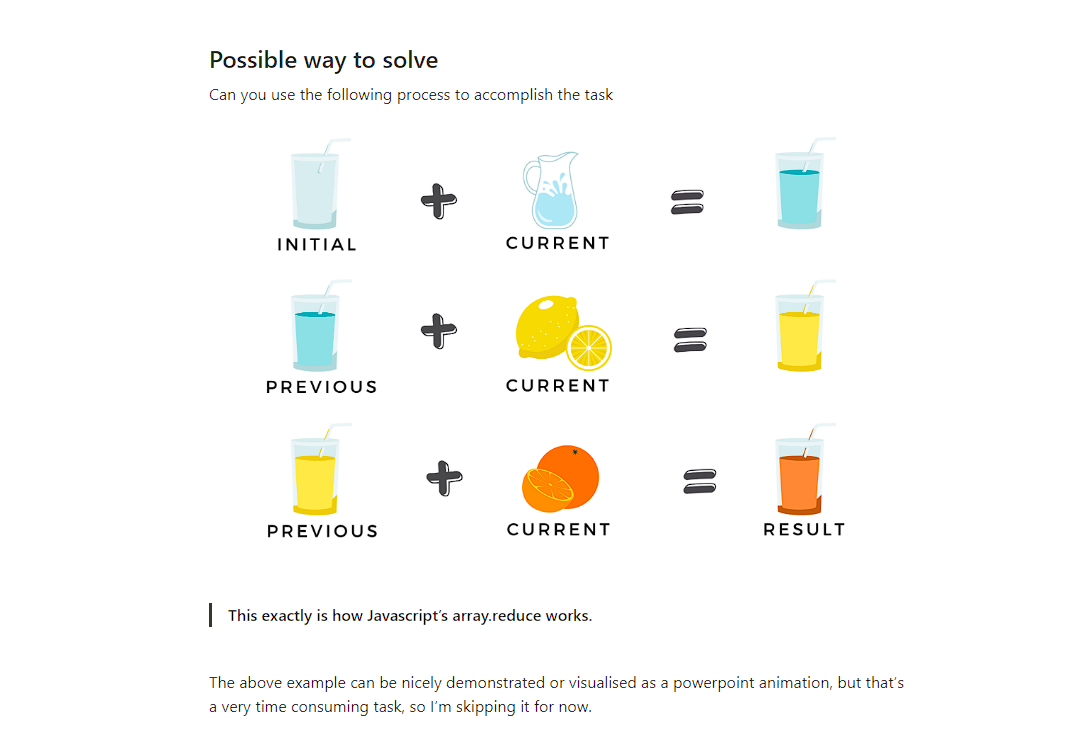
# Array.reduce()

Javascript’s most powerful and flexible array method

## Problem Statement

Imagine that you have a glass which is nice and empty **initially**





**Student Task**

Assume this array [2,4,1,7,8] of numbers, you need to use the same iterative approach as our juice example (array.reduce() method) to find the total of all the items (numbers) in the array.

Answer the following questions:

1. How many times would be the reducer-function invoked? In other words, how many iterations would happen?
2. What should be the initial value?
3. What should be returned from the first iteration?
4. What will be the previous value (accumulated value) for the second iteration?
5. What will be the current value in the second iteration?
6. What will be the previous (accumulated) value in the third iteration?

## Syntax

array.reduce(function(){}, <initialValue>)

array.reduce(function(previousValue, currentValue){

// iterates over each item in the array

// in the first iteration the initialValue is the previousValue

// what you return, becomes the previousValue for the next iteration

}, <initialValue>)

## Practical & Real-world use case

You are receiving data from backend in the following format:

let developers\_array = [

{

first: "John",

last: "Doe",

dept: "FE",

commits: 10

},

{

first: "Jane",

last: "Doe",

dept: "BE",

commits: 15

},

{

first: "James",

last: "bond",

dept: "BE",

commits: 8

}

];

But at the frontend, you need to use it in the following way:

{

BE: ["Jane", "James"]

FE: ["John"]

}

**Code-pen:**

<https://codesandbox.io/s/javascript-foundations-reduce-jexnh?file=/src/index.js>

## Student Task

Use the exact same example of the developers\_array and write reduce function that returns the total number of commits by all the developers

## Student Task [optional]

Use the exact same example of the developers\_array and write reduce function that returns the developer (first name only) with maximum number of commits. End result is expected to be in the following format: { starDeveloper: 'Jane', commits: 15 }

**Assignments: \*\*\*\***

1. Write the function getAverageAge(users) that gets an array of objects with property age and returns the average age.

The formula for the average is (age1 + age2 + ... + ageN) / N.

