Hook插件

### afterChange (changes: Array, source: String)：1个或多个单元格的值被改变后调用

    changes：是一个2维数组包含row，prop，oldVal，newVal4个属性。

    source：其值为一个字符串，值可以为：alter，empty，populateFromArray，loadData，autofill，paste

### beforeChange (changes: Array, source: String)：开始改变单元格前被调用

    changes：是一个2维数组，包括[row，prop，oldVal，newVal]这4个公共属性列

    source是被改变的资源的名称

### afterCellMetaReset ()：重置单元格后调用

### afterColumnMove (oldIndex: Number, newIndex: Number)：列顺序被移动后触发

### afterRowMove (oldIndex: Number, newIndex: Number)：行被移动后调用

### afterRowResize (col: Number, size: Number)：行高改变后调用

### afterRemoveCol (index: Number, amount: Number)：当一列或多列被移动后调用

    其中，index为开始移动的列的索引，amount为移动的列的总数量

### afterRemoveRow (index: Number, amount: Number)：当一行或多行被移动后调用

   其中，index为被移动的行的索引，amount为被移动的行的总数量

### beforeRemoveCol (index: Number, amount: Number)：一列或多列被移动前调用

### beforeRemoveRow (index: Number, amount: Number)：一行或多行被移动前被调用

### afterColumnSort (column: Number, order: Boolean)：列排序后调用

### beforeColumnSort (column: Number, order: Boolean)：列排序前被调用

    order：值为true时为升序，false时为降序

### afterCreateCol (index: Number, amount: Number)：添加行后被调用

    index：新列的索引

    amount：新列的数目

### afterCreateRow (index: Number, amount: Number)：添加行后被调用

    index:新行的索引

    amount：新行的数目

### afterGetCellMeta (row: Number, col: Number, cellProperties: Object)：获取单元格的配置信息后被调用

### beforeGetCellMeta (row: Number, col: Number, cellProperties: Object)：获取单元格属性前被调用

### afterSetCellMeta(row: Number, col: Number, key: String, value: \*)：单元格样式被改变后调用

其中，cellProperties是一个单元格的渲染对象，key是改变样式的方式，例如合并单元格（merge），水平对齐（align）等。

### afterGetColHeader (col: Number, TH: DOM Node)：获取列头信息后被调用，TH是行头节点

### afterGetColWidth (col: Number, response: Object)：获取列宽后被调用

### afterColumnResize (col: Number, size: Number)：列宽度被手动修改后调用

### afterCopyLimit (selectedRowsCount: Number, selectedColsCount: Number,copyRowsLimit: Number, copyColsLimit: Number)

    当 copyRowsLimit 或者 copyColumnsLimit实现时被调用

### afterDestroy ()：销毁Handsontable实例后被调用

### afterInit ()：Handsontable实例被初始化后调用

### beforeInit ()：Handsontable实例被初始化前调用

### beforeInitWalkontable()：Walkontable实例被初始化前调用

### afterLoadData ()：新的数据被加载到数据资源后被调用

### afterOnCellCornerMouseDown (event)：鼠标点击单元格边角后被调用

### afterOnCellMouseDown (event: Object, coords: Object, TD: Object)：点击单元格或行头/列头后被调用

   注意：点击行头或列头后索引的坐标为负数。例如点击列头单元格(0,0)，则调用后的坐标为(0,-1)。

### afterOnCellMouseOver (event: Object, coords: Object, TD: Object)：鼠标停悬在单元格或行头/列头后调用

   注意：点击行头或列头后索引的坐标为负数。例如点击行头单元格(0,0)，则调用后的坐标为(0,-1)。

### afterRender (isForced: Boolean)：渲染表格后被调用

     isForced：当其值为true表示是通过改变配置或数据引起的渲染，当值为false时表示通过滚动或移动、选中引起的渲染

### beforeRender (isForced: Boolean)：渲染前被调用

### afterRenderer (TD: Object, row: Number, col: Number, prop: String, value: String, cellProperties: Object)：手动渲染后调用

### beforeChangeRender ()：渲染被改变前调用

### afterDeselect ()：当前单元格被取消选中时调用

### afterSelection (r: Number, c: Number, r2: Number, c2: Number)：当一个或多个单元格被选中后调用

   其中，r：选中的单元格起始行，r2：选中单元格的终止行

             c：选中的单元格的起始列，c2：选中的单元格的终止列

### afterSelectionByProp (r: Number, p: String, r2: Number, p2: String)：通过属性名选中单元格后调用

### afterSelectionEnd (r: Number, c: Number, r2: Number, c2: Number)：选中单元格鼠标抬起后调用

### afterSelectionEndByProp (r: Number, p: String, r2: Number, p2: String)：通过属性选中单元格鼠标抬起后调用

### afterUpdateSettings ()：配置参数配修改后调用

### afterValidate (isValid: Boolean, value: Mixed, row: Number, prop: String,source: String)

   当有验证器的时候调用验证器时被调用，验证结果作为第一个参数。

### beforeValidate (value: Mixed, row: Number, prop: String, source: String)：验证器被调用前调用该事件

### beforeAutofill (start: Object, end: Object, data: Array)：开始自动填充前调动

   start：是一个第一个填充的单元格对象，例如:{row:4,col:3}

   end：是最后一个填充的单元格对象，例如：{row:7,col:5}

   data:是一个2维数组

### beforeKeyDown (event: Object)：按键按下前被调用

### beforeSet (var: Object)：单个配置值被设置前调用

### beforeSetRangeEnd(coords: Array)：设置范围结束前被调用

### coords：是范围坐标

### modifyCol(col: Number)：列被修改时调用

### modifyRow( row: Number)：行被修改时调用

### modifyColWidth (width: Number, col: Number)：列宽被修改时调用

### modifyRowHeight (height: Number, row: Number)：行高被修改时调用

UndoRedo插件

### clear():清楚撤销历史记录

### isRedoAvailable():判断取消撤销是否可用

### isUndoAvailable():判断撤销操作是否可用

### undo():撤销最后的编辑操作

### redo():取消撤销操作