



Școala  
informală  
de IT

# Movie tickets - Workshop



















































# Agenda




- Create HTML
- Style using CSS
- Using JS create the app functionality for:
  - select movies
  - select seats
  - update the price
  - show no of tickets
  - show total price

Select a movie: Rocky (\$10)

Number of tickets: 0

Total Price: \$0

 Unoccupied  Selected  Occupied

# HTML



# Task 1 – Movie Container

Select a movie: Rocky (\$10)

- Create a div of class “movie-container”
- Create a label “Select a movie”
- Create a select containing some movies and prices associated(use value attribute for setting the price)

## Task 1 - Solution

```
<div class="movie-container">
  <label>Select a movie:</label>
  <select id="movie">
    <option value="10">Rocky ($10)</option>
    <option value="12">Joker ($12)</option>
    <option value="8">Home Alone ($8)</option>
    <option value="9">The Lord of the rings ($9)</option>
  </select>
</div>
```

## Task 2 – Number of tickets and Total Price

- Create 2 paragraph of class “text”
- Add to spans:
  1. id=“count”
  2. id =“total”

Number of tickets:0

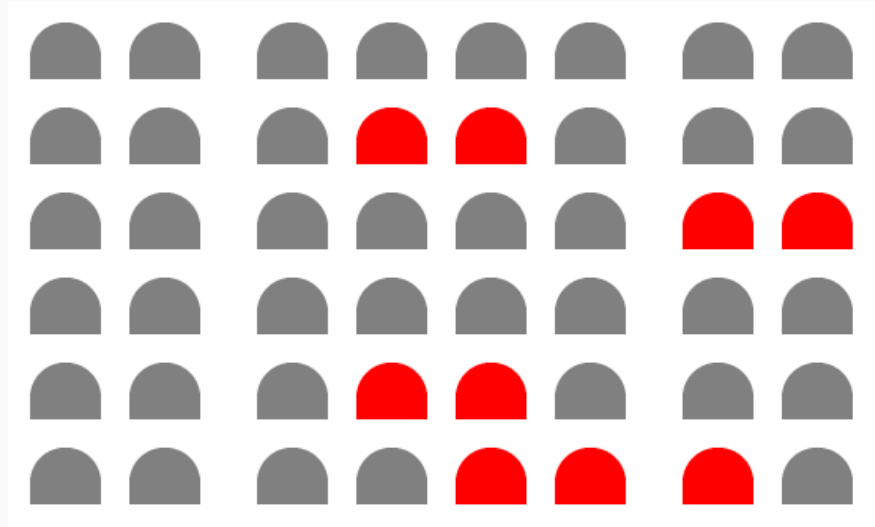
Total Price: \$0

## Task 2 – Solution

```
<p class="text">  
|   Number of tickets: <span id="count">0</span> <br>  
</p>  
<p class="text">  
|   Total Price: $<span id="total">0</span>  
</p>
```

## Task 3 – Seats container

- Create a div of class “container” and inside create 6 div of class “row”
- Each div of class “row” must contain 8 divs of class “seat”, and only some of them should be also of class “occupied”





## Task 3 - Solution

```
<div class="container">
  <div class="row">
    <div class="seat"></div>
    <div class="seat"></div>
    <div class="seat"></div>
    <div class="seat"></div>
    <div class="seat"></div>
    <div class="seat"></div>
    <div class="seat"></div>
  </div>
  <div class="row">
    <div class="seat"></div>
    <div class="seat"></div>
    <div class="seat"></div>
    <div class="seat occupied"></div>
    <div class="seat occupied"></div>
    <div class="seat"></div>
    <div class="seat"></div>
    <div class="seat"></div>
  </div>
</div>
```

[illegible]

```
<div class="row">
  <div class="seat"></div>
  <div class="seat"></div>
  <div class="seat"></div>
  <div class="seat occupied"></div>
  <div class="seat occupied"></div>
  <div class="seat"></div>
  <div class="seat"></div>
  <div class="seat"></div>
</div>
<div class="row">
  <div class="seat"></div>
  <div class="seat"></div>
  <div class="seat"></div>
  <div class="seat"></div>
  <div class="seat occupied"></div>
  <div class="seat occupied"></div>
  <div class="seat occupied"></div>
  <div class="seat"></div>
</div>
</div>
```

## Task 4 – Legend Container



- Create an unordered list containing 3 li
- First li: class “unoccupied”
- Second li: class “selected”
- Third li: class “occupied”

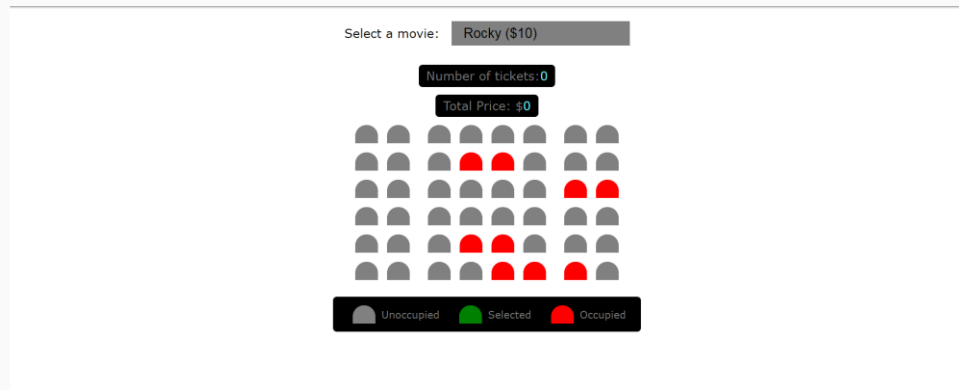
Hint: use `<small>` tag for the text

# CSS



# Task 1 – Creating the layout

- **Reset border-sizing to border-box for all html elements**
- **Use display for flex for center the content**
- **Use other styling values and properties to create the layout from the picture attached**



