

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

- ☺ 事件与事件处理
- ☺ 键盘事件
- ☺ 鼠标事件
- ☺ 网页事件
- ☺ 表单事件

制作人：人工智能学院 张立 博士

2

事件与事件处理

制作人：人工智能学院 张立 博士

3

事件

- ☺ 概念
 - ✧ 用户进行交互时发生的操作
- ☺ 事件流
 - ✧ 冒泡事件：自下而上触发
 - ✧ 捕获事件：自上而下触发

制作人：人工智能学院 张立 博士

4

事件分类

用户动作

- 用户鼠标或键盘操作。
- 如click、keydown等。

状态变化

- 页面的状态发生变化。
- 如load、resize等。

制作人：人工智能学院 张立 博士

5

事件的分类

JavaScript事件

键盘事件 鼠标事件 页面事件 表单事件

制作人：人工智能学院 张立 博士

6

事件处理

- ☺ 概念
 - 为对象的事件绑定或者移除事件处理程序
- ☺ 结果
 - 当事件发生时就会触发相应的事件处理程序
- ☺ 两种方式
 - DOM 0级事件处理
 - DOM 2级事件处理

制作人：人工智能学院 张立 博士

7

DOM 0级事件处理

- ☺ 本质
 - ✧ 将事件处理与对象的事件关联
- ☺ 形式1：行内绑定
 - ✧ `<h2 onclick="console.log('被单击了');">前端开发</h2>`
- ☺ 形式2：动态处理
 - ✧ 动态绑定和动态移除两个过程

制作人：人工智能学院 张立 博士

8

动态绑定

- ☺ 动态绑定
 - ✧ 将对象的事件属性值赋值为事件处理函数
 - ✧ `mydiv.onclick=function(){……}`
- ☺ 动态移除
 - ✧ 将对象的事件属性设置为null
 - ✧ `mydiv.onclick=null;`

制作人：人工智能学院 张立 博士

9

DOM 2级事件处理

优势

✧ 同一个对象的同一个事件可绑定多个事件处理程序。

两个方法

✧ `addEventListener()`: 添加事件处理程序

✧ `removeEventListener()`: 移除事件处理程序



制作者: 人工智能学院 张立 博士

10

一般人我不告诉他

第三个参数是控制事件触发顺序的, 默认为false, 即为事件冒泡, 若为true, 即为事件捕获。

```
h2.addEventListener('click', myf1, false);
h2.addEventListener('click', myf1, true);
```



制作者: 人工智能学院 张立 博士

11

演示与实践

任务: 绑定事件的3种方式。

让我们一起动手吧!



制作者: 人工智能学院 张立 博士

12

参考代码

```
① <h2 onclick="console.log('被单击了');">样本
  </h2>
② <h2>样本</h2>
③ var h2=document.getElementsByTagName(h2)[1];
④ h2.onclick=function () { console.log('事件处理1'); }
⑤ h2.addEventListener('click', function () {
  ⑥   console.log('事件处理2'); }, false);
```



制作者: 人工智能学院 张立 博士

13

参考代码

```
① <h2>样本</h2>
② var h2=document.getElementsByTagName(h2)[1];
③ h2.addEventListener('click', myf, false);
④ function myf(){
  ⑤   console.log('事件处理3');
  ⑥ }
```



制作者: 人工智能学院 张立 博士

14

一般人我不告诉他

对象的同一个事件可绑定多个事件处理程序。

```
h2.addEventListener('click', myf1, false);
h2.addEventListener('click', myf2, false);
```



制作者: 人工智能学院 张立 博士

15

键盘事件

处理键盘事件

字符键

✧ 依次触发 `onkeydown`、`onkeypress`、`onkeyup` 事件。

非字符键 (功能键和控制键)

✧ 依次触发 `onkeydown`、`onkeyup` 事件



制作者: 人工智能学院 张立 博士

17

演示与实践

任务: `Alt`键与`a`键。

让我们一起动手吧!



制作者: 人工智能学院 张立 博士

18

参考代码

```

③ document.onkeydown = f;
③ document.onkeyup = f;
③ document.onkeypress = f;
③ function f(e) {
  ③ console.log(`${e.type}:Unicode值${e.keyCode}`);
  ③ }

```



制作者：人工智能学院 张立 博士

19

一般人我不告诉他

判定字符按键时需要选择keypress事件。

- ❑ keypress事件区分大小写
- ❑ keydown/keyup不区分大小写



制作者：人工智能学院 张立 博士

20

演示与实践

- ③ 任务：移动图片。
- ③ 提示：A/D/S/W按键

让我们一起动手吧！



制作者：人工智能学院 张立 博士

21

参考代码

```

③ document.onkeypress = function(e) {
  ③ if (e.keyCode == 119) // w
    ③ img.style.top = `${img.offsetTop + 10}px`;
  ③ else if (e.keyCode == 115) // s
    ③ img.style.top = `${img.offsetTop - 10}px`;
  ③ else if (e.keyCode == 97) // a
    ③ img.style.left = `${img.offsetLeft + 10}px`;
  ③ else if (e.keyCode == 100) // d
    ③ img.style.left = `${img.offsetLeft - 10}px`;
  ③ }

```

制作者：人工智能学院 张立 博士

22

演示与实践

- ③ 任务：缩放图片。
- ③ 提示：上下方向键

让我们一起动手吧！



制作者：人工智能学院 张立 博士

23

参考代码

```

③ <div id="img"></div>
③ var img = document.querySelector("#img");
③ var j = 1;
③ document.onkeyup = function(e) {
  ③ if (e.keyCode == 38) { // 按↑键放大
    ③ j = j + 0.1; img.style.transform = `scale(${j})`;
  ③ } else if (e.keyCode == 40) { // 按↓键缩小
    ③ j = j - 0.1; img.style.transform = `scale(${j})`;
  ③ }
}

```

制作者：人工智能学院 张立 博士

24

独立实践时间

- ③ 任务：移动缩放图片。

让我们一起动手吧！



制作者：人工智能学院 张立 博士

25

参考代码

```

③ <div id="img"></div>
③ var img = document.querySelector("#img");
③ var j = 1;
③ document.onkeypress = function(e) {
  ③ .....//ADSW键
  ③ }
③ document.onkeyup = function(e) {
  ③ .....//方向键
  ③ }

```



制作者：人工智能学院 张立 博士

26

作业-3

- ③ 任务：键盘事件处理。
- ③ 在网页中心位置显示4个字；
- ③ 按“a”键“s”键缩放字体大小。

让我们一起动手吧！



制作者：人工智能学院 张立 博士

27

演示与实践

- 任务：处理组合键。
- 提示：keydown事件

让我们一起动手吧！



制作者：人工智能学院 张士 博士

28

参考代码

```
document.onkeydown = function() {  
    if ((event.keyCode == 65) && (event.ctrlKey)) {  
        alert("按下了Ctrl和A键");  
        event.returnValue = false;  
    }  
}
```



制作者：人工智能学院 张士 博士

29

鼠标事件

制作者：人工智能学院 张士 博士

30

鼠标点击事件

- onclick (单击)、ondblclick (双击)、onmousedown (鼠标按下) 和 onmouseup (鼠标弹起) 事件。
- 鼠标点击事件有触发事件顺序。



制作者：人工智能学院 张士 博士

31

鼠标单击对象



制作者：人工智能学院 张士 博士

32

鼠标移动事件

- 鼠标经过事件
 - onmouseover
- 鼠标移动事件
 - onmousemove
- 鼠标离开事件
 - onmouseout



制作者：人工智能学院 张士 博士

33

演示与实践

- 任务：鼠标移入和移出。

让我们一起动手吧！



制作者：人工智能学院 张士 博士

34

参考代码

```
  
mouse.onmouseover = function() {  
    document.getElementById("mypic").src = "img/p2.gif";  
}  
mouse.onmouseout = function() {  
    document.getElementById("mypic").src = "img/p1.gif";  
}
```

