提前准备:

1. 今天的历届师兄师姐问过的经典的高频的问题清单都已经整理在小程序->首页->DOM->day04里

2. 今天的课程中所需的单词列表都已经整理在小程序->单词->DOM->day04里

3. 今天的重点视频、扩展高频鄙视题视频已经整理在小程序->在线->DOM->day04里

如果不知道问什么: 谁问的越多！谁培训费花的越值！

1. 笔记中或画图中那句话或者那个位置不明白，立刻问！

2. 案例中哪行代码或注释不明白，立刻问！

3. 课下自己做项目，功能不会做，立刻问！

4. 刷笔试题时遇到不会的笔试题，立刻问！——公众号: 前端大全 定期总结大厂笔试题

5. 看小程序问题清单时，遇到看不懂的问题，立刻问！

今天开始！每个案例三遍:

1. 连注释+代码抄一遍

2. 保留注释，删掉代码，自己试着填回来

3. 删除注释和代码，自己用人话把注释写回来！——才算自己会！

正课:

1. BOM

2. \*\*\*\*\*事件

总结:

1. 查找元素: 4种

(1). 不需要查找就可直接获得的元素: 5个

a. document

b. <html> document.documentElement

c. <head> document.head  
 d. <body> document.body

e. <form> document.forms[i]

(2). 按节点间关系查找: 2大类关系, 6个属性

何时: 已经获得一个元素，想找它周围附近的元素时

a. 父子关系:

1). 获得一个元素的父元素: 元素.parentNode

2). 获得一个元素下的所有直接子元素: 元素.children

3). 获得一个元素下的第一个直接子元素: 元素.firstElementChild

4). 获得一个元素下的最后一个直接子元素: 元素.lastElementChild

b. 兄弟关系:

1). 获得一个元素的前一个兄弟元素: 元素.previousElementSibling

2). 获得一个元素的后一个兄弟元素: 元素.nextElementSibling

(3). 按HTML特征查找：4个

何时: 只需要一个条件就可找到想要的元素时

a. 按id查找一个元素: var 一个元素=document.getElementById(“id名”)

b. 按标签名查找多个元素: var 类数组对象=任意父元素.getElementsByTagName(“标签名”)

c. 按class名查找多个元素: var 类数组对象=任意父元素.getElementsByClassName(“class名”)

d. 按name名查找多个表单元素: var 类数组对象=document.getElementsByName(“name名”)

(4). 按选择器查找: 2个

何时: 查找条件复杂时

a. 只查找一个符合条件的元素: var 一个元素=任意父元素.querySelector(“选择器”)

b. 查找多个符合条件的元素: var 类数组对象=任意父元素.querySelectorAll(“选择器”)

|  |
| --- |
| 总结: 返回值:  1. 如果原函数返回下标位置i, 如果找不到返回: -1  2. 如果原函数返回一个对象或数组，如果找不到返回: null  3. 如果原函数返回一个类数组对象，如果找不到返回: 空类数组对象{ length:0 } |

2. 修改元素: 3种:

(1). 内容: 3种:

a. 获取或修改元素的原始的HTML内容: 元素.innerHTML

b. 获取或修改元素的纯文本内容: 元素.textContent

c. 获取或修改表单元素的值: 元素.value

(2). 属性: 3种:

a. 字符串类型的HTML标准属性: 2种:

1). 核心DOM 4个函数: 元素.getAttribute()/setAttribute()/removeAttribute()/has Attribute();

2). HTML DOM简写: 元素.属性名

b. bool类型的HTML标准属性: 只能: 元素.属性名

c. 自定义扩展属性: 2种:

1). 核心DOM4个函数: 元素.getAttribute()/setAttribute()/removeAttribute()/has Attribute();

2). 如果自定义扩展属性名以data-开头，则可用: 元素.dataset.自定义属性名 (不要加data-前缀)

(3). 样式: 3种:

a. 修改一个元素的一个css属性: 元素.style.css属性

b. 获取一个元素的css属性: var style=getComputedStyle(元素对象); style.css属性名

c. 批量修改一个元素的多个css属性: 元素.className

3. 添加新元素:

(1). 只添加一个新元素: 3步:

a. 创建一个新的空元素对象: var 新元素=document.createElement(“元素名”)

b. 为新元素添加必要属性: 新元素.属性名=属性值

c. 将新元素添加到DOM树: 3种:

1). 追加到指定父元素下的现有子元素结尾: 父元素.appendChild(新元素)

2). 插入到指定父元素下一个现有子元素之前: 父元素.insertBefore(新元素,现有子元素)

3). 替换指定父元素下一个现有子元素: 父元素.replaceChild(新元素,现有子元素)

(2). 优化: 如果要添加多个元素到页面，需要尽量减少操作DOM树的次数

a. 如果同时添加父元素和子元素时，应该先在内存中将所有子元素加入父元素中，最后再一次性将父元素添加到DOM树

b. 如果父元素已经在页面上，需要添加多个平级子元素时，应该先将多个平级子元素加到文档片段中，最后再一次性将文档片段添加到DOM树: 3步:

1). 创建文档片段: var frag=document.createDocumentFragment();

2). 将子元素加入文档片段中: frag.appendChild(子元素)

3). 将文档片段整体加入DOM树: 父元素.appendChild(frag)

4. 删除元素: 父元素.removeChild(子元素)

HTML DOM的简写:

1. <img>: var img=new Image();

2. <table>:

(1). table管着行分组:

a. 添加行分组: var thead=table.createTHead(); var tbody=table.createTBody(); var tfoot=table.createTFoot();

b. 删除行分组: table.deleteTHead(); table.deleteTFoot();

c. 获取行分组: table.tHead table.tFoot table.tBodies[i]

(2). 行分组管着行:

a. 添加行: 3种:

1). 末尾添加一个新行: var tr=行分组.insertRow()

2). 开头插入一个新行: var tr=行分组.insertRow(0)

3). 在中间插入一个新行: var tr=行分组.insertRow(i)

b. 删除行: table.deleteRow(tr.rowIndex);

c. 获取行: 行分组.rows[i]

(3). 行管着格:

a. 添加格: var td=tr.insertCell(); //在行末尾追加一个新格

b. 删除格: tr.deleteCell();

c. 获取格: tr.cells[i]

3. <form>:

(1). 获取<form>中的表单元素:2种:

a. 标准: form.elements[i或name或id]

b. 如果表单元素有name属性可以直接: from.name

(2). 让一个表单元素自动获得焦点: 表单元素.focus();

总结: this

1. obj.fun() this->obj

2. fun() 或 (function(){…})() 或 回调函数 中的this->window

3. new Fun() this->new将来创建的某个子对象

4. 原型对象.prototype.fun=function(){ … } this->将来调用这个方法的.前的某个子对象

5. 访问器属性中的this->当前访问器属性所在的对象

Object.defineProperties(eric,{

eage:{

get(){ return this.\_eage },

set(value){ this.\_eage=value }

}

})

6. 元素.onclick=function(){ … } this->当前正在触发事件的元素