Array.prototype.forEach2 = function (callback) {

     for ( var index in this) { // use for/in instead of normal for-loop,

                            // because for/in do not scan the empty elements of an array.

          if ( this.hasOwnProperty(index) ) { // Scanning only elements of array, not prototype.

               callback(this[index], index, this);

          }

     }

     // return result;

}

var languages = [

     {

          id: 1,

          title: 'HTML, CSS',

          coin: 2

     },

     {

          id: 2,

          title: 'JS',

          coin: 0

     },

     {

          id: 3,

          title: 'ReactJS',

          coin: 0

     },

     {

          id: 4,

          title: 'NodeJS',

          coin: 0

     },

     {

          id: 5,

          title: 'Responsive',

          coin: 0

     },

     {

          id: 6,

          title: 'ReactJS',

          coin: 10

     },

]

var languages = new Array(100);

languages.push('JS', 'C++')

var isFree = languages.forEach2(function (course, index, array) {

     console.log(index, course , array);

})

// console.log(isFree)

For/in: 1) Scan only the “real” elements inside an array, not **empty** elements.

2) However, it even scans prototype-elements. That’s a reason why we need this code line:

[Array].hasOwnProperty(index) => to scan only elements inside our array.

e.g: if (this.hasOwnProperty(index)) { //code };