

# JavaScript第三节

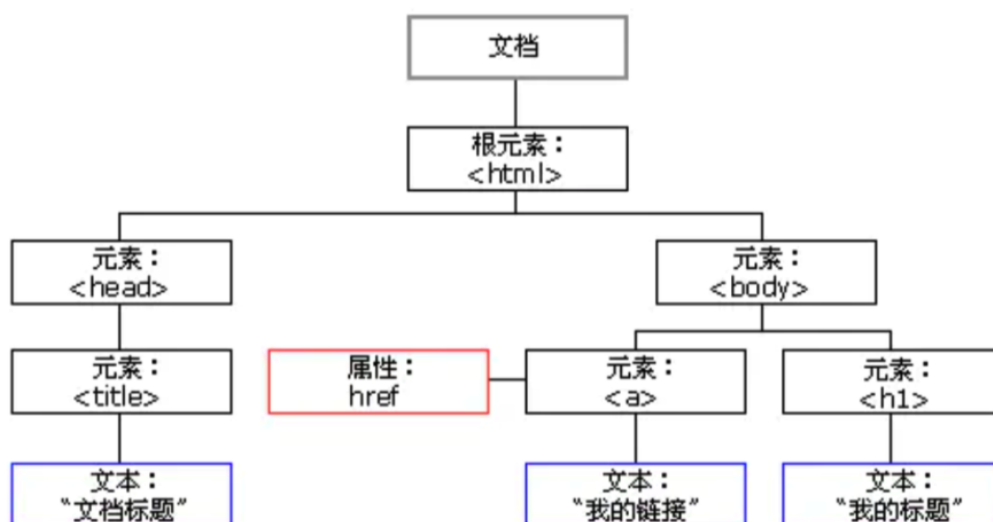
学习目标:

- 1.理解DOM
- 2.掌握dom获取元素的几种方式
- 2.理解事件
- 3.理解定时器

## 一、DOM

### 1、获取标签元素

什么是DOM?



文档对象模型（Document Object Model，简称Dom），是W3C组织推荐的处理可拓展标记语言（HTML）的标准编程接口。

文档：一个页面就是一个文档，DOM中使用document表示。

元素：页面中的所有标签都是元素，DOM中使用element表示。

节点：网页中的所有内容都是节点（标签，属性，文本，注释等），DOM中使用node表示。

DOM在我们实际开发中主要用来操作元素的。

可以使用内置对象**document**上的**getElementById**方法来获取页面上设置了id属性的标签元素，获取到的是一个html对象，然后将它赋值给一个变量。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
```

```

<body>
  <p id="p1">我是一个段落标签</p>
  <script>
    // js的内置对象，html的文档对象
    let op = document.getElementById("p1");
    alert(op);
  </script>
</body>
</html>

```



## 2、选择器获取元素

querySelector会选择捕捉到的第一个元素，querySelectorAll会将捕捉到的全部元素按照数组形式返回。

```

//单个选择
document.querySelector('选择器')
//选取全部
document.querySelectorAll('选择器')

```

```

<div class="father">
  <div class="son">
  </div>
</div>
<script>
  console.log(document.querySelector('.son'))
  console.log(document.querySelectorAll('div'))
</script>

```



## 3、操作标签元素属性

### 1、获取和改变标签的操作

innerText：获取或修改元素内的文字，不包含标签和样式；  
innerHTML：获取或修改元素内所有内容，包括标签和样式

```
<div>123</div>
<script>
  // innerText
  let div = document.querySelector('div');
  // 获取标签内容
  alert(div.innerText);
  div.innerText = '<h3>Hello<h3>';
</script>
```

<h3>Hello<h3>

```
<div>123</div>
<script>
  // innerHTML
  let div = document.querySelector('div');
  // 获取标签内容
  alert(div.innerHTML);
  div.innerHTML = 'Hello';
</script>
```

Hello

## 2.属性的操作

1.属性的获取。2.属性的设置。

属性名在js中的写法：

- 1、html的属性和js里面属性大多数写法一样，但是class属性写成className
- 2、"style"属性里面的属性，有横杠的改成驼峰式，如："font-size"，写成"style.fontSize"

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>

  <style>
    .btnstyle{
      background: red;
      font-size: 30px;
    }
  </style>

</head>
<body>
  <input type="button" value="按钮" id="btn1">
  <script>
    // 根据id获取标签对象
```

```

    let oBtn = document.getElementById("btn1");
    // 获取标签的属性
    alert(oBtn.type);
    alert(oBtn.value);
    // 设置标签的属性
    oBtn.name = "username";
    // 设置样式的属性
    // oBtn.style.background = "red";
    // 相当于设置class，以后通过类选择器给标签添加样式
    // 注意: class这里变成了className
    oBtn.className = "btnstyle"
    // oBtn.style.fontSize = "30px";
  </script>
</body>
</html>

```

## 二、事件

### 1、事件概述

JavaScript使我们有能力创建动态页面，而事件是可以被JavaScript侦测到的行为。

事件是由三部分组成，事件源，事件类型，事件处理程序（事件三要素）

事件源：事件被触发的对象

事件类型：如何触发 什么事件

事件处理程序：当这个事件被触发之后会发生什么样的事情

网页中的每个元素都可以产生

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <button id="btn">唐伯虎</button>
</body>
<script>
  let btn = document.getElementById('btn');
  // 事件点击出发这个函数的操作
  btn.onclick = function(){
    alert('点秋香');
  }
  // 另一种写法:
  //btn.addEventListener("click",()=>{
  //  alert('点秋香')
  //})
</script>
</html>

```

## 三、定时器

# 1、定时器的介绍

定时器就是在一定特定的时间后执行某段程序代码。

## 2、定时器的使用

js定时器有两种创建方式：

1、setTimeout(func,[delay,param1.....]):以指定的时间间隔（以毫秒计）调用一次函数的定时器。

2、setInterval(func,[delay,param1.....]):以指定的时间间隔（以毫秒计）重复调用一个函数的定时器

参数说明：

第一个参数func：表示定时器要执行的函数名

第二个参数delay：表示时间间隔，默认是0，单位是毫秒

第三个参数param1，表示定时器执行函数的第一个参数，以此类推传入多个执行函数对应的参数

示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>

</head>
<body>
  <input type="button" value="停止" onclick="fnStop();">
  <!-- 定时器：根据指定的时间间隔延时调用指定的函数 -->
  <script>
    // 定义一个函数
    // function fnShow(name, age){
    //     alert("OK" + name + age);
    // }

    // 根据时间间隔调用一次函数的定时器
    // 1.定时器要执行的函数
    // 2.时间间隔
    // 3.参数，多个参数使用逗号进行分割
    // let tid = setTimeout(fnShow, 500, '李四', 20)
    // 销毁计时器
    // clearTimeout(tid);

    // 根据时间间隔重复函数的定时器
    function fnShowInfo(name, age){
      alert('OK' + name + age);
    }
    // 定义一个销毁计算器的函数
    function fnStop(){
      // 销毁计时器
      clearInterval(sid);
    }
  </script>
</body>
</html>
```

```
let sid = setInterval(fnShowInfo, 3000, '张三', 22);  
  
    </script>  
</body>  
</html>
```