

javascript 按键事件

作者: [zhuangyan](#)

原文链接: <https://ld246.com/article/1376028585340>

来源网站: [链滴](#)

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

<p>有三种按键事件类型：keydown,keypress,keyup。一个典型的按键会产生所有的这三种事件，次是keydown,keypress,keyup。
如果一个键被按下并自动重复，则可能有多个keypress事件
不能打印的功能键 会产生keydown，keyup事件。在某些浏览器中也可能产生keypress事件然而在IE中，只有按键有一个ASCII码的时候才会发生keypress事件。如方向键就不会触发keypress事件，但是会触发keydown事件。
如何获取按键的字符码？</p>

IE中只有keyCode属性，并且它的解释取决于事件的类型。对于keydown事件，keyCode是一个虚拟按键码；对于keypress事件，keyCode是一个字符码。

firefox中定义了两个属性keyCode和charCode。keyCode存储了一个按键的较低层次的虚拟按键码，并且和keydown事件一起发送。charCode存储了按键所产生的可打印字符的编码，并且和keypress事件一起发送。如果在firefox按下功能键，charCode为0，keyCode包含了虚拟按键码。在firefox中，可以通过event.which获取keyCode的值。

<p>字符码可使用静态函数String.fromCharCode(value)转换为字符。</p>

<p>jquery中对按键事件进行了封装，可统计使用event.which获取字符码。
jquery中which但可以判断键盘的键值，也可以判断鼠标的键值。
可以参考jquery源码的event.js部分</p>

```
keyHooks: {
  props: &quot;char charCode key keyCode&quot;.split(&quot; &quot;),
  filter: function( event, original ) {
    // Add which for key events
    if ( event.which == null ) {
      event.which = original.charCode != null ? original.charCode : original.keyCode;
    }
    return event;
  },
  mouseHooks: {
    props: &quot;button buttons clientX clientY offsetX offsetY pageX pageY screenX screenY toElement&quot;.split(&quot; &quot;),
    filter: function( event, original ) {
      var eventDoc, doc, body,
          button = original.button;
      // Calculate pageX/Y if missing and clientX/Y available
      if ( event.pageX == null & & original.clientX != null ) {
        eventDoc = event.target.ownerDocument || document;
        doc = eventDoc.documentElement;
        body = eventDoc.body;
        event.pageX = original.clientX + ( doc & & doc.scrollLeft || body & & body.scrollLeft || 0 ) - ( doc & & doc.clientLeft || body & & body.clientLeft || 0 );
        event.pageY = original.clientY + ( doc & & doc.scrollTop || body & & body.scrollTop || 0 ) - ( doc & & doc.clientTop || body & & body.clientTop || 0 );
      }
      // Add which for click: 1 === left; 2 === middle; 3 === right
      // Note: button is not normalized, so don't use it
      if ( !event.which & & button !== undefined ) {
        event.which = ( button & 1 ? 1 : ( button & 2 ? 3 : ( button & 4 ? 2 : 0 ) ) );
      }
      return event;
    }
  },</pre>
```

<p>参考： <http://www.cnblogs.com/sniper007/archive/2012/11/13/2767637.html></p>