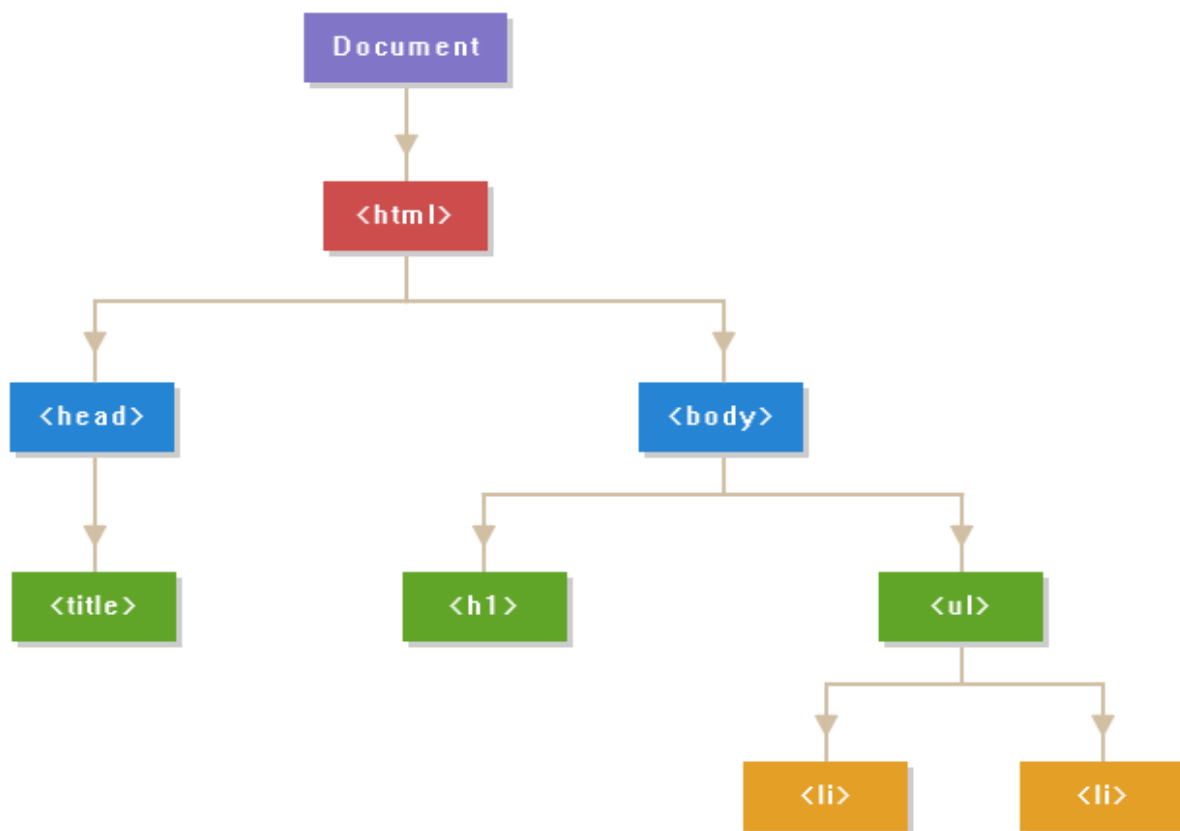


# JavaScript第三节课 Dom

## 一、什么是dom? 我们通过一张图来了解



dom可以理解为一个一个的标签（节点）

## 二、为什么要有dom? 有什么作用

我们可以通过dom访问和修改操作让网页动起来

## 三、获取dom

document 代表整个网页，相当于dom树的入口

要获取 DOM 元素，可以使用 JavaScript 中的各种方法和属性。下面是几个方法介绍：

- **getElementById**: 通过元素的 ID 获取dom。

```
// 语法:  
let element = document.getElementById("elementId");  
// elementId : 标签中的id值，类似于css中的id选择器
```

```
// 举例
<!-- 定义一个p标签,设置id选择器p001 -->
<p id="p001">今天天气很好</p>

<script>
    // p 是我们获取的 id为p的<p>标签的dom节点
    let p1 = document.getElementById("p001");
    // 在控制台打印获取的 p 标签dom信息
    console.log("第一个",p1);
</script>
```

- **getElementsByClassName**: 通过类名获取元素集合（返回的是一个 HTMLCollection）。

```
// 语法:
let elements = document.getElementsByClassName("className")
// className : 标签中的class值, 类似于css中的class选择器
```

```
// 举例:
<!-- 定义了一个p标签, 设定class选择器名p002 -->
<p class="p002">这是我的第二个P标签</p>
<p class="p002">这是我的第三个P标签</p>
<p class="p002">这是我的第四个P标签</p>

<script>
    // 根据类名获取获取该类名对应的所有数据, 返回的是一个数据集合对象
    let p2 = document.getElementsByClassName("p002") ;
    console.log("第二个",p2);
</script>
```

