

# jsonEditor manual

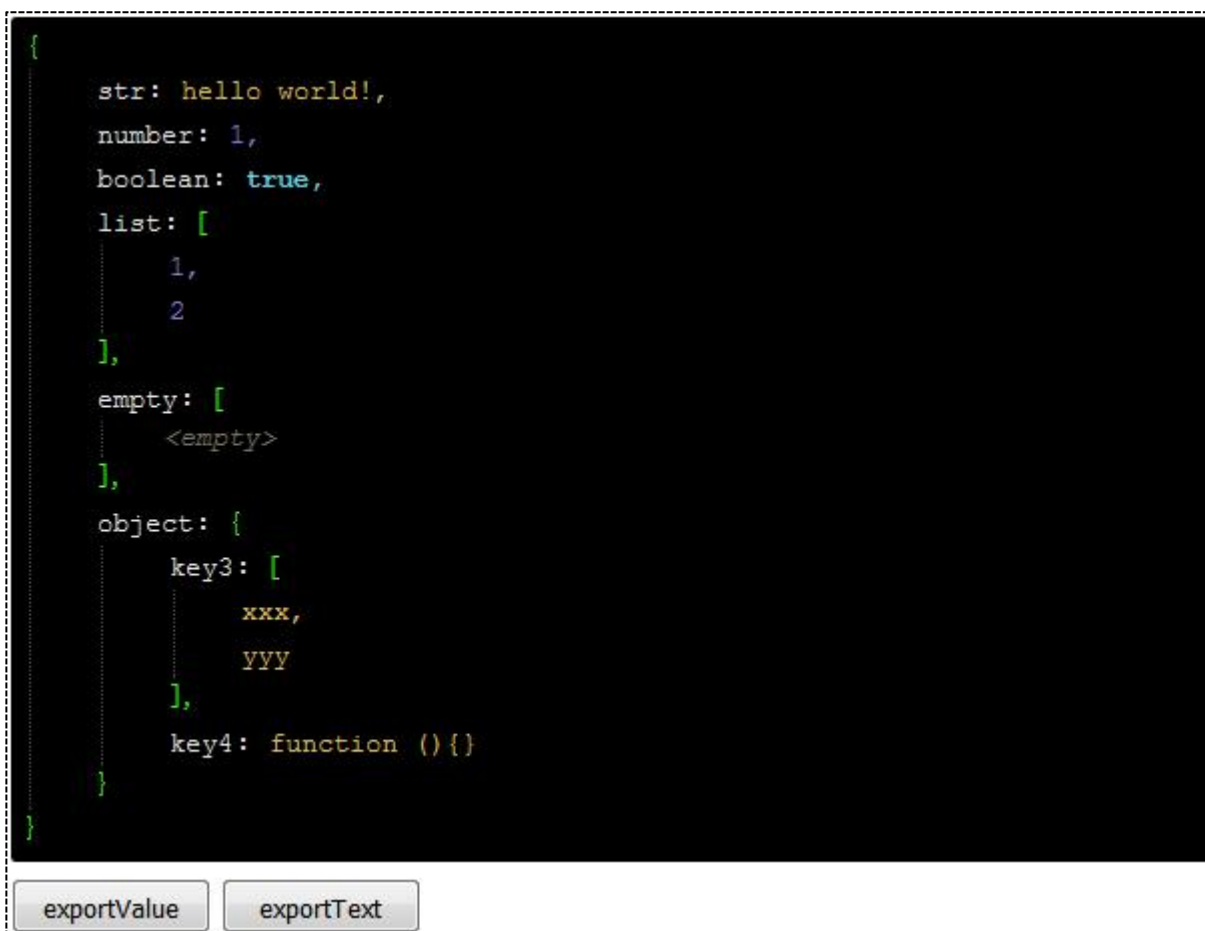
version:	1.0.0
author:	oldfour
data:	2019-08-16

## Introduction

一个使用快捷键进行操作的 JSON 编辑器，使用快捷键让编辑工作更加方便、快捷。  
可自由导入任何数据类型的值到编辑器中进行编辑，编辑后可导出对象或 JSON 字符串。

A JSON editor that uses shortcut keys to make editing work easier and faster.

You can import values of any data type into the editor for editing, and you can export objects or JSON strings after editing.



