ES6 - Use Destructuring Assignment to Pass an Object as a Function's Parameters

In some cases, you can destructure the object in a function argument itself.

Consider the code below:

const profileUpdate = (profileData) => {  
const { name, age, nationality, location } = profileData;  
// do something with these variables  
}

This effectively destructures the object sent into the function. This can also be done in-place:

const profileUpdate = ({ name, age, nationality, location }) => {  
/\* do something with these fields \*/  
}

This removes some extra lines and makes our code look neat.

This has the added benefit of not having to manipulate an entire object in a function; only the fields that are needed are copied inside the function.

Use destructuring assignment within the argument to the function halfto send only maxand mininside the function.

Run the TestsReset All CodeAsk for help

statsshould be an object.

half(stats)should be 28.015

Destructuring was used.

const stats = {

max: 56.78,

standard\_deviation: 4.34,

median: 34.54,

mode: 23.87,

min: -0.75,

average: 35.85

};

const half = (function() {

"use strict"; // do not change this line

// change code below this line

return function half(stats) {

// use function argument destructuring

return (stats.max + stats.min) / 2.0;

};

// change code above this line

})();

console.log(stats); // should be object

console.log(half(stats)); // should be 28.015