For instance:

let john = { name: "John", age: 25 };

let pete = { name: "Pete", age: 30 };

let mary = { name: "Mary", age: 29 };

let arr = [ john, pete, mary ];

alert( getAverageAge(arr) ); // (25 + 30 + 29) / 3 = 28

* Solution
* function getAverageAge(users) {
* return users.reduce((prev, user) => prev + user.age, 0) / users.length;
* }
* let john = { name: "John", age: 25 };
* let pete = { name: "Pete", age: 30 };
* let mary = { name: "Mary", age: 29 };
* let arr = [ john, pete, mary ];
* alert( getAverageAge(arr) ); // 28

1. Let’s say we received an array of users in the form {id:..., name:..., age:... }. Create a function groupById(arr) that creates an object from it, with id as the key, and array items as values.

For example:

let users = [

{id: 'john', name: "John Smith", age: 20},

{id: 'ann', name: "Ann Smith", age: 24},

{id: 'pete', name: "Pete Peterson", age: 31},

];

let usersById = groupById(users);

/\*

// after the call we should have:

usersById = {

john: {id: 'john', name: "John Smith", age: 20},

ann: {id: 'ann', name: "Ann Smith", age: 24},

pete: {id: 'pete', name: "Pete Peterson", age: 31},

}

\*/

Such function is really handy when working with server data.

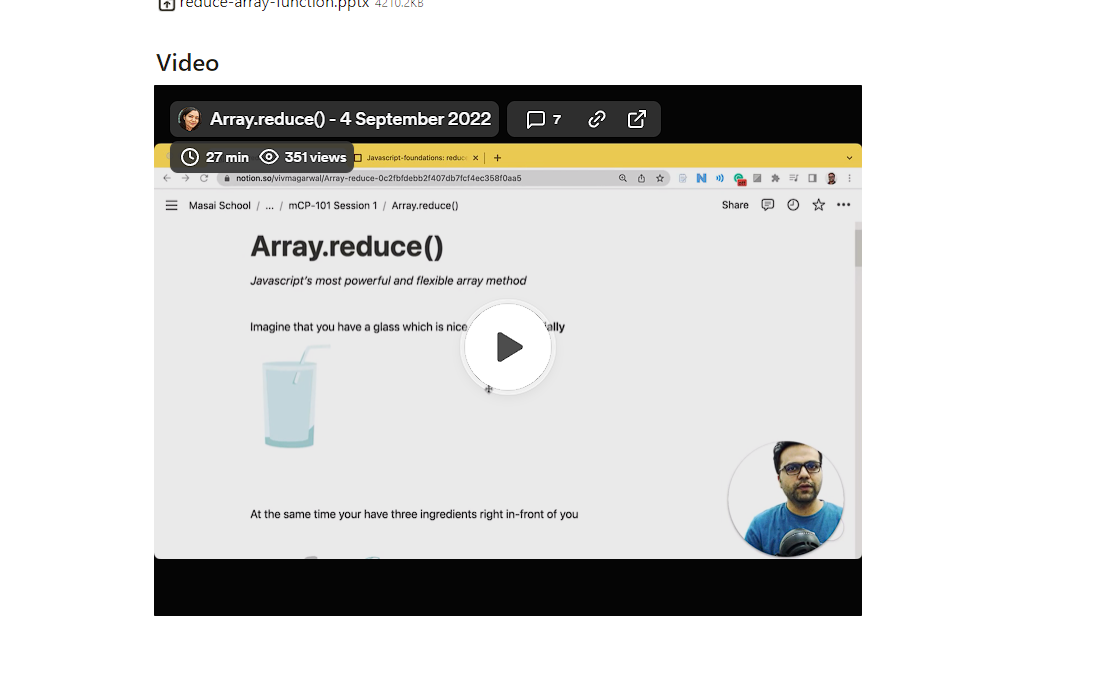
In this task we assume that id is unique. There may be no two array items with the same id.

Please use array .reduce method in the solution.

* Solution
* function groupById(array) {
* return array.reduce((obj, value) => {
* obj[value.id] = value;
* return obj;
* }, {})
* }

**Slide**

[reduce-array-function.pptx](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b55949eb-6335-49fc-b003-0ed5d5ddd054/reduce-array-function.pptx)-🡪check it inside the folder



This video link🡪https://www.loom.com/share/379056bc56a14807a0f0d20db718b201?t=4

Youtube link🡪https://youtu.be/GwT4jLt5Dbc