Let's craft some JavaScript assignments to practice using the spread operator (...). These will cover array spreading, object spreading, function arguments, and some more advanced use cases.

Array Spreading:

1. Combining Arrays:

JavaScript

const arr1 = [1, 2, 3];

const arr2 = [4, 5, 6];

// Use the spread operator to create a new array that combines arr1 and arr2.

1. Copying Arrays:

JavaScript

const originalArray = [10, 20, 30];

// Use the spread operator to create a \*new\* array that is a copy of originalArray. (Important: Why is this better than just `newArray = originalArray`?)

1. Adding Elements to an Array:

JavaScript

const numbers = [1, 2, 3];

// Use the spread operator to create a new array with the numbers array and the values 4, 5, and 6 added to it.

Object Spreading:

1. Merging Objects:

JavaScript

const obj1 = { a: 1, b: 2 };

const obj2 = { c: 3, d: 4 };

// Use the spread operator to create a new object that combines obj1 and obj2.

1. Overriding Properties:

JavaScript

const defaults = { color: "red", size: "medium" };

const userOptions = { size: "large" };

// Use the spread operator to create a new object that combines defaults and userOptions, with userOptions overriding the default values.

1. Adding Properties to an Object:

JavaScript

const person = { name: "Alice", age: 30 };

// Use the spread operator to create a new object with the person object and a new property 'city' with the value "New York".

Spread Operator with Functions:

1. Passing Arguments to Functions:

JavaScript

function add(x, y, z) {

return x + y + z;

}

const numbers = [5, 10, 15];

// Use the spread operator to call the add function with the elements of the numbers array as arguments.

1. Rest Parameter and Spread Operator:

JavaScript

function printNumbers(a, b, ...rest) {

console.log("First two:", a, b);

console.log("Rest:", rest);

}

// Call printNumbers with some arguments, demonstrating how the rest parameter and spread operator work together.

Advanced Scenarios:

1. Spread Operator and Destructuring:

JavaScript

const person = { name: "Bob", age: 25, address: { city: "London" } };

const { ...rest, address: { city } } = person;

// Explain what the 'rest' object and 'city' variable will contain after this destructuring.

1. Creating a Shallow Copy with Modifications:

JavaScript

const product = { name: "Laptop", price: 1200, features: ["fast CPU", "large RAM"] };

// Use the spread operator to create a \*shallow\* copy of the product object. Then, modify the 'features' array in the copy (e.g., add a new feature). Explain why the original product object is also affected.

1. Spread Operator and Iterables:
   * Demonstrate how the spread operator can be used to convert a string into an array of characters, or a NodeList (returned by querySelectorAll) into a regular array.

These assignments cover a variety of use cases for the spread operator, from basic array and object manipulation to more advanced scenarios involving functions, destructuring, and iterables. They should give you a good grasp of how to use this powerful operator effectively. Let me know if you have any other questions or would like more specific examples!