

# JS常用函数（四）

## 四十一、JS将驼峰字符串转为下划线字符串

```
/**
 * JS将驼峰字符串转为下划线字符串
 * @param str
 * @returns {*}
 */
function convertCamelCase(str) {
    if (typeof (str) !== 'string') {
        console.error('[convertCamelCase] argument is not String.');
```

return str;

```
    }
    var pattern = /[A-Z]/g;
    return str.replace(pattern, function(match, index) {
        if (index !== 0) {
            return '_' + match.toLowerCase();
        } else {
            return match.toLowerCase();
        }
    });
}
```

调用方法：

```
console.log(convertCamelCase("HelloWorld")); //
hello_wolrd
```

## 四十二、JS数据求交集

代码如下：

```
/**
 * JS数据求交集
 * @param array1
 * @param array2
 * @returns {*}
 */
function intersection(array1, array2) {
    return array1.filter(function(n) {
        return array2.indexOf(n) !== -1;
    });
}
```

调用方法：

```
let arr1 = [1,2,3,4,5];
let arr2 = [3,4,5,6,7];
console.log(intersection(arr1,arr2)); // [3,4,5]
```

## 四十三、JS Base64字符串编码与解码

### JS Base64字符串编码与解码

代码如下：

```
/**
 * BASE64加密
 * @param str
 * @returns {string}
 */
function base64Encode(str) {
    return btoa(unescape(encodeURIComponent(str)));
}

/**
 * BASE64解密
 * @param str
```

```

    * @returns {string}
    */
    function base64Decode(str) {
        return decodeURIComponent(escape(atob(str)));
    }

```

调用方法:

```

console.log(base64Encode("test"));      // dGVzdA==
console.log(base64Decode("dGVzdA=="));  // test

```

## 四十四、JS Url参数转对象

### JS Url参数转对象

代码如下:

```

/**
 * JS Url参数转对象
 * @param {String} url default: window.location.href
 * @return {Object}
 */
function parseQueryString(url) {
    url = url == null ? window.location.href : url
    var search = url.substring(url.lastIndexOf('?') + 1)
    if (!search) {
        return {}
    }
    return JSON.parse('{"' +
decodeURIComponent(search).replace(/"/g,
'\\"').replace(/&/g, '&').replace(/=/g, '":"'') + '"}')
}

```

调用方法:

```

console.log(parseQueryString("http://www.baidu.com?
id=1&name=hello")); // {id: "1", name: "hello"}

```

## 四十五、JS判断是否为空

### JS判断是否为空

代码如下：

```
/**
 * JS判断是否为空
 * @param val
 * @returns {boolean}
 */
function isNull(val) {
    if (val == undefined || val == null || val == "" ||
    val == ' '
        || val == "undefined" || val == "null" || val ==
"NULL") {
        return true;
    }
    return false;
}
```

调用方法：

```
console.log(isNull(undefined)) // true
console.log(isNull(null))      // true
console.log(isNull(""))        // true
console.log(isNull(' '))       // true
console.log(isNull("undefined")) // true
console.log(isNull("null"))     // true
console.log(isNull("NULL"))     // true
```

## 四十六、JS生成指定位数的随机整数

### JS生成指定位数的随机整数（可以用于短信验证码的生成）

代码如下：

```

/**
 * JS生成指定位数的随机整数
 * @param count
 * @returns {string}
 */
function getRandomNum (count) {
  let arr = new Array
  let reNum = ""
  for (let i = 0; i < count; i++) {
    arr[i] = parseInt(Math.random() * 10)
    reNum += String(arr[i])
  }
  return reNum
}

```

调用方法：

```
console.log(getRandomNum(6));
```

## 四十七、JS校验Ip地址格式

### JS校验Ip地址格式

代码如下：

```

/**
 * JS校验IP地址格式
 * @param ip
 * @returns {boolean}
 */
function checkIp(ip) {
  var regex = /^([1-9]|[1-9]\d|1\d{2}|2[0-1]\d|22[0-3])(\.(\\d|[1-9]\\d|1\d{2}|2[0-4]\\d|25[0-5])){3}$/
  var b = regex.test(ip)
  return b
}

```

调用方法：

```
console.log(checkIp("127.0.0.1"));    // true
console.log(checkIp("0.0.0.1"));      // false
```

## 四十八、JS根据文件名获取文件格式

### JS根据文件名获取文件格式

代码如下：

```
/**
 * JS根据文件名获取文件格式
 * @param str
 * @returns {string}
 */
function getFileTypeByFileName(str){
    return
    str.substr(str.lastIndexOf(".") + 1).toLowerCase();
}
```

