

**Udemy**  
**JavaScript Fundamentals 2**  
**Coding Challenge #1**

### The Problem

There are two gymnastic teams; the Dolphins and the Koalas.

Each team competes 3 times and the average of the 3 scores is calculated (one average score per team.).

A team only wins if it has at least double the average score of the other team, otherwise, no team wins.

1. Create an arrow function 'calcAverage' to calculate the average of 3 scores.
2. Use the function to calculate the average for both teams.
3. Create a function 'checkWinner' that takes the average score of each team as parameters ('avgDolphins' and 'avgKoalas'), and then logs the winner to the console, together with the victory points, according to the rule above. Example: "Koalas win (30 vs 13)".
4. Use the 'checkWinner' function to determine the winner for both DATA 1 and DATA 2.
5. Ignore draws this time.

**TEST DATA 1:** Dolphins: 44, 23 and 71. Koalas: 65, 54 and 49.

**TEST DATA 2:** Dolphins: 85, 54, 41. Koalas: 23, 34 and 27.

### Method

First create the team name variables. Also implement variables to hold the team scores. This means that we will be able to change the scores here and the functions should be able to use the data throughout the rest of the code.

```
//Create Team Names:
//Koala Name and Scores
const teamOne = 'Koalas';
const teamOneScoreOne = 5 ;
const teamOneScoreTwo = 3 ;
const teamOneScoreThree = 1 ;
const koalaAverage = () => {
    return ((teamOneScoreOne + teamOneScoreTwo +
teamOneScoreThree)/3);
}
console.log(koalaAverage());
```

The first arrow function is the variable 'koalaAverage'. Notice that NO PARAMETERS have been set. This is because we do not want any parameters implemented. We want to use

the variables that we defined earlier. We cannot put these variable names into the parameter parentheses. This is because within the parentheses any variable names we use will operate a bit like local variables and are not connected with any variables external to the parentheses. In the return block we can then call the variables that we defined earlier. We can then use the `console.log(koalaAverage());` - with no arguments – to log and check that our function works.

The number returned to the console is 9. We know that our 'koalaAverage' function works.

We can now effectively repeat this code for the Dolphins.

The functions work and we now have two functions that we can potentially call on later.

However we might need a function with 3 parameters that can take data later. Here is an example arrow function that can do this:

```
//Create Function that Determines Average
const calcAverage = (scoreOne, scoreTwo, scoreThree) => {
  return ((scoreOne + scoreTwo + scoreThree)/3);
}
console.log(`The average of the 3 scores is: ${calcAverage(1, 3, 5)}.`);
```

We now have 2 different ways of calculating the average for our teams. Taking our second function as an example we could;

```
console.log(`The average score for ${teamOne} is
${calcAverage(teamOneScoreOne, teamOneScoreTwo,
teamOneScoreThree)}.`)
```

In this way, we have used our variables as the parameter arguments. This means we can change the variables at the beginning of our code and this will still output the same answer. We can now create the same line, but for the Koalas.

The advantage of this is that we now only have one function performing the same roll our two separate functions did earlier. We call the function with different parameters.

We can now store our averages in two new variables:

```
const avgKoala = calcAverage(teamOneScoreOne, teamOneScoreTwo,
teamOneScoreThree);
const avgDolphins = calcAverage(teamTwoScoreOne, teamTwoScoreTwo,
teamTwoScoreThree);
```

This is a simple way of calling our functions. Note that we have used the variables that we declared earlier. This way we can edit the variables at the top of the document and the rest of the code will continue to executed correctly.

## chkWinner

We now need to determine who the winner is. If we were writing an if/else statement it might look like this;

```
if (avgKoala >= avgDolphins * 2) {  
    console.log('Koalas win.');
```

```
} else if (avgDolphins >= avgKoala * 2) {  
    console.log('Dolphins win.');
```

```
} else {  
    console.log('There are no winners.');
```

```
}
```

Because we can only have a winner if a team scores double the average of the other score we need a simple mathematical calculation to determine if this condition has been met. Our condition parameter looks like this:

```
avgKoala >= avgDolphins * 2
```

If the average Koala score is greater than or equal to double the average Dolphin score.

We can verify that the code works by checking our console and using various figures.

Now that we have an idea of the type of decision making we want a function to use, we can implement this into a chkWinner function.

And here it is:

```
//chkWinner Function  
const chkWinner = function() {  
    if(avgKoala >= avgDolphins * 2) {  
        return 'Koalas win!';  
    } else if (avgDolphins >= avgKoala * 2) {  
        return 'Dolphins win!';  
    } else {  
        return 'Neither team achieved an average score of double  
their opponents average score. There are no winners.';  
    }  
} ;  
console.log(chkWinner());
```

Our function has no parameters. This is because our parameters have already been set in separate functions.

Notice that when we call our function with a console.log the parentheses have no values. This is because the values are not set here. Without the parentheses the function would not be successfully called.

### Results

**TEST DATA 1:** Dolphins: 44, 23 and 71. Koalas: 65, 54 and 49.

---

The Koalas scored an average of 56 point(s). The Dolphins scored an average of 46 point(s).  
Neither team achieved an average score of double their opponents average score. There are no winners.

---

**TEST DATA 2:** Dolphins: 85, 54, 41. Koalas: 23, 34 and 27.

---

The Koalas scored an average of 28 point(s). The Dolphins scored an average of 60 point(s).  
Dolphins win!

---

### Final Code Block:

```
'use strict';

//Create Team Names:
//Koala Name and Scores
const teamOne = 'Koalas';
const teamOneScoreOne = 23 ;
const teamOneScoreTwo = 34 ;
const teamOneScoreThree = 27 ;
const koalaAverage = () => {
    return ((teamOneScoreOne + teamOneScoreTwo + teamOneScoreThree)/3);
}

//console.log(`The average score for the ${teamOne} is
${koalaAverage()}.`);

//Dolphin Name and Scores
const teamTwo = 'Dolphins';
const teamTwoScoreOne = 85
const teamTwoScoreTwo = 54
const teamTwoScoreThree = 41
const dolphinAverage = () => {
    return ((teamTwoScoreOne + teamTwoScoreTwo + teamTwoScoreThree)/3);
```

```

}

//AVERAGE FUNCTION
const calcAverage = (scoreOne, scoreTwo, scoreThree) => {
    return ((scoreOne + scoreTwo + scoreThree)/3);
}

const avgKoala = calcAverage(teamOneScoreOne, teamOneScoreTwo,
teamOneScoreThree);
const avgDolphins = calcAverage(teamTwoScoreOne, teamTwoScoreTwo,
teamTwoScoreThree);
console.log(`The ${teamOne} scored an average of ${avgKoala} point(s).
The ${teamTwo} scored an average of ${avgDolphins} point(s).`);

//chkWinner Function
const chkWinner = function() {
    if(avgKoala >= avgDolphins * 2) {
        return 'Koalas win!';
    } else if (avgDolphins >= avgKoala * 2) {
        return 'Dolphins win!';
    } else {
        return 'Neither team achieved an average score of double their
opponents average score. There are no winners.';
    }
} ;
console.log(chkWinner());

```