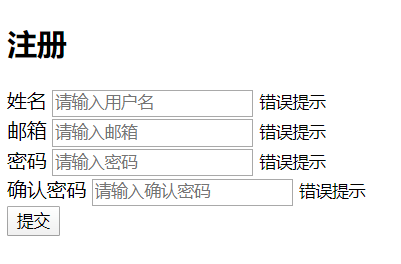
# 客户端表单验证demo

# 构建html



# 编辑css

box-sizing: border-box;

* 因为后续要添加padding,不希望padding影响到盒子的大小

display: flex;

* 设置成弹性布局, 因为整个container是一个居中的效果

min-height: 100vh;

* 覆盖整个浏览器视口高度

cursor: pointer;

* 鼠标移动到上面时变成一只手

outline: 0;

* 取消默认的边框

visibility: hidden;

* 隐藏内容, 区别于display: none; 前者视觉消失文档流中有占位, 后者文档直接不解析.

:root{

    --success-color: #2ecc71;

    --error-color: #e74c3c;

}

* 定义颜色, 使用var(--success-color)或者var(--error-color)赋予颜色

# js表单验证

  e.preventDefault();

* 阻止表单提交的默认事件

username.value

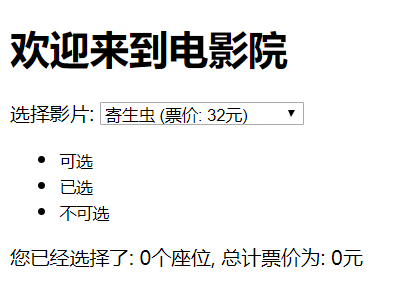
* 获取内容

.trim()

* 去掉首尾的空格

# MOVIE SEAT BOOKING

# 构建html



# 编辑css

盒模型模式(box-sizing): 标准盒模型(content-box)/怪异盒模型(border-box)

标准盒模型: 设置width的时候, 值是内容区域的宽度,想再设置border/padding, 盒子的实际宽度就会增加

怪异盒模型: 设置width的时候, 值就是盒子实际的宽度,border/padding不会影响实际的宽高.

display: flex;

* 通过弹性布局使内容处于中心位置

flex-direction: column;

* 设置弹性方向, 纵向排布

justify-content: center;

* 使内容纵向居中

align-items: center;

* 使内容横向居中

给selec标签设置padding时出现没反应的情况, 解决方法: 添加appearance属性

-moz-appearance: none;

     -webkit-appearance: none;

     appearance: none;

:nth-of-type(n)

:nth-last-of-type(n)

* 匹配同类型中的(第n个)/(倒数第n个)同级兄弟元素。

.seat:not(.occupied):hover{

    cursor: pointer;

    transform: scale(1.2);

}

* :not( ) 表示seat类中没有occupied类的属性, **“它的作用是防止特定的元素被选中”---MDN**
* transform: scale( n ) 表示放大n倍

list-style-type: none;

* 取消li中的默认小圆点

transform: rotateX(-45deg);

* “定义沿着 X 轴的 3D 旋转。”---W3C

perspective: 1000px;

* 视距1000px <https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/perspective>

# js

EventTarget.addEventListener()

* <https://developer.mozilla.org/zh-CN/docs/Web/API/EventTarget/addEventListener>

container.addEventListener("click", e => {

    ……

})

movieSelect.addEventListener('change', e => {

……

})

**classList.add( newClassName )；**

**添加新的类名，如已经存在，取消添加**

**classList.contains( oldClassName );**

**确定元素中是否包含指定的类名，返回值为true 、false；**

**classList.remove( oldClassName )；**

**移除已经存在的类名;**

**classList.toggle( className )；**

**如果classList中存在给定的值，删除它，否则，添加它；**

**classList.replace( oldClassName，newClassName )；**

**类名替换**

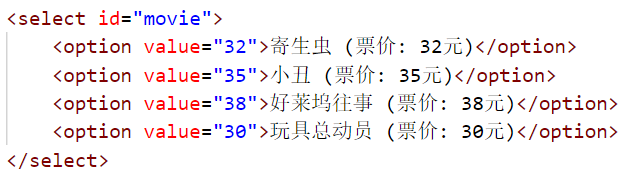
container.addEventListener("click", e => {

    if (e.target.classList.contains("seat") && !e.target.classList.contains("occupied")){

        e.target.classList.toggle("selected");

