

querySelector()元素选择器

功能: `querySelector()` 方法返回文档中匹配指定 CSS 选择器的第一个元素; 如果你需要返回所有的元素, 请用 `querySelectorAll()` 方法替代;



语法: `document.querySelector(CSS selectors);`

参数值: String 必须。指定一个或多个匹配元素的 CSS 选择器。使用它们的 id, 类, 类型, 属性, 属性值等来选取元素。

对于多个选择器, 使用逗号隔开, 返回一个匹配的元素。

返回值: 匹配指定 CSS 选择器的第一个元素。 如果没有找到, 返回 null。如果指定了非法选择器则抛出 SYNTAX_ERR 异常。

```
1 <div class="box">第1行</div>
2 <div class="box">第2行</div>
3 <div class="box">第3行</div>
4
5 <script>
6     let box = document.querySelector(".box");
7     box.style.color = "red";
8     console.log(box);
9
10    let boxes = document.querySelectorAll(".box");
11    console.log(boxes);
12    for(let i = 0; i < boxes.length; i++)
13        console.log(boxes[i]);
14 </script>
```

第1行
第2行
第3行



querySelector和getElement(s)Byxxx区别

getElement(s)Byxxxx 获取的是动态集合，querySelector 获取的是静态集合。

动态就是选出的元素会随文档改变，静态不会随文档改变，取出来之后就与文档的改变无关了。

```
1 <ul id="wrap">
2   <li>item1</li>
3   <li>item2</li>
4   <li>item3</li>
5 </ul>
6
7 <script>
8   let wrap = document.getElementById("wrap");
9
10  let li = document.getElementsByTagName("li");
11
12  for(let i = 0; i < 10; i++){
13    let newli = document.createElement("li");
14    newli.innerHTML = "item" + i;
15    wrap.appendChild(newli);
16  }
17
18  console.log(li.length);
19 </script>
```

控制台输出13。

```
1 // let li = document.getElementsByTagName("li");
2 let li = document.querySelectorAll("li");
```

控制台输出3。

window.onload方法

window.onload() 方法用于在网页加载完毕后立刻执行的操作，即当 HTML 文档加载完毕后，立刻执行某个方法。

window.onload() 通常用于 元素，在页面完全载入后（包括图片、css文件等等）执行脚本代码。

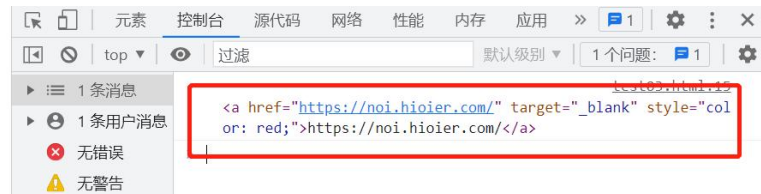
因为 JavaScript 中的函数方法需要在 HTML 文档渲染完成后才可以使用，如果没有渲染完成，此时的 DOM 树是不完整的，这样在调用一些 JavaScript 代码时就可能报出"undefined"错误。

```

1 <a href="https://noi.hioier.com/"
  target="_blank">https://noi.hioier.com/</a>
2
3 <script>
4   window.onload = function(){
5     let obj = document.querySelector("a[target]");
6     console.log(obj);
7
8     obj.style.color = "red";
9   }
10 </script>

```

<https://noi.hioier.com/>



键盘事件

键盘事件就是按下、抬起按键所触发的事件，区分不同按键使用键码值，即ASCII值。

常用键码值							
按键	0	1	9	<u>BachSpace</u>	Tab	Enter
keycode	48	49	57	8	9	13

```

1 window.onload = function(){
2   let input = document.getElementById("txt");
3   input.onkeyup = function(e) {
4     let keyCode = e.keyCode;
5     if (keyCode >= 48 && keyCode <= 57) {
6       alert("啦啦，你按下的是数字键！");
7     } else {
8       alert("哈哈，你没有按下数字键哦！")
9     }
10  }
11 }

```

addEventListener事件监听

常见的触发函数有：

鼠标

- click: 鼠标左键点击
- dblclick: 鼠标左键双击
- contextmenu: 鼠标右键点击
- mousedown: 鼠标按下，包括左键、滚轮、右键

- event.button: 0表示左键, 1表示中键, 2表示右键
- mouseup: 鼠标弹起, 包括左键、滚轮、右键
- event.button: 0表示左键, 1表示中键, 2表示右键

键盘

- keydown: 某个键是否被按住, 事件会连续触发
- keyup: 某个按键是否被释放
- keypress: 紧跟在keydown事件后触发, 只有按下字符键时触发。适用于判定用户输入的字符。
- event.altKey、event.ctrlKey、event.shiftKey分别表示是否同时按下了alt、ctrl、shift键。

表单

- focus: 聚焦某个元素
- blur: 取消聚焦某个元素
- change: 某个元素的内容发生了改变

窗口: 需要作用到window元素上。

- resize: 当窗口大小发生变化
- scroll: 滚动指定的元素
- load: 当元素被加载完成

```
1 window.addEventListener("keydown", function(e){
2     let keyCode = e.keyCode;
3     if (keyCode >= 48 && keyCode <= 57) {
4         alert("啦啦, 你按下的是数字键!");
5     } else {
6         alert("哈哈, 你没有按下数字键哦!")
7     }
8 });
```