# Task 1 – Solution

```
* {  
  box-sizing: border-box;  
}  
  
body {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: center;  
  height: 100%;  
  font-family: 'Verdana', sans-serif;  
  margin: 0;  
}
```

## Task 2 – Styling Movie Container

Select a movie:


Rocky (\$10)

- **Margin**
- **Padding**
- **Background color**
- **Font size**

2018 - Front-End Web Development Demo

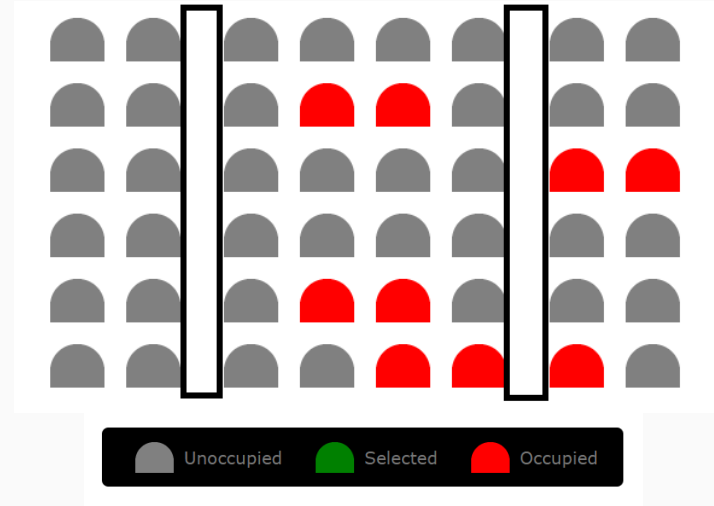


## Task 2 - Solution

```
.movie-container {  
  margin: 20px 0;  
}  
  
.movie-container select {  
  background-color:  gray;  
  font-size: 18px;  
  margin-left: 10px;  
  padding: 5px 15px 5px 15px;  
  appearance: none;  
}
```




## Task 3 – Styling the seats, legend and text

- Add background colors for different class of seats
- Add borders padding, height, width, margins, border-radius
- Use specific selector for adding space between the rows or different colors for occupied/unoccupied








## Task 3 – Solution

```
.seat {  
  background-color:  gray;  
  height: 24px;  
  width: 30px;  
  margin: 6px;  
  border-top-left-radius: 20px;  
  border-top-right-radius: 20px;  
}  
  
.seat.selected {  
  background-color:  green;  
}  
  
.seat.occupied {  
  background-color:  red;  
}
```

```
.seat:nth-of-type(2) {  
  margin-right: 18px;  
}  
  
.seat:nth-last-of-type(2) {  
  margin-left: 18px;  
}  
  
.seat:not(.occupied):hover {  
  cursor: pointer;  
  transform: scale(1.2);  
}  
  
.legend .seat:not(.occupied):hover {  
  cursor: default;  
  transform: scale(1);  
}
```

## Task 3 – Solution 2

```
.legend, .text {  
  background:  black;  
  padding: 5px 10px;  
  border-radius: 5px;  
  color:  #777;  
  list-style-type: none;  
  display: flex;  
  justify-content: space-between;  
}  
  
.legend li {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  margin: 0 10px;  
}
```

```
.legend li small {  
  margin-left: 2px;  
}  
  
.row {  
  display: flex;  
}  
  
p.text {  
  margin: 5px 0;  
}  
  
p.text span {  
  color:  #6feaf6;  
}
```

# JavaScript



# Task 1 – Creating global variables

- Create the following global variables using `const` keyword instead of `var`:
  - `container`
  - `seats`
  - `count`
  - `total`
  - `movieSelect`
  - `ticketPrice`(using `let`)

# Task 1 – Solution

```
const container = document.querySelector('.container');  
const seats = document.querySelectorAll('.row .seat:not(.occupied)');  
const count = document.getElementById('count');  
const total = document.getElementById('total');  
const movieSelect = document.getElementById('movie');  
let ticketPrice = Number(movieSelect.value);
```

## Task 2 – Update prices and tickets count function

- Create a function called “updateSelectedCount” used for
  - query all selected seats
  - count selected seats
  - modify the values from informational paragraphs  
(count, total)

## Task 2 – Solution

```
✓ function updateSelectedCount() {  
    const selectedSeats = document.querySelectorAll('.row .seat.selected');  
  
    const selectedSeatsCount = selectedSeats.length;  
  
    count.innerText = selectedSeatsCount;  
    total.innerText = selectedSeatsCount * ticketPrice;  
}
```

## Task 2 – Create event listeners

- **Create 2 event listeners for**
  - **seat click event**
  - **movie set event**
  - **update the count and total prices when it's needed**
  -



## Task 2 – Solution

```
// Movie select event
movieSelect.addEventListener('change', function(e) {
    ticketPrice = Number(e.target.value);
    updateSelectedCount();
});

// Seat click event
container.addEventListener('click', function(e) {
    if (
        e.target.classList.contains('seat') &&
        !e.target.classList.contains('occupied')
    ) {
        e.target.classList.toggle('selected');

        updateSelectedCount();
    }
});
```