## Node.js

Node. js是一个基于 Chrome v8引擎的服务器端 **JavaScript运行环境**；Node. js是一个事件驱动、**非阻塞式I/O的模型**，轻量而又高效.

****Node. js有哪些全局对象？****

**global、 process, console、 module和 exports。**

****Webpack****

**Webpack 是一个用于现代JavaScript应用程序的**静态模块打包工具**，可以很方面的管理模块的依赖。**

****Jquery****

**hover()方法用于模拟光标悬停事件。**

**$("#p1").hover(**

**function(){**

**alert("你进入了 p1!");**

**},**

**function(){**

**alert("拜拜! 现在你离开了 p1!");**

**}**

**);**

**focus()**

**当元素获得焦点时，发生 focus 事件。**

**$("input").focus(function(){**

**$(this).css("background-color","#cccccc");**

**});**

**blur()**

**当元素失去焦点时，发生 blur 事件。**

**$("input").blur(function(){**

**$(this).css("background-color","#ffffff");**

**});**

**通过 jQuery，您可以使用 hide() 和 show() 方法来隐藏和显示 HTML 元素**

**$("p").hide(); $("p").show();**

**显示被隐藏的元素，并隐藏已显示的元素**

**$("p").toggle();**

**text() - 设置或返回所选元素的文本内容**

**html() - 设置或返回所选元素的内容（包括 HTML 标记）**

**val() - 设置或返回表单字段的值**

**jQuery attr() 方法用于获取属性值。**

**append() - 在被选元素的结尾插入内容**

**append() 方法在被选元素的结尾插入内容（仍然在该元素的内部）**

**prepend() - 在被选元素的开头插入内容**

**after() - 在被选元素之后插入内容**

**before() - 在被选元素之前插入内容**

**与text()、html()区别：**

**前者是设置并替换内容，而append()则是插入相应内容，并不做替换。**

**如需删除元素和内容，一般可使用以下两个 jQuery 方法：**

**remove() - 删除被选元素（**及其子元素**）**

**empty() - 从被选元素中删除子元素（类似清空内容）**

**addClass() - 向被选元素添加一个或多个类**

**removeClass() - 从被选元素删除一个或多个类**

**toggleClass() - 对被选元素进行添加/删除类的切换操作**

**css() - 设置或返回样式属性**

**如需返回指定的 CSS 属性的值，请使用如下语法：**

**下面的例子将**返回**首个匹配元素的 background-color 值：**

**$("p").css("background-color");**

**如需设置指定的 CSS 属性，请使用如下语法：**

**$("p").css("background-color","yellow");**

**如需设置多个 CSS 属性，请使用如下语法：**

**$("p").css({"background-color":"yellow","font-size":"200%"});**

**width() 方法设置或返回元素的宽度（**不包括内边距**、边框或外边距）。**

**height() 方法设置或返回元素的高度（**不包括内边距**、边框或外边距）。**

**parent() 方法返回被选元素的直接父元素。**

**$("div").parent();**

**children() 方法返回被选元素的所有直接子元素。**

**$("div").children();**

**find() 方法返回被选元素的后代元素，一路向下直到最后一个后代。**

**$("div").find("span");**

**siblings() 方法返回被选元素的所有同胞元素。**

**$("h2").siblings();**

**next() 方法返回被选元素的下一个同胞元素。(该方法只返回一个元素。)**

**$("h2").next();**

**Vue事件修饰符**

**prevent：阻止默认事件（常用）；**

**一般用于a标签，作用是阻止a标签跳转到href属性的地址。**

**stop：阻止事件冒泡（常用）；**

**点击里部div的click事件，不会触发外层div。**

**once：事件只触发一次（常用）；**

**capture：使用事件的捕获模式；**

**优先触发使用了capture修饰符的事件。**

****UNIAPP****

**scroll-view**

**可滚动视图区域。用于区域滚动。**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| scroll-x | Boolean | false | 允许横向滚动 |
| scroll-y | Boolean | false | 允许纵向滚动 |

**swiper**

**滑块视图容器。**

**一般用于左右滑动或上下滑动，比如banner轮播图。**

|  |  |  |  |
| --- | --- | --- | --- |
| autoplay | Boolean | false | 是否自动切换 |
| current | Number | 0 | 当前所在滑块的 index |
| circular | Boolean | false | 是否采用衔接滑动，即播放到末尾后重新回到开头 |

**icon**

**图标。**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| **type** | String |  | icon的类型 |
| size | Number | 23 | icon的大小，单位px |
| color | Color |  | icon的颜色，同css的color |

| **平台** | **type 有效值** |
| --- | --- |
| App、H5、微信小程序、QQ小程序 | success, success\_no\_circle, info, warn, waiting, cancel, download, search, clear |

**progress**

**进度条。**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| percent | Number | 无 | 百分比0~100 |
| activeColor | Color | #09BB07 |  |
| active | Boolean | false |  |

**button**

**按钮。**

**属性名 类型 默认值 说明**

**size String default 按钮的大小**

**type String default 按钮的样式类型**

**plain Boolean false 按钮是否镂空，背景色透明**

**disabled Boolean false 是否禁用**

**loading Boolean false 名称前是否带 loading 图标**

****size 有效值****

| **值** | **说明** |
| --- | --- |
| default | 默认大小 |
| mini | 小尺寸 |

****checkbox-group****

**多项选择器，内部由多个 checkbox 组成。**

| **属性名** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| @**change** | EventHandle |  | <checkbox-group>中选中项发生改变是触发 change 事件，detail = {value:[选中的checkbox的value的数组]} |

**if(values.includes(item.value)){ 判断数组中是否含某一项**

**this.$set(item,'checked',true) 设置**对象变量**的属性值**

**}else{**

**this.$set(item,'checked',false)**

**}**

**input**

**输入框。**

|  |  |  |  |
| --- | --- | --- | --- |
| **password** | Boolean | false | 是否是密码类型 |

**@**input**|EventHandle||当键盘输入时，触发input事件，event.detail = {value}|差异见下方 Tips| |**

**@**focus**|EventHandle||输入框聚焦时触发，event.detail = { value, height }**

**picker**

**从底部弹起的滚动选择器**

| **属性名** | **类型** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| range | Array / Array<Object> | [] | mode为 selector 或 multiSelector 时，range 有效 |  |
| value | Number | 0 | value 的值表示选择了 range 中的第几个（下标从 0 开始） |  |
| disabled | Boolean | false | 是否禁用 | 快手小程序不支持 |
| @change | EventHandle |  | value 改变时触发 change 事件，event.detail = {value: value} |  |
| @cancel | EventHandle |  | 取消选择或点遮罩层收起 picker 时触发 | 快手小程序不支持 |

**picker-view**

**嵌入页面的滚动选择器。**

**picker-view-column**

**<picker-view /> 的子组件，仅可放置于 <picker-view /> 中，其子节点的高度会自动设置成与 picker-view 的选中框的高度一致。**

* app下退出应用，Android平台可以使用[plus.runtime.quit(](https://www.html5plus.org/doc/zh_cn/runtime.html" \l "plus.runtime.quit" \t "https://uniapp.dcloud.net.cn/component/_blank))。iOS没有退出应用的概念。
* 重启当前的应用
* void plus.runtime.restart();