JavaScript Array and Objects

Methods and code snippets for item values from JavaScript arrays and object data.

Laurence Svekis https://basescripts.com/

How to Create an Array JavaScript

```
const arr = ['Laurence',4,5,5,5,6,'Svekis'];
//arr.length = 10;
const arr1 = new Array('Laurence','Svekis');

console.log(arr);
console.log(arr1);
console.log(arr[1]);
console.log(arr1[0]);
console.log(arr.length);
console.log(arr1.length);

console.log(arr[arr.length-1]);



const arr2 =
[true,10,'String',[1,2,3,4],{first:'Laurence'},null,undefined,,'last'];
console.log(arr2);
```

Nested Arrays Select Values from inner Array JavaScript

1	2	3
4	5	6
7	8	9

```
<!DOCTYPE html>
<html>
  <head><title>JavaScript</title>
    <style>
      *{
        box-sizing:border-box;
      }
      .box{
        border:1px solid #ddd;
        width:33%;
        display:inline-block;
        text-align:center;
      }
    </style>
  </head>
  <body>
    <script src='js/array.js'></script>
  </body>
</html>
```

```
const main = document.createElement('div');
```

```
document.body.prepend(main);

const arr = [[1,2,3],[4,5,6],[7,8,9]];
console.log(arr);
for(let i=0;i<arr.length;i++){
  //console.log(arr[i]);
  for(let x = 0;x<arr[i].length;x++){
     //console.log(arr[i][x]);
     const ele = document.createElement('div');
     ele.textContent = arr[i][x];
     ele.classList.add('box');
     main.append(ele);
  }
}
console.log(arr);
console.log(arr[1]);
console.log(arr[1][1]);
```

Common Array Static Methods and creating Arrays coding examples

```
const str = 'Laurence';
const arr = Array.from(str);
//console.log(arr);
const arr1 = [1,2,3,4];
const arr2 = Array.from(arr1,(ele,index)=>{
  //console.log(ele);
```

```
  return ele * 2;
});
//console.log(arr2);
const arr3 = Array.from(arr1,ele=>ele*2);
//console.log(arr3);

const lis = document.querySelectorAll('li');
console.log(lis);
const newLis = Array.from(lis,ele=>ele.textContent);
console.log(newLis);

console.log(Array.isArray(arr));
console.log(Array.isArray(str));
console.log(Array.isArray([]));
console.log(Array.isArray(3));

const arr4 = Array.of(5);
console.log(arr4);

const arr5 = Array.of(5,6,7,4,8,9);
console.log(arr5);

const str1 = '1,2,3,4,5,6';
const arr6 = Array.of(str1);
console.log(arr6);

const arr7 = str1.split(',');
console.log(arr7);
```

```
const arr8 = Array.from(str1);
console.log(arr8);

const arr9 = Array.from(str1,(val)=>{
  if(val != ',') return val;
});
console.log(arr9);

const arr10 = [...str1];
console.log(arr10);
```

```
    <ul>
      <li>1</li><li>2</li><li>3</li><li>4</li>
    </ul>
```

JavaScript Objects object literal and constructor

name:values pairs

```
const arr = [1,2,3];
console.log(typeof(arr));

const obj1 = {};
const obj2 = new Object();

obj1.first = 'Laurence';
obj2.first = 'Laurence';
obj1['last'] = 'Svekis';
```

```
obj2['last']  = 'Svekis';

console.log(obj1);
console.log(obj2);

console.log(obj1.first);
console.log(obj2.first);

const obj3 = {
  first:'Laurence',
  last:'Svekis',
  id:100,
  member:true,
  courses:['JavaScript','HTML','CSS'],
  id:5000,
  obj:{id:10,val:500}
};

console.log(obj3);
console.log(obj3.id);
console.log(obj3.courses);
console.log(obj3.courses[0]);
console.log(obj3.obj.id);

const obj4 = {
  "first" : "Laurence"
};
```

```
console.log(obj4);

const obj5 = {
  "first name" : "Laurence"
};

console.log(obj5);
console.log(obj5["first name"]);

const obj6 = {
  id1 : 1,
  id2 : 2,
  id3 : 3
};

for(let i=1;i<4;i++){
  console.log(obj6['id'+i]);
}

const obj7 = {
  first : 'Laurence',
  last : 'Svekis',
  full : function(){
     return `${this.first} ${this.last}`;
  }
}
obj7.last = 'Smith';
console.log(obj7);
```

```
console.log(obj7.full());

const obj8 = {
  first : 'Laurence',
  last : 'Svekis',
  full : ()=>{
     return `${obj8.first} ${obj8.last}`;
  }
}
console.log(obj8);
obj8.last = 'Smith';
console.log(obj8.full());
```

