# Reduce()

The reduce() method executes a **reducer** function (that you provide) on each element of the array, resulting in a single output value.

The **reducer** function takes four arguments:

1. Accumulator (acc)
2. Current Value (cur)
3. Current Index (idx)
4. Source Array (src)

Your **reducer** function's returned value is assigned to the accumulator, whose value is remembered across each iteration throughout the array and ultimately becomes the final, single resulting value.

## Syntax

*arr.reduce(callback(accumulator, currentValue[, index[, array]])[, initialValue])*

Parameters

**callback**

A function to execute on each element in the array (except for the first, if no initialValue is supplied), taking four arguments:

**accumulator**

The accumulator accumulates the callback's return values. It is the accumulated value previously returned in the last invocation of the callback, or initialValue, if supplied (see below).

**currentValue**

The current element being processed in the array.

**index**Optional

The index of the current element being processed in the array. Starts from index 0 if an initialValue is provided. Otherwise, starts from index 1.

**array**Optional

The array reduce() was called upon.

**initialValue**Optional

A value to use as the first argument to the first call of the callback. If no initialValue is supplied, the first element in the array will be used and skipped. Calling reduce() on an empty array without an initialValue will throw a TypeError.

Return value

The single value that results from the reduction.

const companies = [

  { name: "Company One", category: "Finance", start: 1981, end: 2004 },

  { name: "Company Two", category: "Retail", start: 1992, end: 2008 },

  { name: "Company Three", category: "Auto", start: 1999, end: 2007 },

  { name: "Company Four", category: "Retail", start: 1989, end: 2010 },

  { name: "Company Five", category: "Technology", start: 2009, end: 2014 },

  { name: "Company Six", category: "Finance", start: 1987, end: 2010 },

  { name: "Company Seven", category: "Auto", start: 1986, end: 1996 },

  { name: "Company Eight", category: "Technology", start: 2011, end: 2016 },

  { name: "Company Nine", category: "Retail", start: 1981, end: 1989 }

];

let res = companies.reduce((total, company) => total + (company.end - company.start), 0 );

console.log(res); // 118

let ages = [20, 40, 100, 30, 50]

let res = ages.reduce((total, age)=> total+age, 0);

console.log(res);  // 240