

Objects and Classes

Subtitle

Created by: Niko Kodzhabashev

# Heroic Inventory

In the era of heroes, every hero has his own items which make him unique. Create a function which creates a **register for the heroes**, with their **names**, **level**, and **items**, if they have such. The register should accept data in a specified format, and return it presented in a specified format.

The **input** comes as array of strings. Each element holds data for a hero, in the following format:

“{heroName} / {heroLevel} / {item1}, {item2}, {item3}…”

You must store the data about every hero. The **name** is a **string**, the **level** is a **number** and the items are all **strings.**The **output** is a **JSON representation** of the data for all the heroes you’ve stored. The data must be an **array of all the heroes**. Check the examples for more info.

A screenshot of a cell phone

Description automatically generated

### **Hints**

* We need an array that will hold our hero data. That is the first thing we create.

A close up of a person

Description automatically generated

* Next, we need to loop over the whole input, and process it. Let’s do that with a simple for loop.

A picture containing bird

Description automatically generated

* Every element from the input holds data about a hero, however the **elements from the data** we need are **separated by some delimiter**, so we just split each string with that **delimiter**.
* Next, we need to take the elements from the **string array**, which is a result of the **string split**, and by [destructuring assignment syntax](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment) we assign the array properties. Don’t forget to parse the number.
* However, here we remember there is something special about the items, so read the problem definition again, you will notice that there might be a **case** where the hero **has** **no items**; in that case, using **destructuring** is ok and when there are no items, our property items will be undefined and trying to spit it will throw an error. That is why we need to perform a simple check using the [ternary operator](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Conditional_Operator).



* If **there are any items** in the **input**, the **variable** will be set to the **split version of them**. If not, it will just be set to an **empty array**.
* We have now extracted the needed data – we have stored the **input name** in a **variable**, we have parsed the **given level** to a **number**, and we have also **split** the **items** that the **hero holds** by their **delimiter**, which would result in a **string array** of elements. By definition, the **items** are **strings**, so we don’t need to process the array we’ve made anymore.
* Now what is left is to add that data into **an object** and **add** that object to the **array**.

A picture containing bird

Description automatically generated

* Lastly, we need to turn the array of objects we have made, into a JSON string, which is done by the JSON.stringify() function

A screenshot of a cell phone

Description automatically generated

1. A screenshot of a cell phone

   Description automatically generated
2. A screenshot of a social media post

   Description automatically generated
3. A screenshot of a cell phone

   Description automatically generated
4. A screenshot of a social media post

   Description automatically generated
5. A screenshot of a social media post

   Description automatically generated
6. A screenshot of a cell phone

   Description automatically generated

A screenshot of a cell phone

Description automatically generated

You’re tasked with calculating the total sum of income for a number of Towns. You will receive an array of strings representing towns and their incomes, every **even** index will be a **town** and every **odd** index will be an **income** belonging to that town. Create an object that will hold all the **towns as keys** and their **total income** (the sum of their incomes) **as values** to those keys and print it as a JSON.  
  
The **input** comes as an array of strings - each even index is the name of a town and each odd index is an income belonging to that town.

The **output** should be printed on the console - JSON representation of the object containing all towns and their total incomes.

A screenshot of a cell phone

Description automatically generated

+++++++

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated