

# 网站前端开发技术

主讲人：张立 GIS博士

制作人：人工智能学院 张立 博士

1

## 教学内容安排

- ☺ DOM简介
- ☺ 网页元素操作
- ☺ 节点操作

制作人：人工智能学院 张立 博士

2

## DOM简介

制作人：人工智能学院 张立 博士

3

## DOM概述

### 本质

- 处理HTML文档的技术

### 用途

- 操纵页面元素的内容和样式

制作人：人工智能学院 张立 博士

4

## 核心 DOM

### ☺ 功能

- ✧ 提供了操作文档的公有属性和方法

### ☺ 对象

- ✧ Document文档, Element元素
- ✧ Attribute属性, Text文本
- ✧ Comment注释, Node节点

制作人：人工智能学院 张立 博士

5

## document对象

### 本质

- DOM文档树的根

### 特点

- 提供元素对象的属性和方法

### 用途

- 通过它来操控页面元素

制作人：人工智能学院 张立 博士

6

## 网页元素操作

制作人：人工智能学院 张立 博士

7

## 获取网页元素-1

- ☺ `getElementsByTagName()`:  
✧ 返回指定标签对象的元素集合。
- ☺ `getElementById()`:  
✧ 返回指定id对象的元素。
- ☺ `getElementsByName()`:  
✧ 返回指定name对象的元素集合。
- ☺ `getElementsByClassName()`:  
✧ 返回所有指定类名的元素集合。

制作人：人工智能学院 张立 博士

8

## 获取网页元素-2

- ☺ `querySelector()`:  
✧ 返回匹配制定CSS选择器的第一元素。
- ☺ `querySelectorAll()`:  
✧ 返回匹配CSS选择器的所有元素集合。

制作人：人工智能学院 张立 博士

9

### 获取元素的内容

#### innerHTML

- 返回对象内的全部内容

#### innerText

- 返回对象内的去除HTML标签后的内容

#### outerHTML

- innerHTML 内容+对象标签本身

制作者：人工智能学院 张立 博士

10

### 元素的位置与大小

- ③ offsetLeft和offsetTop
  - ✧ 获取元素到父元素边框的距离
- ③ offsetWidth和offsetHeight
  - ✧ 获取元素自身的大小
  - ✧ 包括内边距与边框宽度

制作者：人工智能学院 张立 博士

11

### 演示与实践

- ③ 任务：网页元素操作。

让我们一起动手吧！

制作者：人工智能学院 张立 博士

12

### 标签代码

```

③ <body>
③ <h2 id="header">网页前端开发技术</h2>
③ <ul class="list1">
③ <li>HTML5</li>
③ <li>css3</li>
③ <li>JavaScript</li>
③ </ul>
③ </body>

```

制作者：人工智能学院 张立 博士

13

### 获取网页元素-1

```

③ var li=document.getElementById("li");
③ console.log(li.innerHTML);
③ li[0].innerHTML="PHP";
③ var h2=document.getElementById("header");
③ console.log(h2.innerHTML);
③ h2.innerHTML="前端开发学习";
③ var ul=document.getElementsByClassName("list1")[0];
③ console.log(ul.innerHTML);

```

制作者：人工智能学院 张立 博士

14

### 获取网页元素-2

```

③ var li=document.querySelector("li");
③ console.log(li);

③ var li=document.querySelectorAll("li");
③ console.log(li[2].innerHTML);
③ li[2].innerHTML="jQuery";

```

制作者：人工智能学院 张立 博士

15

### 演示与实践

- ③ 任务：修改元素样式。

让我们一起动手吧！

制作者：人工智能学院 张立 博士

16

### 直接设置style属性

```

③ <h2 id="header">前端开发学习</h2>
③ var h2=document.getElementById(header);
③ h2.style.color='red';
③ h2.style['color']='red';
③ h2.style['background-color']='red';
③ h2.style.backgroundColor='red';

```

制作者：人工智能学院 张立 博士

17

### setProperty函数

```

③ <h2 id="header">前端开发学习</h2>
③ var h2=document.getElementById(header);
③ h2.style.setProperty('color','red');
③ h2.style.setProperty('background-color','red');

```

制作者：人工智能学院 张立 博士

18

## 设置className

- ③ CSS选择器
  - ✧ `.myclass1{ color:red;}`
- ③ JSD代码
  - ✧ `var h2=document.getElementById("header");`
  - ✧ `h2.className="myclass1";`



制作者：人工智能学院 张士 博士

19

## 节点操作

制作者：人工智能学院 张士 博士

20

## 子节点

- ③ `childNodes`
  - ✧ 返回的是节点的子节点集合
  - ✧ 包括元素节点、文本节点和属性节点
- ③ `children`
  - ✧ 返回的只是节点的元素节点集合



制作者：人工智能学院 张士 博士

21

## 一般人我不告诉他

在判断某节点是否有子节点的时候，应尽量使用`children`。



制作者：人工智能学院 张士 博士

22

## 子节点

- ③ 当前节点的第一个子节点
  - ✧ `firstChild` (返回文本/元素节点等)
  - ✧ `firstElementChild` (元素节点)
- ③ 当前节点的最后一个子节点
  - ✧ `lastChild` (返回文本/元素节点等)
  - ✧ `lastElementChild` (元素节点)



制作者：人工智能学院 张士 博士

23

## 父节点

### `parentNode`

- ✧ 获取指定节点的父节点

### `offsetParent`

- ✧ 获取距离该子元素最近的有定位的父元素

### `closest`

- ✧ 获取满足筛选条件的最近的父节点



制作者：人工智能学院 张士 博士

24

## 兄弟节点

- ③ 当前节点后面的第一个同级节点
  - ✧ `nextSibling` (返回文本/元素节点等)
  - ✧ `nextElementSibling` (仅返回元素节点)
- ③ 当前节点前面的第一个同级节点
  - ✧ `previousSibling` (返回文本/元素节点等)
  - ✧ `previousElementSibling` (仅返回元素节点)



制作者：人工智能学院 张士 博士

25

## 演示与实践

- ③ 任务：节点关系。

让我们一起动手吧！



制作者：人工智能学院 张士 博士

26

## 创建节点

- ③ 创建方法
  - ✧ `createElement()` 创建元素节点对象
  - ✧ `createTextNode()` 创建文本节点对象
- ③ 本质
  - ✧ `Document`对象的方法



制作者：人工智能学院 张士 博士

27

## 添加节点

- ③ `appendChild()`
  - ❖ 将节点添加到子节点列表的末尾
- ③ `insertBefore(node,[refnode])`
  - ❖ 在`refnode`节点之前插入`node`



制作者：人工智能学院 张立 博士

28

## 演示与实践

- ③ 任务：移动节点。

让我们一起动手吧！



制作者：人工智能学院 张立 博士

29

## 参考代码-1

```

③ <ul id="mylist">
③   <li style="color:red">项目01</li>
③   <li>项目02</li>
③   <li>项目03</li>
③   <li>项目04</li>
③ </ul>
③ <button onClick="moveNode1()">移动列表-1</button>
③ <button onClick="moveNode2()">移动列表-2</button>

```



制作者：人工智能学院 张立 博士

30

## 参考代码-2

```

③ function moveNode1() {
③   ❖ var parNode = document.getElementById("mylist");
③   ❖ var firstNode = parNode.firstChild;
③   ❖ parNode.appendChild(firstNode);
③ }

```



制作者：人工智能学院 张立 博士

31

## 参考代码-3

```

③ function moveNode2() {
③   ❖ var parNode = document.getElementById("mylist");
③   ❖ var firstNode = parNode.firstChild;
③   ❖ var lastNode = parNode.lastElementChild;
③   ❖ parNode.insertBefore(lastNode, firstNode);
③ }