## Be applicable

- Chrome
- Firefox

- Safari
- Opera
- IE >= 9

## Use

### 1. import "json-editor.js"

```
<script type="text/javascript" src="json-editor.js"></script>
```

### 2. instance "jsonEditor"

```
window.onload = function () {  
    var editor = new jsonEditor(  
        "jsonEditorBox",  
        {  
            'str': 'hello world!',  
            'number': 1,  
            'boolean': true,  
            'list': [1, 2, 3, 4],  
            'empty': [],  
            'object': {  
                'key1': null,  
                'key2': false,  
                'key3': ['xxx', 'yyy', 'zzzz'],  
                'key4': function (){}  
            }  
        },  
        {  
            hotKey: 'J',  
            width: 680,  
            height: 480  
        }  
    );  
}
```

# API

## 1. (new)

创建一个 JSON 编辑器。

```
new jsonEditor(container, data, attributes)
```

name	description	value type
container	目标节点（在该目标节点内创建编辑器）	可以是一个存在的 Element 对象，也可以是 Element 的 ID（string 类型）
data	导入到编辑器中进行编辑的值	可以是任何类型的数据
attributes	编辑器的属性	对象类型，包括： <b>hotKey:</b> 激活编辑器的热键（可任意字母） <b>width:</b> 编辑器宽度，可取值"auto"、"inherit"、"?px"（"?px"可省略单位，直接使用数字） <b>height:</b> 编辑器高度（取值同上） <b>minWidth:</b> 编辑器最小宽度（取值同上） <b>minHeight:</b> 编辑器最小高度（取值同上） <b>overflow:</b> 编辑器内内容溢出时的显示方式，可取值："auto"、"hidden"、"visible"、"scroll"、"inherit"（默认"auto"）

## 2. importing

导入数据到编辑器中。

```
editor = new jsonEditor(container, null, {})  
editor.importing(data)
```

name	description	value type
data	导入到编辑器中进行编辑的值	可以是任何类型的数据

## 3. value

获取编辑器的 value（返回结果可能是 null、boolean、number、string、object）。

```
editor = new jsonEditor(container, {k1:'xxx', 'yyy', 'zzzz'}, {})  
editor.value()
```

## 4. text

获取编辑器的 value 的 JSON 字符串。

```
editor = new jsonEditor(container, {k1:'xxx', 'yyy', 'zzzz'}, {})  
editor.text()
```

## Noun

### 1. Editable node

“可编辑节点”指所有元素的 value 节点、object 类型元素的所有 key 节点，以及 object 和 Array 类型元素的子元素节点。



如上图所示，黄色边框的元素都是可编辑元素

## Hot key

### 1. 激活编辑器

创建编辑器时，如果指定了快捷键，那么可以使用“Shift+Alt+快捷键”激活编辑器；

### 2. 切换到相邻的 Key 和 Value 节点

- 使用 **Tab** 切换到下一个最近的 Key 或者 Value 节点；

```
{
  str: hello world!,
}
```

按下 Tab 键



```
{
  str: hello world!,
}
```

切换到下一个可编辑节点

- 使用 **Shift + Tab** 切换到上一个最近的 Key 或者 Value 节点;

```
{
  str: hello world!,
}
```

按下 Shift + Tab 键



```
{
  str: hello world!,
}
```

切换到上一个可编辑节点

- object 或 Array 类型元素下的子元素，按下 **↑**、**↓** 切换到相邻的兄弟元素;

```
object: {
  key1: null,
  key2: false,
  key3: [

```

按下 ↓ 键



```
object: {
  key1: null,
  key2: false,
  key3: [

```

切换到相邻的兄弟元素

- object 或 Array 类型元素下的子元素，按下 **PgUp**、**PgDn** 切换到第一个或最后一个兄弟元素;

```
object: {
  key1: null,
  key2: false,
  key3: [
    xxx,
    yyy
  ],
  key4: function () {}
}
```

按下 PgDn 键



```
object: {
  key1: null,
  key2: false,
  key3: [
    xxx,
    yyy
  ],
  key4: function () {}
}
```

切换到最后一个兄弟元素

- object 或 Array 类型元素下的子的元素，按下 **Ctrl + ↑**、**Ctrl + ↓**，将当前元素向上、向下移动一位;

```
object: {
  key1: null,
  key2: false,
  key3: [
    xxx,
    yyy
  ],
  key4: function () {}
}
```

按下 Ctrl + ↑ 键



```
object: {
  key1: null,
  key2: false,
  key4: function () {},
  key3: [
    xxx,
    yyy
  ]
}
```

将当前元素向上移动一位

### 3. 切换到上一级可编辑节点

- object 或 Array 类型元素的子元素，按下 **ESC** 切换到整个 object 或 Array;

```
object: {
  key1: null,
  key2: false,
  key4: function () {},
  key3: [
    xxx,
    yyy
  ]
}
```

在元素节点上按下 ESC 键



```
object: {
  key1: null,
  key2: false,
  key4: function () {},
  key3: [
    xxx,
    yyy
  ]
}
```