JavaScript Object static Methods Using Object Data

```
//'use strict'

const obj1 = {first:'Laurence'};
const obj2 = {last:'Svekis'};

const obj3 = Object.assign(obj1,obj2);
obj3.id = 100;
console.log(obj1);
console.log(obj2);
console.log(obj3);

const obj4 = obj3;
```

```
obj4.val = 1000;
console.log(obj1);

const obj5 = Object.assign({},obj1);
obj1.val = 'hello';
console.log(obj5);
console.log(obj1);

const val1 = JSON.stringify(obj1);
console.log(val1);
const obj6 = JSON.parse(val1);
console.log(obj6);
obj1.val1 = 'test';
console.log(obj6);
console.log(obj1);

const obj7 = {a:1};
const obj8 = {b:2};
const obj9 = {c:3};
const obj10 = {a:4};

const obj11 = Object.assign({},obj7,obj8,obj9,obj10);
console.log(obj11);

Object.freeze(obj11);
//obj11.a = 999;
console.log(obj11.a);
```

```
console.log(Object.isFrozen(obj11));
console.log(Object.isFrozen(obj1));

Object.seal(obj10);
console.log(Object.isSealed(obj10));
obj10.a = 'Test';
console.log(obj10);
delete obj10.a;
console.log(obj10);

const obj12 = obj10;
console.log(Object.is(obj10,obj12));

const obj13 = {a:'Test'};
console.log(Object.is(obj10,obj13));
console.log(obj13);
console.log(obj10);
```

Loop Object Values Iterate Key and Values JavaScript

```
const main = document.createElement('div');
document.body.prepend(main);
const obj1 = {first:'Laurence',last:'Svekis',id:1};
for(let i=0;i<1;i++){
  obj1[`val${i+1}`] = i;
```

```
}
console.log(obj1);

const arr1 = Object.entries(obj1);
console.log(arr1);
adder('<h1>Entries</h1><hr>');
for (const [key,value] of Object.entries(obj1)){
  const ele = adder(`${key} = ${value}`);
}
adder('<h1>Keys</h1><hr>');
const arr2 = Object.keys(obj1);
console.log(arr2);
arr2.forEach(val =>{
  const ele = adder(`${val} = ${obj1[val]}`);
})

adder('<h1>Values</h1><hr>');
const arr3 = Object.values(obj1);
console.log(arr3);
arr3.forEach((val,ind) =>{
  const ele = adder(`${ind} = ${val}`);
})
function adder(html){
  const ele = document.createElement('div');
  ele.innerHTML = html;
  return main.appendChild(ele);
}
```

```
const main = document.createElement('div');
document.body.prepend(main);



function adder(html){
  const ele = document.createElement('div');
  ele.innerHTML = html;
  return main.appendChild(ele);
}
```