```



制作者：人工智能学院 张立 博士

32

## 复制节点

- ③ `pNode.cloneChild(deep)`
- ③ 其中
  - ❖ `pNode`是父元素对象
  - ❖ `deep`值
    - \* `true`: 深度复制，复制当前节点+子节点；
    - \* `false`: 普通复制，只复制当前节点。



制作者：人工智能学院 张立 博士

33

## 演示与实践

- ③ 任务：复制节点。

让我们一起动手吧！



制作者：人工智能学院 张立 博士

34

## 参考代码

```

③ <ul id="mylist" style="border:1px solid blue">
③   ❖ <li style="color:red">项目001</li>
③   ❖ <li>项目002</li>
③   ❖ <li>项目003</li>
③   ❖ <li>项目004</li>
③ </ul>
③ <div id="div1" style="border:1px solid red"></div>
③ <button onClick="AddClone(false)">普通复制</button>
③ <button onClick="AddClone(true)">深度复制</button>

```

制作者：人工智能学院 张立 博士

35

## 参考代码

```

③ function AddClone(flag) {
③   ❖ var myul=document.getElementById("mylist");
③   ❖ var newul=myul.cloneNode(flag);
③   ❖ var mydiv=document.getElementById("div1");
③   ❖ mydiv.appendChild(newul);
③   ❖ var newbr=document.createElement("br");
③   ❖ mydiv.appendChild(newbr);
③ }

```



制作者：人工智能学院 张立 博士

36

### 替换节点

#### 方法

• pNode.replaceChild(new,old)

#### 说明

• new是要替换的节点，old是原来的节点

#### 功能

• 将子节点列表中的old节点用new节点替换

制作者：人工智能学院 张士 博士

37

### 演示与实践

☺任务：替换节点。

让我们一起动手吧！



制作者：人工智能学院 张士 博士

38

### 参考代码

```
☺ <ul id="mylist">
  ✧ <li>项目 01</li>
  ✧ <li>项目 02</li>
  ✧ <li>项目 03</li>
  ✧ <li>项目 04</li>
☺ </ul>
☺ <button onclick="replaceNode()">替换列表</button>
```



制作者：人工智能学院 张士 博士

39

### 参考代码

```
☺ function replaceNode() {
  ✧ var parentNode = document.getElementById("mylist");
  ✧ var firstP = parentNode.firstChild;
  ✧ var newLi = document.createElement("li");
  ✧ var newText = document.createTextNode("新的选项");
  ✧ newLi.appendChild(newText);
  ✧ parentNode.replaceChild(newLi, firstP);
☺ }
```



制作者：人工智能学院 张士 博士

40

### 删除元素节点

#### 方法

• pNode.removeChild(cnode)

#### 说明

• pNode是要删除节点的父元素  
• cnode是要删除的子元素节点名称

制作者：人工智能学院 张士 博士

41

### 演示与实践

☺任务：删除节点。

让我们一起动手吧！



制作者：人工智能学院 张士 博士

42

### 参考代码

```
☺ <h1>深圳</h1>
☺ <hr />
☺ <h2>地理位置</h2>
☺ <p>深圳地处广东省南部。</p>
☺ <hr />
☺ <h2>历史简述</h2>
☺ <p>深圳于1979年建市，1980年设立经济特区。</p>
☺ <button onclick="delhr()">删除水平线</button>
```

制作者：人工智能学院 张士 博士

43

### 参考代码

```
☺ function delhr() {
☺ var hrs = document.querySelectorAll("hr");
☺ for (var i = 0; i <= hrs.length - 1; i++)
  ✧ {
    ✧ var hr = hrs[i];
    ✧ hr.parentNode.removeChild(hr);
  }
☺ }
```



制作者：人工智能学院 张士 博士

44

### 独立实践时间

☺任务：放大镜。

✧参见教材P165-167。

让我们一起动手吧！



制作者：人工智能学院 张士 博士

45