切换到整个 object 或 Array

- object 类型元素下的子元素 Key 或 Value 节点，使用 **ESC**，可切换到当前所在 object 的子元素;

```
object: {
  key1: null,
  key2: false,
  key4: function () {},
  key3: [
    xxx,
    yyy
  ]
}
```

在 key 节点上按下 ESC 键



```
object: {
  key1: null,
  key2: false,
  key4: function () {},
  key3: [
    xxx,
    yyy
  ]
}
```

切换到当前所在 object 的子元素

### 4. 进入到下一级可编辑节点

- object 类型元素下的的 Key 节点，使用 **Enter** 可离开当前编辑框，切换到 Value (与

按下 Tab 键一样的效果)；

```
{
  str: hello world!,
}
```

在 key 节点上按下 Enter 键



```
{
  str: hello world!,
}
```

切换到 value 节点

```
object: {
  key1: null,
  key2: false,
  key4: function () {},
  key3: [
    xxx,
    yyy
  ]
}
```

在 key 节点上按下 Enter 键



```
object: {
  key1: null,
  key2: false,
  key4: function () {},
  key3: [
    xxx,
    yyy
  ]
}
```

切换到 value 节

- object 或 Array 类型元素下的 Value 节点，按下 **Enter** 进入其中第一个子元素（如果 Value 不包含任何元素，自动预留一个空白子元素）；

```
object: {
  key1: null,
  key2: false,
  key4: function () {},
  key3: [
    xxx,
    yyy
  ]
}
```

在 value 节点上按下 Enter 键



```
object: {
  key1: null,
  key2: false,
  key4: function () {},
  key3: [
    xxx,
    yyy
  ]
}
```

进入其中第一个子元素

```
empty: [
  <empty>
],
object: {
  key1: null,
  key2: false,
}
```

在 value 节点上按下 Enter 键



```
empty: [
  null
],
object: {
  key1: null,
  key2: false,
}
```

进入其中第一个子元素

## 5. 添加或删除 object 或 Array 类型元素的子元素

- object 或 Array 类型元素的子元素，按下 **Shift + Enter**，在当前元素上面插入一个兄弟元素；



```
object: {
  key1: null,
  key2: false,
  key4: function () {},
  key3: [
    xxx,
  ]
}
```

在元素节点上按下 Shift+Enter 键



```
object: {
  <empty>:
  key1: null
  key2: boolean
  key4: number () {},
  key3: string
}
```

在当前元素上面插入一个新的元素

- object 或 Array 类型元素的子元素，按下 **Ctrl + Enter**，在当前元素下面插入一个兄弟元素；

```
number: 1,
boolean: true,
list: [ ... ],
empty: [
  <empty>
],
```

在元素节点上按下 Ctrl+Enter 键



```
number: 1,
boolean: true,
<empty>:
list: null
empty: boolean
number
```

在当前元素下面插入一个新的元素

- object 或 Array 类型元素的 Value 节点，按下 Shift + Enter，在最前面插入一个子元素；

```
key3: [
  xxx,
  YYY
]
```

在 Value 节点上按下 Shift+Enter 键



```
key3: [
  null
  xx boolean
  YY number
  string
  array
]
```

在最前面插入一个新的元素

- object 或 Array 类型元素的 Value 节点，按下 **Ctrl + Enter**，在最后面插入一个子元素；

```
key3: [
  xxx,
  YYY
]
```

在 Value 节点上按下 Ctrl+Enter 键



```
key3: [
  xxx,
  YYY,
  null
  boolean
  number
]
```

在最后面插入一个新的元素

- object 或 Array 类型元素的子元素，按下 **Shift + Del**，删除当前子元素；

```
object: {
  key1: null,
  key2: false,
  key4: function () {},
  key3: [
    xxx,
  ]
}
```

在元素节点上按下 Shift+Del 键



```
object: {
  key1: null,
  key4: function () {},
  key3: [
    xxx,
    yyy
  ]
}
```

删除该元素

## 6. 展开或关闭 object 或 Array 类型元素的子元素

- object 或 Array 类型元素的 Value 节点，按下 →、← 可展开或关闭 Value 节点；

```
list: [
  1,
  2
],
```

在 value 节点上按下 → 键



```
list: [ ... ],
empty: [
  <empty>
],
```

关闭 Value 节点

```
list: [ ... ],
empty: [
  <empty>
],
```

在 value 节点上按下 ← 键



```
list: [
  1,
  2
],
```

展开 Value 节点

## 7. 改变元素的数据类型

- 在 Key 节点上，按下 Ctrl + ~ 可重新选择元素的数据类型；

```
boolean: true,
list: [ ... ],
empty: [
  ...
]
```

在 key 节点上按下 Ctrl + ~ 键



```
boolean: true,
list: null
boolean
empty: number
```

重新选择元素的数据类型