How to clone and duplicate JavaScript Arrays

```
const arr = [1,2,3,4];
const arr1 = [1,2,3,4,5];
const arr2 = arr;
const arr3 = arr.concat(arr1);
const arr4 = arr.concat([5,6,7,8,9]);
const arr5 = arr1.concat(arr,arr,arr,[10,10]);
const arr6 = arr.concat();
const arr7 = arr.concat('1','2');
const arr8 = arr.concat(['1','2']);
const str = '5,6';
const arr9 = arr.concat(str );
const arr10 = str.split(',');
const arr11 = arr.concat(str.split(','));
```

```
console.log(arr);
arr.length = 10;
console.log(arr);
arr.length = 1;
console.log(arr);
arr.length = 0;
console.log(arr);
console.log(arr2);
console.clear();
console.log(arr3);
console.log(arr4);
console.log(arr5);
console.log(arr6);
console.log(arr7);
console.log(arr8);
console.log(arr9);
console.log(arr10);
console.log(arr11);
```

Iterate and Loop Array items Output Values with JavaScript

```
const main = document.createElement('div');
document.body.prepend(main);

const arr = ['Laurence','Linda','Joe','Jane'];
adder(`<h2>For Loop</h2>`);
for(let i=0;i<arr.length;i++){
```

```
 adder(`index =${i} value = ${arr[i]}`);
}
adder(`<h2>While</h2>`);
let i=0;
while(i<arr.length){
  adder(`index =${i} value = ${arr[i]}`);
 i++;
}
adder(`<h2>ForEach Method</h2>`);
arr.forEach((item,index,array)=>{
  adder(`index =${index} value =${item}`);
})
adder(`<h2>Map Method</h2>`);
const arr1 = arr.map((item,index)=>{
  adder(`index =${index} value =${item}`);
 return `${index} ${item}`;
})
adder(`<h2>Filter</h2>`);
const arr2 = arr.filter((item,index)=>{
  adder(`index =${index} value =${item}`);
 return item;
});
console.log(arr2);

function adder(html){
  const ele = document.createElement('div');
  ele.innerHTML =  html;
  return main.appendChild(ele);
```

```
}
```

JavaScript Array and Object entries Next iterator Values

```
const obj1 = {first:'Laurence',last:'Svekis',id:1};
const arr1 = ['Laurence','Svekis',100];
const adder = arr1.entries();
for(const[key,value] of Object.entries(obj1)){
  console.log(`Key ${key} : ${value}`);
}
for(const[key,value] of arr1.entries()){
  console.log(`Key ${key} : ${value}`);
}

console.log(adder);
//console.log(adder.next());
//console.log(adder.next().value);
for(let item of adder){
 console.log(item[1]);
}
const adder1 = arr1.entries();
let val = adder1.next();
while(val.value){
 console.log(val.value[1]);
 val = adder1.next();
}
```

Array Update Methods Changing Array Values

```
const main = document.createElement('div');
document.body.prepend(main);
const arr1 = [1,2,3,4];
arr1[0] = 100;
for(let i=0;i<5;i++){
  arr1[arr1.length] = i+5;
}
const holder = [];
const arr2 = arr1.slice(4,6);
console.log(arr2);
const arr3 = arr1.slice(-3);
console.log(arr3);
const arr4 = arr1.slice(4,-2);
console.log(arr4);

const arr5 = arr1.splice(4,2,'10','11','12');
console.log(arr5);

arr1.push('End');
arr1.unshift('START');
const val1 = arr1.shift();
holder.push(val1);
holder.push(arr1.pop());
```

```
arr1.forEach((value,index)=>{
  adder(`${index} = ${value}`);
})
adder('<h3>String toString()</h3>');
adder(arr1.toString());
adder('<h3>String join()</h3>');
adder(arr1.join('* *'));


adder('<h3>Removed</h3>');
holder.forEach((value,index)=>{
  adder(`${index} = ${value}`);
})

adder('<h3>Slice</h3>');
arr2.forEach((value,index)=>{
  adder(`${index} = ${value}`);
})


function adder(html){
  const ele = document.createElement('div');
  ele.innerHTML = html;
  return main.appendChild(ele);
}
//console.log(arr1);
```