- **querySelector**: 通过选择器获取第一个匹配的元素。

```
//语法:
let element = document.querySelector("selector");
//可以通过元素的ID、类名、标签名或其他CSS选择器来选择和获取元素。
```

```
//举例:
//选择ID为"myElement"的元素:
var element = document.querySelector("#myElement");
//选择类名为"myClass"的第一个元素:
var element = document.querySelector(".myClass");
//选择标签名为"div"的第一个元素:
var element = document.querySelector("div");
// 可以打印dom在控制台来看看效果
```

## 四、修改样式

要修改 DOM 元素的样式, 可以使用 JavaScript 中的 `style` 属性。`style` 属性允许我们直接访问和修改元素的样式属性。

```
// 实例：
//使用 style.属性名：通过直接设置元素的样式属性来修改样式。
let bon = document.querySelector('button')
// 通过style控制样式 color 为red
bon.style.color = 'red'
// 修改button按钮的 字体大小bon.style.fontSize= '60px'
bon.style.fontSize= '60px'
```

## 五、获取文本内容

- `innerHTML` 获取或设置元素的 `HTML` 内容

```
<p>文本内容 哈哈</p>

<script>
// 获取
let p1 = document.querySelector('p')
console.log(p1.innerHTML);
// 修改
p1.innerHTML = "<p>呵呵呵</p>"
console.log(p1.innerHTML);
/*
innerHTML 属性不仅可以获取文本内容，还可以获取包括 HTML 标签在内的整个元素内容。
*/
</script>
```

- `value` 获取到输入框中的内容(默认值内容)

```
<input type="text" value="12312">

<script>
// 获取
let text = document.querySelector('input')
console.log(text.value);
// 修改
text.value = "999999999"
console.log(text.value);
/*
需要注意的是，当获取输入框的值时，value 属性返回的是字符串类型。如果需要进行数值计算或其他操作，可能需要将其转换为适当的数据类型。
*/
</script>
```

## 六、事件

`addEventListener()` 是 JavaScript 中用于添加事件监听器的函数。它允许指定一个特定的事件类型，并在事件发生时执行相应的处理函数。

```
// addEventListener() 的语法如下：
dom对象.addEventListener(event,listener)
// event : 要监听的事件类型
// listener : 回调函数，事件发生时要执行的处理函数。
```

## 1. click 点击事件

应用场景：点击后发生什么事情

案例：

```
<button id="myButton">点击登录</button>

<script>
  let button = document.getElementById('myButton')
  button.addEventListener('click', ()=>{
    // 在点击事件发生时执行的处理函数
    alert("按钮被点击了！")
  })
</script>
```

通过点击事件，您可以在用户与页面进行交互时执行特定的操作，例如提交表单、打开链接、显示/隐藏元素等等。

## 2. change 触发事件

**change 事件是在表单元素的值发生改变并失去焦点时触发的事件。**

应用场景：获取输入框值

```
<div>
  请输入账号：
  <input id="myInput" type="text">
</div>
<script>
  // 获取 input 标签的 dom
  let userName = document.getElementById("myInput") ;
  // 添加点击监听事件
  userName.addEventListener('change', ()=>{
    let x = userName.value;
    console.log("刚刚输入的用户名是： "+x);
  })
</script>
```

## 3. input 触发事件

**input 事件是在用户输入内容时实时触发的事件。它在输入框的值发生变化时立即触发。而不需要等待失去焦点（change 事件）**

```
<div>
  请输入密码：
  <input id="password" type="password">
</div>
<script>
  // 获取 密码输入框的 dom
  let password = document.getElementById("password") ;
  // 给密码输入框dom对象 添加一个事件监听器，监听input事件
  // 只要该事件发生，就在控制台打印密码输入框获取的值
  password.addEventListener('input', ()=>{
```

```
let y = password.value;  
console.log("当前输入的密码是: "+y);  
})  
</script>
```

## 课后作业:

---

1、实现敲木鱼小程序, 根据老师提供基础代码, 实现:

需求一: 实现点击+1按钮, 让功德数字+1

需求二: 新增一个电脑托管按钮, 实现让电脑帮我们敲 (定时器)

拓展: 在完成以上几点需求后, 尝试利用`clearInterval`函数实现让定时器停下来。 (关于`clearInterval`函数需要同学们上网搜寻答案)

2、实现登录界面。

需求:

1、用户在输入账户和密码后, 在登录过程中, 检测密码是否是八位数

4、登录是否成功有弹窗提示

5、登录成功后调转到其他页面

拓展: 登录成功后调转到其他页面 提示: `window.location.href="https://www.bilibili.com/?spm_id_from=333.788.0.0";`