

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
```

DOM 称为文档对象模型, 通过DOM javascript 可以访问HTML文档中的所有元素

JS可以改变页面中所有的HTML元素
JS可以改变页面中所有的HTML属性
JS可以改变页面中所有的CSS样式
JS可以对页面中的所有事件做出反应

通常JAVASCRIPT需要操作HTML元素之前必须要找到这个元素, 有三种方法

- 1.通过ID找到HTML元素
- 2.通过标签名找到HTML元素
- 3.通过类名找到HTML元素

```
<input type="text" id="username"><br/>
<button onclick="f1()">获取input内容</button>
<br/>
```

```
<div id="m1">
  <p>M1下的P</p>

</div><br/>
```

```
<button onclick="f2()">通过标签来获取</button><br/>
```

```
<p class="c1">通过class获取内容</p><br/>
<p class="c1">通过class获取内容2</p><br/>
```

```
<button onclick="f3()">获取c的内容</button>
```

```
<script>
  function f1(){
    //根据ID获取 标签的ID是全局唯一的
    var x = document.getElementById("username");
    document.write(x.value)
```

```
}
```

```
function f2(){  
    //根据标签类型获取，标签ID下有多个相同的标签， 所以根据 y[0] y[1] 这种形式来获取
```

```
    var x = document.getElementById("m1");  
    var y = x.getElementsByTagName("p");  
    document.write(y[0].innerHTML)
```

```
}
```

```
function f3(){  
    // 多个标签拥有相同的class, 所以用 x[0] x[1] 这种来获取属于同一个  
    //class的标签
```

```
    var x= document.getElementsByClassName("c1");  
    document.write(x[0].innerHTML);  
    document.write(x[1].innerHTML);
```

```
}
```

```
</script>
```

DOM改变HTML

```
<p id="pp1">这是老的内容</p><br/>  
<button onclick="f4()">点我</button><br/>
```

```
<br/>  
<button onclick="f5()">改变属性</button>
```

```
<script>
```

```
    //改变HTML内容
```

```
    function f4(){  
        var x = document.getElementById("pp1");  
        x.innerHTML = "这是新的内容;"  
    }
```

```
    //改变标签属性
```

```

function f5(){
    var x = document.getElementById("img1");
    x.src = "https://ss1.baidu.com/6ONXsjip0QIZ8tyhnq/it/
u=691297921,1036193612&fm=173&app=25&f=JPEG?
w=218&h=146&s=CFBA29C54A0638CE17B019810300C08A"
}

```

</script>

通过JS改变HTML样式

<p id="pp2" style="background-color: red">背景是红色</p>

<button onclick="f6()">点击改变颜色</button>

```

<script>
function f6(){
    var x = document.getElementById("pp2");
    x.style.backgroundColor = "green";
}

```

</script>

创建新的HTML元素-appendChild()

<div id="div1"></div>

<script>

```

var p = document.createElement("p"); //用于创建p元素
var v = document.createTextNode("第一行数据"); //p元素中添加文本
p.appendChild(v); //将文本节点添加到p标签中
document.getElementById("div1").appendChild(p); //将p标签添加到div1中

```

</script>

insertBefore 在指定的标签之前插入....

<div id="div2">

```
<p id="p1">第一行</p>
<p id="p2">第二行</p>
</div>

<script>

    var p = document.createElement("p");
    var v = document.createTextNode("这是最新的一行");
    p.append(v); //创建新的一行

    var d = document.getElementById("div2");
    var f = document.getElementById("p1");
    d.insertBefore(p, f) //在div2里面的p1元素之前 添加新的元素p

</script>
```

HTML元素的删除

```
<div id="div3">
    <p>第一行</p>
    <p>第二行</p>
    <p>第三行</p>
    <p>第四行</p>
</div><br/>

<button onclick="f7()">点我删除</button>

<script>

    function f7(){
        var x = document.getElementById("div3");
        var y = x.getElementsByTagName("p");
        console.log(y.length - 1);
        x.removeChild(y[y.length-1]) //点一下鼠标删掉一行p
    }

</script>
```

HTML元素的替换

```
<div id="div4">
    <p id="xp1">老数据</p>
```

```
</div>
```

```
<button onclick="f8()">点我替换</button>
```

```
<script>
```

```
    function f8(){  
        var n = document.createElement("p");  
        var nt = document.createTextNode("新数据");  
        var news = n.appendChild(nt);  
  
        var parent = document.getElementById("div4");  
        var old = document.getElementById("xp1");  
  
        parent.replaceChild(news, old)}
```

```
</script>
```

前面我们使用getElementsByTagName() 这些方法的时候返回 HTMLCollection 对象， 这个对象的结构是类似一个HTML的数组

可以使用索引 y[0] y[1] 这种获取指定的元素
也可以使用 length 获取其长度