    }

})

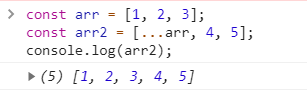
对于select标签

使用 .selectedIndex获取对应的索引值

#LocalStorage本地存储

展开语法

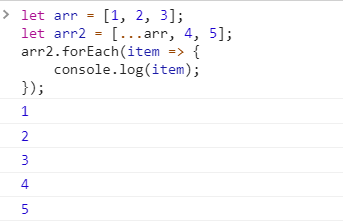
字面量数组构造或字符串 :



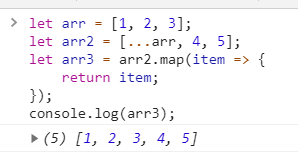
还有函数调用和构造字面量对象时, 进行克隆或者属性拷贝, 详情参考MDN

* <https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/Spread_syntax>

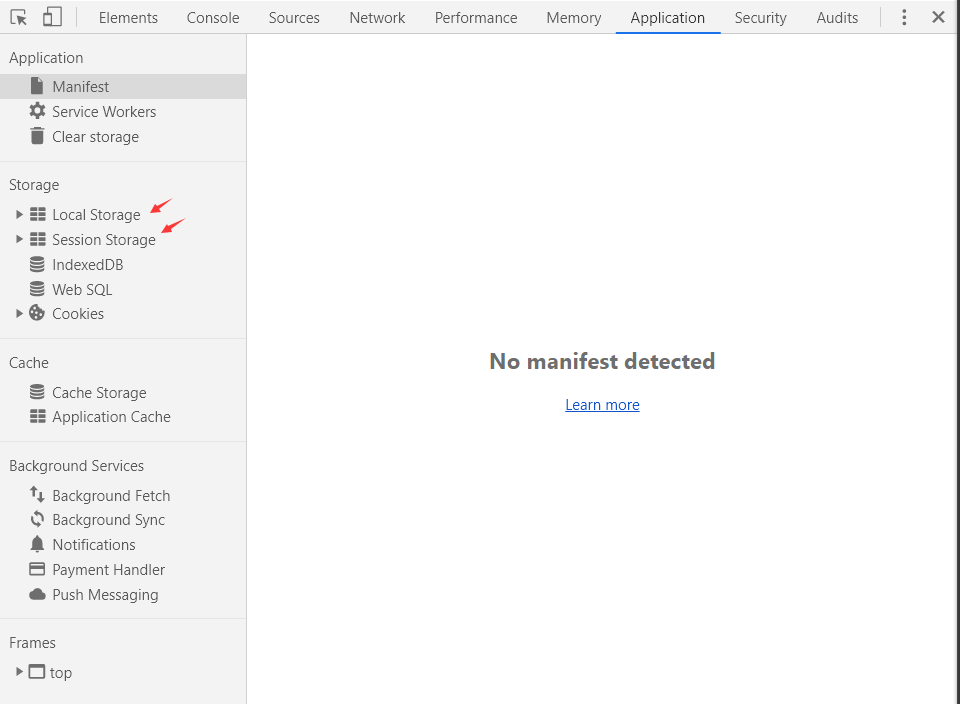
forEach



map (会返回一个新的数组)



开始使用LocalStorage, 关于LocalStorage的说明 :

 ! 数据内容存储在浏览器中

! 值类型限定为string, 通常使用json进行转换

! LocalStorage属于永久性的存储(不主动删除则一直存在), 而sessionstorage在会话结束/关闭页面时, 里面的键值对内容就会被清空

localStorage.setItem('selectedSeats', JSON.stringify(seatsIndex));

* 此方法是添加localStorage的数据
* selectedSeats 是键, JSON.stringify(seatsIndex)是值
* 如果值是数组, 则如上使用JSON.stringify转化成字符串

JSON.parse(localSrotage.getItem("selectedSeats"));

* 此方法是通过”键”获取本地数据”值”
* JSON.parse字符串转化成数组 (如有需要就转化)

selectedSeats !== null && selectedSeats.length > 0

* 记得要判断数组不为空 且 长度大于0

# Custom Video Player

#html

**视频和音频API**

HTML5提供了用于在文档中嵌入富媒体的元素 — [<video>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/video)和[<audio>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/audio) — 这些元素通过自带的API来控制视频或音频的播放，定位进度等。本文将向你展示如何执行一些常见的任务，如创建自定义播放控件。 --MDN

* <https://developer.mozilla.org/zh-CN/docs/Learn/JavaScript/Client-side_web_APIs/Video_and_audio_APIs>

引入字体图标 fontawesomecdn

* <https://www.bootstrapcdn.com/fontawesome/>

关于video标签

* poster属性 – 设置默认图片(填图片路径)
* controls属性 (无) 🡪(有)



#css

@media(max-width: 800px){

    .screen,

    .controls{

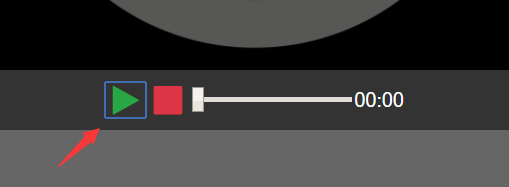
        width: 90%;

    }

}

* 屏幕宽度小于800px时, 播放屏幕和控件宽度变成90%

关于点击是出现蓝色边框线的问题



* :focus{outline: 0}可以解决

🡪

.btn:focus{

    outline: 0;

}

#js

Video.currentTime

🡪 对应视频当前时间

Video.duration

🡪 视频总时长

Math.floor

* 向下取整 例: Math.floor(66/60) = 1, Math.floor(59/60) = 0

# 汇率计算器

#html



#css

justify-content: space-between;

* flex布局 justify-content：space-between; 解决最后一排数量不够自动向两端排列问题 --CSDN 🡪 <https://blog.csdn.net/lucky___star/article/details/87876830>
* --RUNOOB 🡪 <https://www.runoob.com/try/playit.php?f=playcss_justify-content&preval=space-between>

#js

使用第三方的api接口

**fetch**

* fetchAPI为我们提供了JavaScript的接口, 用于访问和操纵HTTP管道的一些具体部分, 例如请求和响应, 它还提供了一个全局的fetch方法, 这种方法简单合理地跨网络异步获取资源, fetch不是Ajax的进一步封装, 它是原生的js.

**http的请求方法**

* Get请求: 只是加载页面/从服务器获取数据的时候
* Post请求: 向服务器提交被处理的数据
* Put请求: 要服务器上的内容更新/替换的时候
* Delete请求: 要在服务器上删除内容的时候
* …

(number).toFixed(2)

* 保留两位小数

# DOMArrayMethods

#html



#css

font-weight

* font-weight: *bold*;
* font-weight: 300;
* 该属性用于设置显示元素的文本中所用的字体加粗。数字值 400 相当于 关键字 normal，700 等价于 bold。每个数字值对应的字体加粗必须至少与下一个最小数字一样细，而且至少与下一个最大数字一样粗。

#js

**async** **function**

* **async function** 用来定义一个返回 [AsyncFunction](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/AsyncFunction) 对象的异步函数。异步函数是指通过事件循环异步执行的函数，它会通过一个隐式的 [Promise](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Promise) 返回其结果。如果你在代码中使用了异步函数，就会发现它的语法和结构会更像是标准的同步函数
* 异步函数可以包含[await](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/await)指令，该指令会暂停异步函数的执行，并等待Promise执行，然后继续执行异步函数，并返回结果。
* 记住，await 关键字只在异步函数内有效。如果你在异步函数外使用它，会抛出语法错误。

**function** updateDOM(*providedData* = *data*){

}

* 如果updataDOM函数没有传出值, 那么默认等于data, 如果传入了值, 那就是等于传入的值.

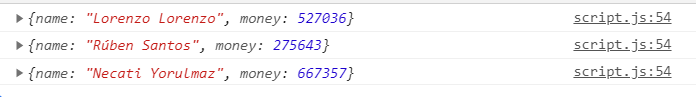
Array.prototype.map

* **map()** 方法**创建一个新数组**，其结果是该数组中的每个元素都调用一个提供的函数后返回的结果。

*data* =*data*.map(*user* **=>** {

        return {...*user*, *money*:*user*.*money* \* 2}

    });

* user 🡪

var obj1 = { foo: 'bar', x: 42 };

var obj2 = { foo: 'baz', y: 13 };

var clonedObj = { ...obj1 };

