

This is some pretty straight-forward CSS to help the elements mimic the website's style when the existing SquareSpace CSS wouldn't cover it. It might have been clever to try and name the HTML elements after HTML elements created by SquareSpace in the hopes that they would inherit the styling, but that also sounded hard.

Pure HTML, just a paragraph with some links

This is the Javascript. It's almost all handled within one custom function that gets called when the button is clicked.

The first step declares a few necessary variables. The second step fills one of those variables, an array, with the values attached to the days of the week from HTML form in the next section. This gives us the base rent, and the number of days desired.

The next section then takes the number of days and multiplies it with a calculated estimate of fees per day.

After that, a series of *if* statements and a *switch... case* statement (which is like multiple *if* statements) determine which areas of the theatre are being looked at, and adjusts the total accordingly. Finally the total value is calculated and can be displayed to the user.

There is also a second function, but it was not something that I wrote. I am not 100% certain how it works, but for our uses, it sums and rounds the numbers in the array.

This is the HTML form that feeds data into the Javascript program. Pardon the inline CSS styling.

The first chunk defines the radio buttons that allow the type of event to be chosen. Because they are grouped, they are exclusive to one another.

The next section defines a multi-select list of days. Usage isn't super intuitive for some people so, there are some additional instructions.

The third section defines a set of check boxes, which are not exclusive. They allow users to select several items at once from the group, in this case areas of theatre available for rent.

The last two items are the button that calls the Javascript function to do the calculation, and a paragraph object that shows the calculated value.

```

<html>
<body>
<style>
.selectLabel {
display: inline-block;
float: left;
width: 250px;
margin-right: 10px;
}
.btnBox {
width: 200px;
padding: 5px;
text-align: center;
color: white;
background-color: #992013;
border-radius: 5px;
}
.wline {
display: inline;
float: left;
background: black;
width: 1px;
height: 60px;
margin: 15px 20px;
}
</style>
<div style="margin-left: 35px;">
<p>Quick Estimate</p>
<p>The generated value should not be understood as a quote. It is meant only as a tool to help potential renters estimate the cost of renting the Morton Theatre. To get an actual quote and begin the rental process, please fill out a <a href="https://www.mortontheatre.com/spaces">Space Reservation Form</a> and submit the application fee. For more information please reach out to us at <a href="mailto:morton.theatre@ccogov.com">Morton Theatre</a>.</p>
<p id="demo"></p>
<script>
function submitForm() {
var selected = [];
var spaces = 0; //can't be null

//this loop fills an array with the values of the selected days in the multiple, it's handy because we get both the number of days and the value of the rent
for (var option of document.getElementById("days").options) {
if (option.selected) {
selected.push(option.value);
}
}

//evaluates which radio button is selected and then multiplies the number of selected days * the daily rates
if (document.getElementById("dance").checked == true) {
dailyRt = selected.length * 800;
dailyLs = selected.length * 300;
}
else if (document.getElementById("small").checked == true) {
dailyRt = selected.length * 900;
dailyLs = selected.length * 300;
}
else if (document.getElementById("large").checked == true) {
dailyRt = selected.length * 1400;
dailyLs = selected.length * 600;
}
else if (document.getElementById("private").checked == true) {
dailyRt = selected.length * 440;
dailyLs = selected.length * 375;
}

let rent = selected.reduce(getDom, 0); //getDom is kinda weird to me, but I haven't looked at the source needs the 0 or it doesn't iterate

//these if statements evaluate which check boxes are ticked for space and add 1, 2, or 4 to get a value of 0-7 that uniquely identifies the selection
if (document.getElementById("auditorium").checked == true) {
spaces += 1;
}
if (document.getElementById("pharmacy").checked == true) {
spaces += 2;
}
if (document.getElementById("conference").checked == true) {
spaces += 4;
}

//the switch then evaluates that unique number and adjusts rent accordingly.
//Cases:
//0 is no selection: ignore and run like an auditorium only
//1 is just auditorium: default
//2 is just pharmacy: rent is 60% of total
//3 is auditorium and pharmacy: add $100
//4 is just conference: rent is $100
//5 is auditorium and conference: $30/day
//6 is pharmacy and conference: rent is 60% of total + $30/day
//7 is all the above: add $100 and $30/day
switch (spaces) {
case 0:
//do nothing
break;
case 1:
//do nothing
break;
case 2:
rent += 100;
break;
case 3:
rent = selected.length * 100;
break;
case 4:
rent = selected.length * 100;
break;
case 5:
rent = selected.length * 30;
break;
case 6:
rent = selected.length * 120;
break;
case 7:
rent = selected.length * 120;
break;
}

//writes the final values to the screen
document.getElementById("total").innerHTML = "$" + Number(dailyRt + rent) + " + " + Number(dailyLs + rent);
}

function getDom(total, num) {
return Number(total) + Number(num); //does not iterate if num isn't also passed to this function
}
</script>
<div style="margin-left: 35px;">
<form>
<input type="radio" id="dance" name="event_type" checked="checked" /> Dance
<input type="radio" id="small" name="event_type" /> Small Theatrical Event
<input type="radio" id="large" name="event_type" /> Large Theatrical Event
<input type="radio" id="private" name="event_type" /> Private Event
</form>
<div style="margin-left: 35px;">
<form>
<label class="selectLabel" for="days">Select the days of the week you will be in the theatre. Hold down the Ctrl (Windows) / Command (Mac) button to select multiple options.</label>
<select id="days" name="days" multiple>
<option value="575">Monday</option>
<option value="575">Tuesday</option>
<option value="575">Wednesday</option>
<option value="770">Thursday</option>
<option value="770">Friday</option>
<option value="770">Saturday</option>
<option value="770">Sunday</option>
</select>
</div>
<div style="margin-left: 35px; margin-top: 8px;">
<form id="areas">
<label class="selectLabel">Select the areas of the theatre you would like to rent.</label>
<input type="checkbox" id="auditorium" /> Auditorium
<input type="checkbox" id="pharmacy" /> Pharmacy
<input type="checkbox" id="conference" /> Conference Room
</form>
<div style="margin-left: 35px;">
<button style="display: inline-block;" id="submit" onclick="submitForm()">Get Quote</button>
<p id="rent" class="btnBox" style="display: inline-block; margin-left: 15px;"></p>
</body>
</html>