制作者：人工智能学院 张士 博士

35

鼠标按键

- Event对象的button属性
 - 0: 左键被按下
 - 1: 中键被按下
 - 2: 右键被按下



制作者：人工智能学院 张士 博士

36

鼠标的位置

- ③ screenX和screenY
 - ❖ 以电脑屏幕左上角为原点
- ③ clientX和clientY
 - ❖ 以浏览器左上角为原点
- ③ offsetX和offsetY
 - ❖ 以当前事件的目标对象左上角为原点
- ③ pageX和pageY
 - ❖ 以Document对象（文本窗口）左上角为原点

制作者：人工智能学院 张士 博士

37

演示与实践

③任务：鼠标的位置。

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

```

var mypic = document.getElementById("mypic");
document.onmousemove = function(e) {
    var x = e.pageX;
    var y = e.pageY;
    mypic.style.left = x + 20 + "px";
    mypic.style.top = y - 20 + "px";
}
```



制作者：人工智能学院 张士 博士

39

网页事件

制作者：人工智能学院 张士 博士

40

处理页面事件

onload事件

- 当前HTML文档装载完成就会触发
- 用途：保存页面载入后立即执行的脚本

onresize事件

- 改变浏览器大小时触发
- 用途：用于固定浏览器的大小



制作者：人工智能学院 张士 博士

41

一般人我不告诉他

<body>元素的事件也可以绑定到Window对象，效果基本相同。

- ☐ document.body.onload=.....
- ☐ window.onload=.....



制作者：人工智能学院 张士 博士

42

演示与实践

③任务：网页事件。

让我们一起动手吧！



制作者：人工智能学院 张士 博士

43

参考代码

```
<div id="div1"></div>
var d=document.getElementById("div1");
window.onload=function() {
    d.style.backgroundColor="yellow";
}
window.onresize = function () {
    d.style.backgroundColor="red";
};
```



制作者：人工智能学院 张士 博士

44

表单事件

制作者：人工智能学院 张士 博士

45

操作表单对象

- ☉ 禁用和启用
 - ✧ disabled属性=true, 代表禁用
 - ✧ disabled属性=false, 代表启用
- ☉ 获得和失去焦点
 - ✧ focus()方法: 获得焦点
 - ✧ blur()方法: 失去焦点



制作者: 人工智能学院 张立 博士

46

演示与实践

- ☉ 任务: 禁用和启用表单对象。

让我们一起动手吧!



制作者: 人工智能学院 张立 博士

47

参考代码

```

☉ <form>
  ✧ 姓名: <input id="fname" type="text" />
  ✧ <input type="button" value="禁用" onclick="disable()" />
  ✧ <input type="button" value="启用" onclick="enable()" />
☉ </form>

```



制作者: 人工智能学院 张立 博士

48

参考代码

```

☉ function disable() {
  ✧ document.getElementById("fname").disabled = true;
☉ }
☉ function enable() {
  ✧ document.getElementById("fname").disabled = false;
☉ }

```



制作者: 人工智能学院 张立 博士

49

独立实践事件

- ☉ 任务: 表单对象获得和失去焦点。

让我们一起动手吧!



制作者: 人工智能学院 张立 博士

50

表单验证

本质

- 在用户提交表单前, 对用户输入的数据进行有效性验证。

用途

- 防止无效数据传送到服务器, 减轻服务器的数据处理负担, 从而提高系统性能。

制作者: 人工智能学院 张立 博士

51

一般人我不告诉他

使用JavaScript实现客户端验证, 减轻服务器的数据处理负担, 从而提高系统性能。



制作者: 人工智能学院 张立 博士

52

演示与实践

- ☉ 任务: 表单验证。

让我们一起动手吧!



制作者: 人工智能学院 张立 博士

53

参考代码

```

☉ <form name="fm" id="myform" method="get"
  action="login.php" onsubmit="return checkField()">
  ✧ <input type="text" name="txtUserName" size="20">
  ✧ <input type="submit" value="登录">
☉ </form>

```



制作者: 人工智能学院 张立 博士

54

参考代码

```
function checkFields()//是否为空
{
    if (fm.txtUserName.value == "") {
        * alert("用户名不能为空");
        * fm.txtUserName.focus();
        * return false;
    }
    * return true;
}
```



制作者：人工智能学院 张立 博士

55