

<a> 元素

`<a>`元素用来设置链接。除了网页元素的通用接口（`Node`接口、`Element`接口、`HTMLElement`接口），它还继承了`HTMLAnchorElement`接口和`HTMLHyperlinkElementUtils`接口。

属性

URL 相关属性

`<a>`元素有一系列 URL 相关属性，可以用来操作链接地址。这些属性的含义，可以参见`Location`对象的实例属性。

- hash: 片段识别符（以`#`开头）
- host: 主机和端口（默认端口80和443会省略）
- hostname: 主机名
- href: 完整的 URL
- origin: 协议、域名和端口
- password: 主机名前的密码
- pathname: 路径（以`/`开头）
- port: 端口
- protocol: 协议（包含尾部的冒号`:`）
- search: 查询字符串（以`?`开头）
- username: 主机名前的用户名

```
```javascript
```

```
// HTML 代码如下
```

```
// test
```

```
var a = document.getElementById('test');
```

```
a.hash // "#foo"
```

```
a.host // "example.com:8081"
```

```
a.hostname // "example.com"
```

```
a.href // "http://user:passwd@example.com:8081/index.html?bar=1#foo"
```

```
a.origin // "http://example.com:8081"
```

```
a.password // "passwd"
```

```
a.pathname // "/index.html"
```

```
a.port // "8081"
```

```
a.protocol // "http:"
```

```
a.search // "?bar=1"
```

```
a.username // "user"
```

```
```
```

除了`origin`属性是只读的，上面这些属性都是可读写的。

accessKey 属性

`accessKey`属性用来读写`<a>`元素的快捷键。

```
```javascript
// HTML 代码如下
// test
var a = document.getElementById('test');
a.accessKey = 'k';
```
```

上面代码设置`<a>`元素的快捷键为`k`，以后只要按下这个快捷键，浏览器就会跳转到`example.com`。

注意，不同的浏览器在不同的操作系统下，唤起快捷键的功能键组合是[不一样](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes/accesskey)的。比如，Chrome 浏览器在 Linux 系统下，需要按下 Alt + k，才会跳转到`example.com`。

download 属性

`download`属性表示当前链接不是用来浏览，而是用来下载的。它的值是一个字符串，表示用户下载得到的文件名。

```
```javascript
// HTML 代码如下
// 下载
var a = document.getElementById('test');
a.download = 'bar.jpg';
```
```

上面代码中，`<a>`元素是一个图片链接，默认情况下，点击后图片会在当前窗口加载。设置了`download`属性以后，再点击这个链接，就会下载对话框，询问用户保存位置，而且下载的文件名为`bar.jpg`。

hreflang 属性

`hreflang`属性用来读写`<a>`元素的 HTML 属性`hreflang`，表示链接指向的资源语言，比如`hreflang="en"`。

```
```javascript
// HTML 代码如下
// test
var a = document.getElementById('test');
a.hreflang // "en"
```
```

referrerPolicy 属性

`referrerPolicy`属性用来读写`<a>`元素的 HTML 属性`referrerPolicy`，指定当用户点击链接时，如何发送 HTTP 头信息的`referer`字段。

HTTP 头信息的`referer`字段，表示当前请求是从哪里来的。它的格式可以由`<a>`元素的`referrerPolicy`属性指定，共有三个值可以选择。

- `no-referrer`：不发送`referer`字段。
- `origin`：`referer`字段的值是`<a>`元素的`origin`属性，即协议 + 主机名 + 端口。
- `unsafe-url`：`referer`字段的值是`origin`属性再加上路径，但不包含`#`片段。这种格式提供的信息最详细，可能存在信息泄漏的风险。

```
```javascript
// HTML 代码如下
// test
var a = document.getElementById('test');
a.referrerPolicy // "no-referrer"
```
```

rel 属性

`rel`属性用来读写`<a>`元素的 HTML 属性`rel`，表示链接与当前文档的关系。

```
```javascript
// HTML 代码如下
// license.html
var a = document.getElementById('test');
a.rel // "license"
```
```

tabIndex 属性

`tabIndex`属性的值是一个整数，用来读写当前`<a>`元素在文档里面的 Tab 键遍历顺序。

```
```javascript
// HTML 代码如下
// test
var a = document.getElementById('test');
a.tabIndex // 0
```
```

target 属性

`target`属性用来读写`<a>`元素的 HTML 属性`target`。

```
```javascript
// HTML 代码如下
// test
```
```

```
var a = document.getElementById('test');
a.target // "_blank"
```
```

### ### text 属性

`text`属性用来读写`<a>`元素的链接文本，等同于当前节点的`textContent`属性。

```
```javascript
// HTML 代码如下
// <a id="test" href="https://example.com">test</a>
var a = document.getElementById('test');
a.text // "test"
```
```

### ### type 属性

`type`属性用来读写`<a>`元素的 HTML 属性`type`，表示链接目标的 MIME 类型。

```
```javascript
// HTML 代码如下
// <a id="test" type="video/mp4" href="example.mp4">video</a>
var a = document.getElementById('test');
a.type // "video/mp4"
```
```

### ## 方法

`<a>`元素的方法都是继承的，主要有以下三个。

- `blur()`：从当前元素移除键盘焦点，详见`HTMLElement`接口的介绍。
- `focus()`：当前元素得到键盘焦点，详见`HTMLElement`接口的介绍。
- `toString()`：返回当前`<a>`元素的 HTML 属性`href`。