

# ban.js

---

## 1.ban()

作用： 禁止鼠标右键、复制、选择

# dataType.js

---

## 1. getData Type(value)

作用： 获取数据类型

参数： **value** => 需要判断数据类型的值

返回值： **Number、String、Object、Array**

## 2. isArray(arr)

作用： 判断数据是否为数组

参数： **arr** => 需要判断的值

返回值： **true、false**

## 3. isObj(obj)

作用： 判断数据是否为对象

参数： **obj** => 需要判断的值

返回值： **true、false**

## 4. isFunction(fn)

作用： 判断数据是否为函数

参数： **fn** => 需要判断的值

返回值：true、false

## date.js

---

### 1.dateFormat(date, fmt)

作用：格式化时间

参数：date => 需要格式化的日期 fmt => 转换的格式

返回值：fmt 格式的 date

备注：date 可以为字符串、时间戳、"，为 " 时默认当前时间。fmt 为字符串，不传时按照 'yyyy-MM-dd HH:mm:ss' 格式转换

### 2. dayEnd(date)

作用：获取日期当天结束时间的时间对象，结束时间：yyyy-MM-dd 23:59:59

参数：date => 日期：可为 String 或时间戳

返回值：date 的结束时间的时间对象

## deepCopy.js

---

### 1.deepCopy(obj)

作用：将数据进行深拷贝

参数：obj => 需要拷贝的数据

返回值：深拷贝之后的 obj

## float.js

---

### 1.floatMultiply(arg1, arg2)

作用：解决 js 中浮点类型的数字相乘精度丢失

参数: `arg1`、`arg2` => 进行乘法运算的两个数字

返回值: `arg1 * arg2` 的值

## 2. `floatSub(arg1, arg2)`

作用: 解决 js 中浮点类型的数字相减精度丢失

参数: `arg1` => 减数 `arg2` => 被减数

返回值: `arg1 - arg2` 的值

## 3. `floatAdd(arg1, arg2)`

作用: 解决 js 中浮点类型的数字相加精度丢失

参数: `arg1`、`arg2` => 进行加法运算的两个数字

返回值: `arg1 + arg2` 的值

## 4. `floatDivide(arg1, arg2)`

作用: 解决 js 中浮点类型的数字相除精度丢失

参数: `arg1` => 除数 `arg2` => 被除数

返回值: `arg1 / arg2` 的值

# frequencyHandle.js

---

## 1. `debounce(fn, time)`

作用: 防止 js 高频触发事件中函数连续执行问题, 函数从事件开始触发到结束触发中不会执行, 事件结束之后 `time` 毫秒后开始执行 函数

参数: `fn` => 事件中需要执行的函数 `time` => 从事件结束到开始执行函数的时间 单位: ms

## 2. `throttle(fn, time)`

作用：防止 js 高频触发事件中函数连续执行的问题，函数在事件开始触发到结束触发过程中每间隔 time 毫秒执行一次

参数：fn => 事件中需要执行的函数 time => 事件中函数执行的间隔时间 单位：ms

## fullScreen.js

---

### 1. toFullScreen(str)

作用：开启全屏模式

参数：str => 不支持全屏时弹出框中的提示语

### 2. exitFullScreen(str)

作用：退出全屏模式

参数：str => 退出全屏失败时弹出框中的提示语

## getExplorerInfo.js

---

### 1.getExplorerInfo()

作用：获取浏览器信息

返回值： Object, Object.type : 当前浏览器类型, Object.version: 当前浏览器版本号

例：

```
{
  type: 'Chrome',
  version: 74
}
```

## reg.js

---

作用：按照规则匹配的正则表达式，详情见文内注释

# unique.js

---

## 1. unique(arr)

作用: 数组去重

参数: **arr** => 需要去重的数组

返回值: 去重后的新数组, 不会改变原数组

# treeNode.js

---

## 1. getTreeNode(tree, key, value)

作用: 获取某个树对象下相应的节点

参数: **tree** => 类型: **Array** 树对象的类数组, **key**: 键, **value**: 值

返回值: 符合条件的树节点

例:

html:

```
<div class="div1"></div>
<div class="div2"></div>
<div class="div3"></div>
```

js:

```
let divs = document.getElementsByTagName('div')
getTreeNode(divs, 'className', 'div1')
```

results:

```
[div.div1]
```