarters);
  console.log("%d dimes",    numDimes);
  console.log("%d nickels",  numNickels);
  console.log("%d pennies",  numPennies);
}
```

Week 1 - Fundamentals - Thursday

This week you will familiarize yourself with basic programming constructs. Here is a list of methods for you to study. Some or all of these will be used to solve this week's challenges.

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

Statistics to Doubles

Implement a 'die' that randomly returns an integer between 1 and 6 inclusive. Roll a pair of these dice, tracking the statistics until doubles are rolled. Display the *number of rolls*, *min*, *max*, and *average*.

Answer:

Sum To One Digit

Implement a function sumToOne() that, given a number, sums that number's digits repeatedly until the sum is only one digit. Return that final one digit result.

Answer:

Tomorrow: (Mathematical) Law & Order: the Fibonacci Episode