Clean up Array items remove duplicates and False values

```
const arr =
['Laurence','Jack','Jane','',,,,'Sam','Laurence','Jack','Jane','',null,false,undefin
ed,0,'Sam'];
console.log(arr);

const arr1 = [... new Set(arr)];
console.log(arr1);

const arr2 = arr.filter(Boolean);
console.log(arr2);

const arr3 = [... new Set(arr.filter(Boolean))];
console.log(arr3);
```

How to Filter Array Items Filter Method with JavaScript

```
const arr = ['Svekis','Laurence','test','new',5,23,54,5,1213343];
const arr1 = arr.filter(val => val.length >= 4);
console.log(arr1);

const arr2 = arr.filter(checker);
console.log(arr2);
```

```
function checker(item,index,array){
  //console.log(item);
 /// console.log(index);
  //console.log(array);
  return index >=2;
}

const arr3 = arr.filter((item,index)=>{
  //console.log(typeof(item) );
  return typeof(item) == 'string';
});
console.log(arr3);
arr.push(null);
arr.push(false);
console.log(arr);
const arr4 = arr.filter(Boolean);
console.log(arr4);

const arr5 = arr.filter((str)=>{
  if(typeof(str)=='string'){
     const first = str[0].toUpperCase();
     return first === str[0];
  }
})

console.log(arr5);
```

Index value and checking if array Includes values

```
const arr = ['Laurence','Svekis','Svekis',100,'Svekis',1000];
const boo1 = arr.includes('Svekis');
console.log(boo1);
const boo2 = arr.includes('test');
console.log(boo2);

const ind1 = arr.indexOf('Svekis');
console.log(ind1);
const ind2 = arr.indexOf('test');
console.log(ind2);

const lind1 = arr.lastIndexOf('Svekis');
console.log(lind1);
const lind2 = arr.lastIndexOf('test');
console.log(lind2);

const find1 = arr.findIndex(call1);
console.log(find1);
const find2 = arr.findIndex(call2);
console.log(find2);

function call1(item){
  //console.log(item);
  if(item === 'Svekis' ){
     return item;
  }
```

```
}

function call2(item){
  if(typeof(item)=='number'){
      return item;
  }
}
```

How to use the Array Map Method create new array

```
const arr = ['Laurence',4,545,false,'Test'];
const arr2 = arr;
arr2.push('NEW');
console.log(arr);

const arr1 = arr.map((item,index,array)=>{
  const temp = `${index} ${item}`;
  return temp;
})
arr2.push('NEW');
console.log(arr1);

const arr3 = [3,54,62,4334,1232,444];
const arr4 = arr3.map(val => val*2);
console.log(arr4);

const arr5 = arr3.map(callback1);
```

```
console.log(arr5);

function callback1(item){
  console.log(item);
  return item*2;
}

const arr6 =
[{first:'Laurence',last:'Svekis'},{first:'John',last:'Smith'},{first:'Sam',last:'Jo
nes'}]
console.log(arr6);
const arr7 = arr6.map(({first,last})=>{
  //console.log(first,last);
  //return `${first} ${last}`;
  return {fullName:`${first} ${last}`}
})

console.log(arr7);

const arr8 = arr6.map(({first,last})=> ({full:`${first} ${last}`}));
console.log(arr8);
```

How to get Unique Values from an Array with filter Method

```
const people = ['Laurence','Linda','Jane','Jack','Jack'];
for(let i=0;i<10;i++){
```

```
  people.push('Laurence');
}
let people1 = people;
console.log(people);
const newPeople = [... new Set(people)];
console.log(newPeople);
people1.push('Mike');
console.log(people);
people1 =  [... new Set(people)];
console.log(people1);
people1.push('New');
console.log(people);
console.log(people1);
console.log(newPeople);

const people2 = people.filter((item,i)=> people.indexOf(item) === i);
console.log(people2);
```

