Corporate Training Program / Front-End Class
Fundamental Learning Assignment - Part 2

Assignment 0: Treehouse courses

1. JavaScript Basics
2. JavaScript Functions
3. JavaScript Loops
4. JavaScript Arrays
5. JavaScript Objects
6. Object-Oriented JavaScript
7. Callback Functions
8. Array Iteration Methods

Total length: 14 hours

Corporate Training Program / Front-End Class
Fundamental Learning Assignment - Part 2

Assignment 1: Function and Array

Complete the function below to find the max number of the passing array of numbers.

Reminder: you *can't* use built-in Math.max() or sort() to complete this assignment.

```
function max(numbers) {
        // your code here
}
console.log(max([1, 2, 4, 5])); // expected output: 5
console.log(max([5, 2, 7, 1, 6])); // expected output: 7
```

Corporate Training Program / Front-End Class
Fundamental Learning Assignment - Part 2

Assignment 2: Object

Complete the function below to calculate the result of the passing object.

```
function calculate(args) {
        // your code here
}
console.log(calculate({ n1: 2, n2: 3, op: '+' })); // expected output: 5
console.log(calculate({ n1: 5, n2: 2, op: '-' }));  // expected output: 3
console.log(calculate({ n1: 1, n2: 6, op: 'x' }));  // expected output: 'Not supported'
```

Assignment 3: Function, Array, and Object

Complete the function below to calculate the total price of all products after applying a discount.

```javascript
function calculate(data) {
        // your code here
}
const discountedPrice = calculate({
        discount: 0.1,
        products: [
                {
                        name: "Product 1",
                        price: 100
                },
                {
                        name: "Product 2",
                        price: 700
                },
                {
                        name: "Product 3",
                        price: 250
                }
        ]
});
console.log(discountedPrice) // show the total price of all products after applying a discount
```

Assignment 4: Algorithm Practice

Given an array of integers, return indices of the two numbers such that they add up to a specific target. You may assume that each input would have exactly one solution, and you may not use the same element twice.

```
function twoSum(nums, target) {
        // your code here
}
/*
        For example:
                twoSum([2, 7, 11, 15], 9);
        Should return:
                [0, 1]
        Because:
                nums[0] + nums[1] is 9
*/
```

Assignment 5: Callback Function

Complete the function below to show a delayed result in the console.

```
function delayedResult(n1, n2, delayTime, callback) {
        // your code here
}
delayedResult(4, 5, 3000, function(result) {
        console.log(result);
}); // 9 (4+5) will be shown in the console after 3 seconds
delayedResult(-5, 10, 2000, function(result) {
        window.alert(result);
}); // 5 (-5+10) will be shown in an alert dialog after 2 seconds
```

Hint: You should use window.setTimeout() for time scheduling.