

# Week 1 - Fundamentals - Thurs. Recap



## Statistics to Doubles

Implement 'dice' that randomly return ints between 1 and 6 inclusive. Roll dice, tracking stats until doubles are rolled. Display number of rolls, min, max, average.

```
// Get num from 0 to "almost 1". Make it
// from 0 to "almost 6". Drop the decimal to
// make [0,1,2,3,4,5]. Add 1: [1,2,3,4,5,6]

function rollDie()
{
    var rand = Math.random() * 6;
    rand = Math.floor(rand) + 1;
    return rand;
// return Math.floor(Math.random() * 6) + 1;
}

function statsBeforeDoubles()
{
    var minRoll = 12;
    var maxRoll = 2;
    var sum = 0;
    var numRolls = 0;
    var die1, die2;
    do {
        die1 = rollDie();
        die2 = rollDie();
        var total = die1 + die2;

        if(minRoll > total) { minRoll = total; }
        if(maxRoll < total) { maxRoll = total; }
        sum += total;
        numRolls++;
    } while (die1 != die2);

    console.log(numRolls, "rolls.");
    console.log("Average:", sum/numRolls);
    console.log("Lowest:", minRoll);
    console.log("Biggest:", maxRoll);
    console.log("Last:", die1, "and", die2);
}
```

## Sum To One Digit

Implement sumToOne(num) that sums num's digits repeatedly until the sum is one digit. Return the result.

```
function sumToOneDigit(num)
{
    if (num < 1)    {    return 0;    }

    num = Math.floor(num);
    while (num >= 10)
    {
        while (num)
        {
            var digitSum;
            var onesDigit = num % 10;
            num -= onesDigit;
            num /= 10;
            digitSum += onesDigit;
        }
        num = digitSum;
    }
    return num;
}

// Or if you're clever, how about this?
function sumTo1Pro(num)
{
    if (num < 1) return 0;
    return (num - 1) % 9 + 1;
}
```

# Week 1 - Friday - Fundamentals

---



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*

## **Fibonacci**

Implement the Fibonacci function, a famous mathematical equation that generates a numerical sequence such that each number is the sum of the previous two. The Fibonacci numbers at index 0 and 1, coincidentally, have values of 0 and 1. Your function should accept an argument of which Fibonacci number.

Examples: fibonacci(2) = 1, fibonacci(3) = 2, fibonacci(4) = 3, fibonacci(5) = 5, etc.

**Answer:**

**Weekend:** parsley, sage, rosemary, and ... modulo madness