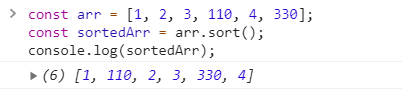
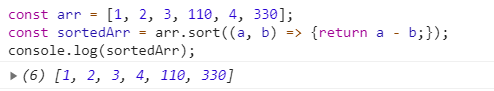
// 克隆后的对象: { foo: "bar", x: 42 }

var mergedObj = { ...obj1, ...obj2 };

// 合并后的对象: { foo: "baz", x: 42, y: 13 }

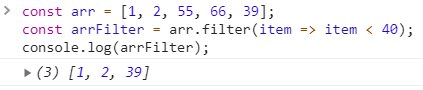
* 合并后, 若有相同键, 后者覆盖前者

Array.prototype.sort(compareFunction)

* 
* 
* 如果 compareFunction(a, b) 小于 0 ，那么 a 会被排列到 b 之前；
* 如果 compareFunction(a, b) 等于 0 ， a 和 b 的相对位置不变。备注： ECMAScript 标准并不保证这一行为，而且也不是所有浏览器都会遵守（例如 Mozilla 在 2003 年之前的版本）；
* 如果 compareFunction(a, b) 大于 0 ， b 会被排列到 a 之前。
* compareFunction(a, b) 必须总是对相同的输入返回相同的比较结果，否则排序的结果将是不确定的。

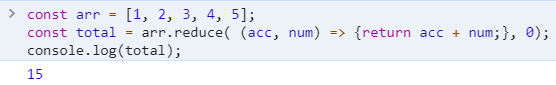
Array.prototype.filter( )

**filter()** 方法创建一个新数组, 其包含通过所提供函数实现的测试的所有元素。

* 

Array.prototype.reduce( )

**reduce()** 方法对数组中的每个元素执行一个由您提供的**reducer**函数(升序执行)，将其结果汇总为单个返回值。

* 

acc 🡪 累计器,

num 🡪 当前值,

0 🡪 默认值(作为第一次调用 callback函数时的第一个参数的值。 如果没有提供初始值，则将使用数组中的第一个元素。 在没有初始值的空数组上调用 reduce 将报错。)

# 折叠侧边栏

#html

引入bootstrapCDN ,使用bootstrap的字体图标

* <https://www.bootstrapcdn.com/fontawesome>

**HTML <nav>元素**表示页面的一部分，其目的是在当前文档或其他文档中提供导航链接。导航部分的常见示例是菜单，目录和索引。

<i class="fa fa-bars fa-2x"></i>

* i标签, 设置字体图标

        <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Atque deleniti sequi perspiciatis, saepe repudiandae suscipit.</p>

lorem15 enter

* 随机生成15个单词

transform: translateX(-100%);

* 左移100%

text-decoration: *underline*;

* 下划线特效

nav ul li:first-of-type

* [CSS](https://developer.mozilla.org/zh-CN/docs/Web/CSS) [伪类](https://developer.mozilla.org/zh-CN/docs/Web/CSS/Pseudo-classes) **:first-of-type**表示一组兄弟元素中其类型的第一个元素。

list-style-type: *none*;

* 清除ul的默认样式

子级绝对定位position: *absolute*; -- 父级绝对定位position: *relative*;

    position: *absolute*;

    top: 50%;

    left: 50%;

transform: translate(-50%, -50%);

* 绝对定位下的居中

@keyframes

* @keyframes 规则通过在动画序列中定义关键帧（或waypoints）的样式来控制CSS动画序列中的中间步骤。这比[转换](https://developer.mozilla.org/zh-CN/docs/Web/CSS/CSS_Transitions)更能控制动画序列的中间步骤。

transition: transform 0.3s *ease*;

* **transition**[CSS](https://developer.mozilla.org/en/CSS) 属性是 [transition-property](https://developer.mozilla.org/zh-CN/docs/Web/CSS/transition-property)，[transition-duration](https://developer.mozilla.org/zh-CN/docs/Web/CSS/transition-duration)，[transition-timing-function](https://developer.mozilla.org/zh-CN/docs/Web/CSS/transition-timing-function) 和 [transition-delay](https://developer.mozilla.org/zh-CN/docs/Web/CSS/transition-delay) 的一个[简写属性](https://developer.mozilla.org/en-US/docs/CSS/Shorthand_properties)。

#js