```

A →
B →
C →
D →
E →
F →
G →
H →
I →
J →
K →

A I'm fairly confident this line is just a relic from a previous version. Despite its id, I don't think it actually does anything, and instead mostly just functions as a complicated `
` tag. That said, I haven't removed it, or tried the program without it.

B Let's break this statement down a little: The `for` loop is first asking how many possible options are there within the element `days`, which is the HTML form object that lists the number of days and allows people to select multiple. It's defined in the HTML form section. The answer is going to be 7 because we use a Gregorian calendar, which means that the code under that first line will run for 7 times, once for each option, one at a time, in order. The code that's running is asking each item if it is selected, and if it is, it's adding its value as an element to the array `selected`. The value is determined in the HTML form where the object is defined. Those values correspond to weekday and weekend rent prices.

C It's a series of `if` and `else if` statements that ask each radio button in turn if it is "true" or in other words, selected. The radio buttons are defined in the HTML form section. When the radio button objects are created, the `dance` item is automatically selected meaning that even if the user doesn't interact with them, one of them is always selected, and one of these `if` statements will always evaluate to true. When that happens two variables, `dailyHi` and `dailyLo`, are set to the number of days selected multiplied by a calculated value. The days selected is determined by checking the length of the `selected` array, or put another way, how many elements it contains. The calculated values came from average daily prices for comparable events, minus rent.

D This line sets the variable `rent` to the combined value of all the elements in the `selected` array. It uses the `reduce()` method which is kind of confusing to me. (https://www.w3schools.com/jsref/jsref_reduce.asp) Truthfully, the `getSum` function, and the way this is set up is likely not the best way to do it, but it seems to do what we want it to do, so I just left it alone. That W3 Schools link is probably where the snippet of code came from, and I suspect I just didn't understand something, but played with it until it worked and then left it at that. It's probably not too hard to improve, but it's also not really hurting anything, and it doesn't do anything that needs to be adjusted, so...

E If you're bored one day, take the numbers 1, 2, and 4 and add them up to make each number 1-7. If you write down what you added to get those sums you might notice a pattern, or even a few. For our purposes, each of those sums can be described as a unique collection of addends and collectively they are comprehensive, meaning 1-7 describes all the possible combinations of 1, 2, and 4 without ambiguity or overlap. Binary math is neat. In this case, we are observing the check boxes defined in the HTML form, and if they are "true" or checked, then their value (1, 2, or 4) is added to a variable called `spaces`.

F `Switch...case` statements are very similar to `if` statements, but basically it's taking a variable (or expression), and asking, "Does it match this case?" Our `switch...case` is taking the `spaces` variable we filled in step E and asking, "What number, 0-7, does it match?" It then adjusts the rent to more accurately reflect reality. The comments do a good job of explaining the different cases. Notice that whenever the rent is multiplied by a decimal number it's rounded to the nearest integer.

You'll also notice a few operators like `+=` and `*=`. Those are special operators that take the value of a variable, perform some arithmetic and then reassign that new value to the same variable name. For example we'll say that if `rent = 100`, then `rent += 100` would make `rent = 200`. Similarly, if `rent = 100`, then `rent *= 0.5` would make `rent = 50`. Note that all expressions on the right side of those operators are evaluated first. If `r = 100`, and `s = 5`, then `r += s * 2` will evaluate to 110, not 210.

Finally, `selected.length` is again, the length of the `selected` array, or the number of elements in it, which corresponds to the number of days selected on the HTML form. Also, `break` is just a keyword that ends a `case`.

G This line finds the HTML element defined as having the ID `head`. In this case, that's a paragraph with some CSS styling that is defined just after the HTML form. It then sets the text of that paragraph to be a formatted string. It appends a \$ to the front, calculates and displays the sum of our calculated values `dailyLo` (Step C) and `rent` (Steps B & F), appends a -, and then calculates and displays the sum of `dailyHi` and `rent`. It should look something like `$200-500`.

H More information can be found under Step D. This method likely does next to nothing, but I'm not really in place to be messing around with it here and now. It isn't actually doing nothing, because I think everything breaks if you delete it, and not just because of the method. If you have the time and the inclination, this might be a "fun" project.

I The HTML form section in general is relatively straightforward if you're comfortable in HTML. You might not know all the options going on, but at least they make an amount of surface level sense. I'm calling this section out in particular to note the values attached to the days of the week. It's a handy feature that select boxes let you assign a `value` to each option. In this case, the values are literally the rent. I forget how I calculated the numbers, but you are probably more up-to-date on the Morton's rental rates, and they might make immediate sense to you. It's also important to note that, by design, the rental rate, and the labor/extra rates were kept separate. `rent` is generally a flat rate multiplied by the number of days selected and then adjusted based on what areas are being rented (Step F), whereas labor/extra is more variable and is calculated as `dailyLo`, and `dailyHi` to give a range (Step C).

J This is the "Get Quote" button. When it is clicked, it calls the `submitFunc()` which is the bulk of our Javascript.

K This is where the calculated quote gets displayed. Note the ID `head`. Step G is where this gets modified when the quote is completed and displayed. Despite being a unique class, I still added more inline styling. Sorry.