调用方法：

```
console.log(getFileTypeByFileName("index.html"));
//html
console.log(getFileTypeByFileName("index.js"));    //js
```

## 四十九、JS查找某一字符串出现的次数

### JS在字符串中查找某一字符串出现的次数

代码如下：

```

/**
 * JS在字符串中查找某一字符串出现的次数
 * @param str
 * @param strSplit
 * @returns {number}
 */
function countStr (str,strSplit){
    return str.split(strSplit).length-1
}

```

调用方法：

```

let
strTest='sad44654blog5a1sd67as9dablog4s5d16zxc4sdweasjkblo
gwqepaskdkblogahseiuadbhjcibloguyeaajzxkcabloguyiwezxc967'
countStr(strTest,"blog");    // 6

```

## 五十、JS适配rem

### JS适配rem

代码如下：

```

/**
 * JS适配rem
 */
function getFontSize () {
    var doc = document, win = window
    var docEl = doc.documentElement,
        resizeEvt = 'orientationchange' in window ?
        'orientationchange' : 'resize',
        reCalc = function () {
            var clientWidth = docEl.clientWidth
            if (!clientWidth) return
            //如果屏幕大于750 (750是根据效果图设置的) , 就设置
            clientWidth=750, 防止font-size会超过100px
            if (clientWidth > 750) { clientWidth = 750 }
            //设置根元素font-size大小
            docEl.style.fontSize = 100 * (clientWidth / 750) +
            'px'
        }
}

```

```

    }
    //屏幕大小改变, 或者横竖屏切换时, 触发函数
    win.addEventListener(resizeEvt, reCalc, false)
    //文档加载完成时, 触发函数
    doc.addEventListener('DOMContentLoaded', reCalc, false)
  }

```

调用方法:

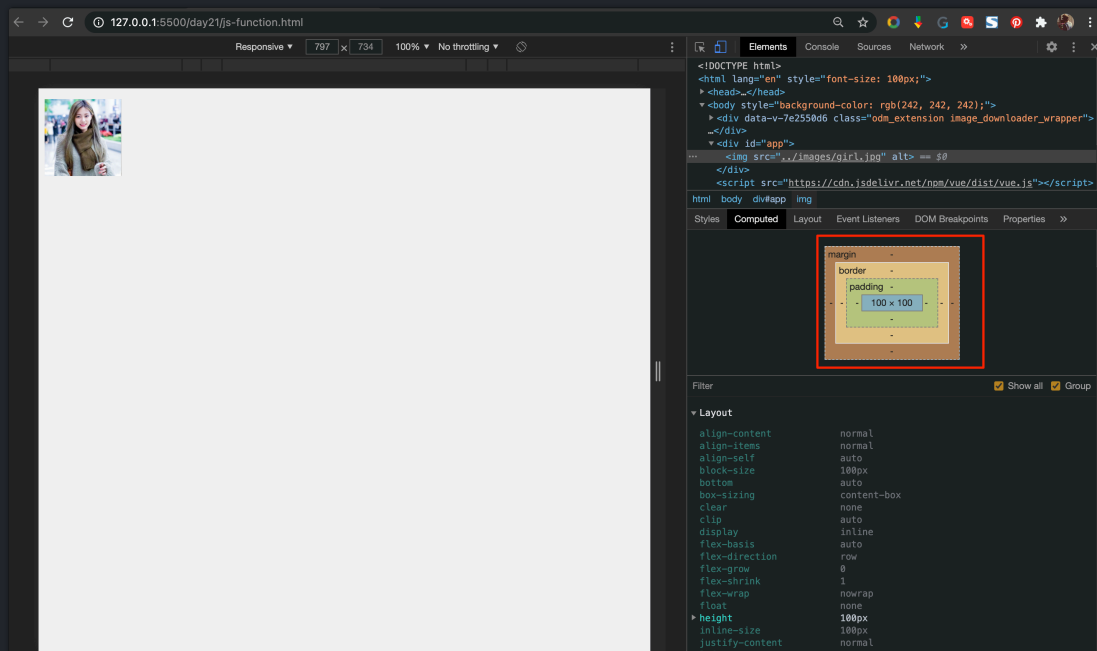
```
getFontSize();
```

```

<style>
  /*使用方式很简单, 有张图片。宽高都是100px;*/
  img{
    width:1rem;
    height:1rem;
  }
  /*这样的设置, 比如在屏幕宽度大于等于750px设备上, 1rem=100px;
  图片显示就是宽高都是100px*/
  /*比如在iphone6(屏幕宽度: 375)上, 375/750*100=50px;就是
  1rem=50px;图片显示就是宽高都是50px;*/
</style>

```

## PC效果





# iPhone6效果

