**Coin Flipping**

Let's say we have a function that simulates flipping a coin...

*function tossCoin() {*

*return Math.random() > 0.5 ? "heads" : "tails";*

*}*

And we want to use it in a function that sees how long it will take to have a coin land on "heads" five times in a row.

*function fiveHeadsSync() {*

*let headsCount = 0;*

*let attempts = 0;*

*while(headsCount < 5) {*

*attempts++;*

*let result = tossCoin();*

*console.log(`${result} was flipped`);*

*if(result === "heads") {*

*headsCount++;*

*} else {*

*headsCount = 0;*

*}*

*}*

*return `It took ${attempts} tries to flip five "heads"`;*

*}*

*console.log( fiveHeadsSync() );*

*console.log( "This is run after the fiveHeadsSync function completes" );*

This will take an indeterminate amount of time. If we run the synchronous function above, it will prevent any code that comes after from running while this is taking place.

Your mission is to rewrite the above function using Promises. Make sure your fiveHeads function will call the resolve function when the coin has flipped "heads" five times in a row.

*function fiveHeads() {*

*return new Promise( (resolve, reject) => {*

*// your code here!*

*});*

*}*

*fiveHeads()*

*.then( res => console.log(res) )*

*.catch( err => console.log(err) );*

*console.log( "When does this run now?" );*

* Top of Form
* Complete the fiveHeads function so that it will resolve() when "heads" has been flipped five times
* Make the fiveHeads function reject() when the coin has been flipped more than 100 times
* Bottom of Form