Andela

Andela Nigeria Cycle 45 Technical Challenge.

Instructions for submission
Create an account on codepen.io and attempt **any 1** of the questions. You are required to make use of **only** HTML, CSS and JavaScript, and **NO FRAMEWORKS**. Please submit via this form before **10am on Sunday, April 28th, 2019.**

Question 1

Dice Gambling	
Context	A dice gambling game
Task	Create a function that takes an array consisting of dice rolls from 1- 6. Return the sum of your rolls with the following conditions: 1. If a 1 is rolled, that is bad luck. The next roll counts as 0. 2. If a 6 is rolled, that is good luck. The next roll is amplified by a factor of 2. 3. The array length will always be 3 or higher. **Examples** ● rolls([1,2,3]) ➞ 4 ● // The second roll, 2, counts as 0 as a result of rolling 1. ● ● rolls([2,6,2,5]) ➞ 17 ● // The 2 following the 6 was amplified by 2. ● ● rolls([6, 1, 1]) ➞ 8 ● // The first roll makes the second roll worth 2, but the // second roll was still 1 so the third roll doesn't count. **Notes** Even if a 6 is rolled after a 1, 6 isn't summed but the 6's "effect" still takes place.
UI Design	● An input field that takes in the dice rolls ● A button to run the program. ● A div to display result.

Question 2

A Knapsack	
Context	Given a knapsack with a certain weight capacity, fill it with loot from a list of items to achieve the highest value possible.

Task	The function takes two parameters: an **int** specifying the maximum weight the knapsack can hold, and an **array** of item objects to choose from. Each item object has a name, a weight, and a value. The total weight of all the chosen items cannot exceed the capacity of the knapsack.
	The function should return an object containing the capacity of the bag, a list of items that were added to the bag (in the same order that they were given), the total weight of those items, and the total value of the items.
	Example
	knapsack(0, items) ➞ {
	  capacity: 0,
	  items: [],
	  weight: 0,
	  value: 0
	}

UI Design	● You should have an input field for the maximum weight. ● Then create a list of 20 different items (you decide the items) that can be selected and put in the knapsack ● No item can be selected when a maximum weight hasn't been imputed ● A div that shows the state of the knapsack in real time and it should be green as long as the maximum weight is not exceeded. ● There should be a done button, to show when the user is done inputting items in the knapsack and the return a div showing the state of the final knapsack

Question 3

Scale Balancing	
Context	A Scale contains two elements, the first being the two positive integer weights on a balance scale (left and right sides) and the second element being a list of available weights as positive integers. Your goal is to determine if you can balance the scale by using the least amount of weights from the list, but using at most only 2 weights
Task	For example: if a scale is ["[5, 9]", "[1, 2, 6, 7]"] then this means there is a balance scale with a weight of 5 on the left side and 9 on the right side. It is, in fact, possible to balance this scale by adding a 6 to the left side from the list of weights and adding a 2 to the right side. Both scales will now equal 11 and they are perfectly balanced. Your program should return a comma separated string of the weights that were used from the list in ascending order, so for this example, your program should return the string 2,6. Conditions ● The first element of the scale can only contain 2 weights ● It is possible to add two weights to only one side of the scale to balance it ● If it is not possible to balance the scale then your program should return "Scale Imbalanced"
UI Design	● 2 inputs to take the 2 elements of the Scale ● A button to calculate the needed weight to balance ● A div to display the result