

# 键盘事件

## 1. 键盘事件的种类

键盘事件由用户击打键盘触发，主要有 `keydown`、`keypress`、`keyup` 三个事件，它们都继承了 `KeyboardEvent` 接口。

- `keydown`：按下键盘时触发。
- `keypress`：按下有值的键时触发，即按下 `Ctrl`、`Alt`、`Shift`、`Meta` 这样无值的键，这个事件不会触发。对于有值的键，按下时先触发 `keydown` 事件，再触发这个事件。
- `keyup`：松开键盘时触发该事件。

如果用户一直按键不松开，就会连续触发键盘事件，触发的顺序如下。

1. `keydown`
2. `keypress`
3. `keydown`
4. `keypress ...`（重复以上过程）
5. `keyup`

## 2. KeyboardEvent 接口概述

`KeyboardEvent` 接口用来描述用户与键盘的互动。这个接口继承了 `Event` 接口，并且定义了自己的实例属性和实例方法。

浏览器原生提供 `KeyboardEvent` 构造函数，用来新建键盘事件的实例。

```
new KeyboardEvent(type, options)
```

`KeyboardEvent` 构造函数接受两个参数。第一个参数是字符串，表示事件类型；第二个参数是一个事件配置对象，该参数可选。除了 `Event` 接口提供的属性，还可以配置以下字段，它们都是可选。

- `key`：字符串，当前按下的键，默认为空字符串。
- `code`：字符串，表示当前按下的键的字符串形式，默认为空字符串。
- `location`：整数，当前按下的键的位置，默认为 0。
- `ctrlKey`：布尔值，是否按下 `Ctrl` 键，默认为 `false`。
- `shiftKey`：布尔值，是否按下 `Shift` 键，默认为 `false`。
- `altKey`：布尔值，是否按下 `Alt` 键，默认为 `false`。
- `metaKey`：布尔值，是否按下 `Meta` 键，默认为 `false`。
- `repeat`：布尔值，是否重复按键，默认为 `false`。

## 3. KeyboardEvent 的实例属性

3.1. `KeyboardEvent.altKey`, `KeyboardEvent.ctrlKey`, `KeyboardEvent.metaKey`, `KeyboardEvent.shiftKey`

以下属性都是只读属性，返回一个布尔值，表示是否按下对应的键。

- `KeyboardEvent.altKey`: 是否按下 `Alt` 键 (Mac 系统是 `option` 键)
- `KeyboardEvent.ctrlKey`: 是否按下 `Ctrl` 键 (Mac 系统是 `control` 键)
- `KeyboardEvent.metaKey`: 是否按下 `meta` 键 (Mac 系统是 `command` 键, Windows 系统是 `windows` 键)
- `KeyboardEvent.shiftKey`: 是否按下 `Shift` 键

下面是一个示例。

```
function showChar(e) {  
  console.log('ALT: ' + e.altKey);  
  console.log('CTRL: ' + e.ctrlKey);  
  console.log('Meta: ' + e.metaKey);  
  console.log('Shift: ' + e.shiftKey);  
}  
  
document.body.addEventListener('keydown', showChar, false);
```

### 3.2. KeyboardEvent.code

`KeyboardEvent.code` 属性返回一个字符串，表示当前按下的键的字符串形式。该属性只读。

下面是一些常用键的字符串形式，其他键请查文档。

- 数字键0 - 9: 返回`digit0` - `digit9`
- 字母键A - z: 返回`KeyA` - `KeyZ`
- 功能键F1 - F12: 返回 `F1` - `F12`
- 方向键: 返回`ArrowDown`、`ArrowUp`、`ArrowLeft`、`ArrowRight`
- `Alt` 键: 返回`AltLeft`或`AltRight`
- `Shift` 键: 返回`ShiftLeft`或`ShiftRight`
- `Ctrl` 键: 返回`ControlLeft`或`ControlRight`

### 3.3. KeyboardEvent.key

`KeyboardEvent.key` 属性返回一个字符串，表示按下的键名。该属性只读。

例如：

```
// 按下 1 键  
KeyboardEvent.code // Digit1  
KeyboardEvent.key  // 1  
  
// 按下 shift 键, 按键 z 左边的键  
KeyboardEvent.code // ShiftLeft  
KeyboardEvent.key  // Shift  
  
// 按下 shift 键, 按键 / 右边的键  
KeyboardEvent.code // ShiftRight
```

```
KeyboardEvent.key // Shift

// 按下逗号键, 按键 M 右边的键
KeyboardEvent.code // Comma
KeyboardEvent.key // ,

// code 会返回按键的类型, key 只返回具体的键名
```

如果按下的键代表可打印字符, 则返回这个字符, 比如数字、字母。

如果按下的键代表不可打印的特殊字符, 则返回预定义的键值, 比如 `Backspace`, `Tab`, `Enter`, `Shift`, `Control`, `Alt`, `CapsLock`, `Esc`, `Spacebar`, `PageUp`, `PageDown`, `End`, `Home`, `Left`, `Right`, `Up`, `Down`, `PrintScreen`, `Insert`, `Del`, `Win`, `F1~F12`, `NumLock`, `Scroll` 等。

如果同时按下一个控制键和一个符号键, 则返回符号键的键名。比如, 按下 `Ctrl + a`, 则返回 `a`; 按下 `Shift + a`, 则返回大写的 `A`。

如果无法识别键名, 返回字符串 `Unidentified`。

### 3.4. KeyboardEvent.location

`KeyboardEvent.location` 属性返回一个整数, 表示按下的键处在键盘的哪一个区域。它可能取以下值。

- `0`: 处在键盘的主区域, 或者无法判断处于哪一个区域。
- `1`: 处在键盘的左侧, 只适用那些有两个位置的键 (比如 `Ctrl` 和 `Shift` 键)。
- `2`: 处在键盘的右侧, 只适用那些有两个位置的键 (比如 `Ctrl` 和 `Shift` 键)。
- `3`: 处在数字小键盘。

### 3.5. KeyboardEvent.repeat

`KeyboardEvent.repeat` 返回一个布尔值, 代表该键是否被按着不放, 以便判断是否重复这个键, 即浏览器会持续触发 `keydown` 和 `keypress` 事件, 直到用户松开手为止。

## 4. KeyboardEvent 的实例方法

### 4.1. KeyboardEvent.getModifierState()

`KeyboardEvent.getModifierState()` 方法返回一个布尔值, 表示是否按下或激活指定的功能键。它的常用参数如下。

- `Alt`: `Alt` 键
- `CapsLock`: 大写锁定键
- `Control`: `Ctrl` 键
- `Meta`: `Meta` 键
- `NumLock`: 数字键盘开关键
- `Shift`: `Shift` 键

```
if (
  event.getModifierState('Control') +
  event.getModifierState('Alt') +
```

```
    event.getModifierState('Meta') > 1  
  ) {  
    return;  
  }
```

上面代码表示，只要 **Control**、**Alt**、**Meta** 里面，同时按下任意两个或两个以上的键就返回。