



链滴

JS 数组对象去重排序等一些常用方法整理

作者: [gjTool](#)

原文链接: <https://ld246.com/article/1571728988300>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<a>持续更新中.....

1、数组对象去重

1. 数组去重，数组中元素为数字或者字符串

1. 单个数组去重

```
//1.
function unique(arr) {
  return [...new Set(arr)]
}
//2.
function unique(arr) {
  return Array.from(new Set(arr))
}
//3.
function unique(arr) {
  var arrs = [];
  for (var i = 0; i < arr.length; i++) {
    if (arrs.indexOf(arr[i]) == -1) {
      arrs.push(arr[i])
    }
  }
  return arrs;
}
```

2. 两个数组合并去重

```
//1.
function unique(arr1, arr2) {
  return Array.from(new Set([...arr1, ...arr2]))
}
//2.
function unique(a, b) {
  let arr = a.concat(b)
  arr = arr.sort(function (a, b) {
    return a - b;
  });
  let result = [arr[0]]
  for (let i = 1, len = arr.length; i < len; i++) {
    arr[i] !== arr[i - 1] && result.push(arr[i])
  }
  return result
}
```

2. 数组对象去重，数组中元素为对象，去除相同 key 值的对象

```

//1.
function unique(arr, keyName) {
  let obj = {};
  return arr.reduce((cur, next) => {
    obj[next[keyName]] ? "" : obj[next[keyName]] = true && cur.push(next);
    return cur;
  }, [])
}
//2.
function unique(oldArr, keyName) {
  var allArr = [];
  for (var i = 0; i < oldArr.length; i++) {
    var flag = true;
    for (var j = 0; j < allArr.length; j++) {
      if (oldArr[i][keyName] == allArr[j][keyName]) {
        flag = false;
      }
    };
    if (flag) {
      allArr.push(oldArr[i]);
    };
  };
  return allArr
}

```

2、判断各种数据类型及无效值

1. 判断是否是一个数组

```

function isArray(arr){
  return Object.prototype.toString.call(arr) === '[object Array]';
  //return arr.constructor === Array;
}
isArray([1,2,3]) //true

```

2. 判断是否是一个函数

```

function isFunction(fn) {
  return Object.prototype.toString.call(fn) === '[object Function]';
  //return fn.constructor === Function;
  //return fn instanceof Function;
  //return typeof fn === "function";
}

```

3. 判断是否是一个字符串

```

function isString(str) {
  return Object.prototype.toString.call(str) === '[object String]';
  //return str.constructor === String;
  //return typeof str === "string";
}

```

4. 判断是否是一个对象

```
function isObject(obj) {
  return Object.prototype.toString.call(obj) === '[object Object]';
  //return obj.constructor === Object;
  //return obj instanceof Object;
  //return typeof obj === "object";
}
```

5. 判断是否是一个数字

```
function isNumber(num) {
  return Object.prototype.toString.call(num) === '[object Number]';
  //return num.constructor === Number;
  //return typeof num === "number";
}
```

6. 判断是否是一个时间对象

```
function isDate(date) {
  return Object.prototype.toString.call(date) === "[object Date]";
  //return date.constructor === Date;
}
```

7. 判断是否是一个布尔值

```
function isBoolean(bool) {
  return Object.prototype.toString.call(bool) === '[object Boolean]';
  //return bool.constructor === Boolean;
  //return typeof bool === "boolean";
}
```

8. 判断是否是一个空、无效的值 (null、undefined、"、[], {}、NaN)

```
function isVoid(obj) {
  if (obj == null || obj == undefined || obj == "") {
    return true
  } else if (Object.prototype.toString.call(obj) === '[object Array]' && obj.length === 0) {
    return true
  } else if (Object.prototype.toString.call(obj) === '[object Object]' && Object.keys(obj).length === 0) {
    return true
  } else if (isNaN(obj)){
    return true
  }
  return false
}
```