JavaScript Array Iterator Keys and next iterator Value

```
const arr1 = ['first',1,2,3,4,5];

const keys = arr1.keys();
const vals = arr1.values();
console.log(vals);
for(const key of arr1.keys()){
```

```
  console.log(key);
}
for(const val of arr1.values()){
  console.log(val);
}

let val1 = vals.next();
console.log(val1);
while(!val1.done){
  console.log(val1.value);
  val1 = vals.next();
  console.log(val1);
}
```

How to use the Reduce Array Method count occurrences of values

$$10 + 12(1) + 32(2) + 4(3) + 5(4) + 6(5) + 55(6) = 124$$

Return unique values of an array in a new array, create an object with the count of the occurrence of a value in an array. Total all the values of an array together.

```
const arr1 = [10,12,32,4,5,6,55];
const main = document.createElement('div');
document.body.prepend(main);
let html = arr1[0];
```

```
const total = arr1.reduce((prev,cur,ind)=>{
  html += ` + ${cur}(${ind})`;
  console.log(prev,cur);
  return prev + cur;
});
console.log(total);
html += ` = ${total}`;
main.innerHTML = html;

const friends =
['Laurence','John','Jane','Jack','Laurence','John','Jane','Laurence','John','Jane'
,'Laurence'];

const obj1 = friends.reduce((total,friend)=>{
  //console.log(total,friend);
  if(friend in total){
     total[friend]++;
  }else{
     total[friend] = 1;
  }
  return total;
},{});
const arr2 = friends.reduce((arr,friend)=>{
  if(!arr.includes(friend)){
     arr.push(friend);
  }
  return arr;
},[]);
```

```
console.log(obj1);
console.log(arr2);
```

Array Sort Method Reverse sort and random sort orders in JavaScript

Start

Arr1 = 10,12,32,4,5,10006,55,10001
Friends = Laurence,John,Jane,Jack

Sort

Arr1 = 10,10001,10006,12,32,4,5,55
Friends = Jack,Jane,John,Laurence

Reverse

Arr1 = 55,5,4,32,12,10006,10001,10
Friends = Laurence,John,Jane,Jack

Sort -1

Arr2 = 10,10001,10006,12,32,4,5,55,Jack,Jane,John,Laurence
Arr2 = Laurence,John,Jane,Jack,55,5,4,32,12,10006,10001,10

Randoms

Arr2 = 10006,5,Laurence,John,Jane,10,4,32,Jack,55,10001,12
Arr2 = 5,10006,10001,12,55,32,4,Laurence,John,Jane,10,Jack

```
const main = document.createElement('div');
document.body.prepend(main);
const arr1 = [10,12,32,4,5,10006,55,'10001'];
const friends = ['Laurence','John','Jane','Jack'];
const arr2 = arr1.concat(friends);
adder('<h2>Start</h2>');
```

```
adder(`Arr1 = ${arr1.toString()}`);
adder(`Friends = ${friends.toString()}`);
adder('<h2>Sort</h2>');
arr1.sort();
friends.sort();
adder(`Arr1 = ${arr1.toString()}`);
adder(`Friends = ${friends.toString()}`);
adder('<h2>Reverse</h2>');
arr1.reverse();
friends.reverse();
adder(`Arr1 = ${arr1.toString()}`);
adder(`Friends = ${friends.toString()}`);


arr2.sort();
console.log(arr2);
adder('<h2>Sort -1</h2>');
adder(`Arr2 = ${arr2.toString()}`);
arr2.sort((a,b)=>{
  return -1;
});
adder(`Arr2 = ${arr2.toString()}`);
adder('<h2>Randoms</h2>');
for(let i=0;i<10;i++){
  arr2.sort((a,b)=>{
     return Math.random() - 0.5;
  });
  adder(`Arr2 = ${arr2.toString()}`);
```

```
}
adder('<h2>Random Item</h2>');
for(let i=0;i<10;i++){
  const ind = Math.floor(Math.random()*arr2.length);
  adder(`Random Item = ${arr2[ind]}`);
}


function adder(html){
  const ele = document.createElement('div');
  ele.innerHTML = html;
  return main.appendChild(ele);
}
```