JavaScript

Traversy Media (35:00)

**Higher Order Functions and Arrays (forEach, Map, Filter, Sort, Reduce)**

**We can combine them all to manipulate the data.**

**For Loop**:

**Syntax:**

*for (let i = 0; i < companies.length; i++) {*

*console.log(companies[i])*

*}*

f**orEach**:

It is a synchronous function. It takes a callback with one or three arguments.

Syntax:

arr.forEach(callback(currentValue [, index [, array]])[, thisArg])

***currentValue***

The current element being processed in the array.

***index***Optional

The index *currentValue* in the array.

***array***Optional

The array forEach() was called upon.

***thisArg***Optional

Value to use as this when executing *callback*.

**companies.forEach(function(company, index, companies ) {**

**console.log(company)}**

**Filter: creates a new array that pass the test**

1. **let canDrink = []**

**for (let i = 0; i < ages.length; i++) {**

**if(ages[i] >= 21) {**

**canDrink.push(ages[i])**

**)}**

1. **const canDrink = ages.filter(function(age) {**

**if(age >= 21) {**

**return true )})**

1. **ES6:**

**const canDrink = ages.filter(age => age >= 21)**

Companies that started in 1980 and 1990

const eightiesCompanies = companies.filter(company => (company.start >= 1980 && company.start < 1990))

**Map:**

Creates new arrays from the current array.

**Sort:**

The sort() method sorts the items of an array. The sort order can be either alphabetic or numeric, and either ascending (up) or descending (down). By default, the sort() method sorts the values as strings in alphabetical and ascending order.

This works well for strings ("Apple" comes before "Banana"). However, if numbers are sorted as strings, "25" is bigger than "100", because "2" is bigger than "1". Because of this, the sort() method will produce an incorrect result when sorting numbers.

You can fix this by providing a "compare function" (See "Parameter Values" below).

**Compare function (a – b)**

**.sort(a, b)**

for ascending (a – b), for descending (b -a)

const sortedCompanies = companies.sort((a, b) => (a.start > b.start ? 1 : -1))

**Reduce:**

.reduce((total, num), 0)

const ageSum = ages.reduce((total, age) => total + age, 0)