

# ASSIGNMENT 7: JADE DELIGHT

Sejal Dua, Tufts University

03/02/2020

**URL:** [https://sejaldua.com/web-programming/hw7/jade\\_delight.html](https://sejaldua.com/web-programming/hw7/jade_delight.html)

## Specification

Task: Add validation (using Javascript) to the provided HTML form called `jade_delight.html`.

1. When a user selects a quantity:

- The amount should automatically be filled in to the corresponding total cost box.
- The subtotal, tax and total should update and be filled in.

2. When the user clicks submit:

- Verify that the last name and phone are entered. The phone should be a valid phone number.
- If the order was pickup, calculate the time 15 min from the order time.
- If the order was delivery, calculate the time 30 min from the order time for expected delivery time.
- If the order was delivery, the street and city are required.
- If there are validation issues, indicate the issue to the user.
- If there are no validation issues, display a message that thanks the user for their order, displays the pickup or delivery time and the total amount for the order.

### Listing 1: HTML and Javascript code for Jade Delight

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www↵
    .w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Jade Delight</title>
6 <link rel="stylesheet" href="style.css">
7 </head>
8 <body>
9 <script language="javascript">
```

---

```
1 function MenuItem(name, cost)
2 {
3     this.name = name;
4     this.cost=cost;
5 }
6
7 menuItems = new Array(
8     new MenuItem("Chicken Chop Suey", 4.5),
9     new MenuItem("Sweet and Sour Pork", 6.25),
10    new MenuItem("Shrimp Lo Mein", 5.25),
11    new MenuItem("Moo Shi Chicken", 6.5),
12    new MenuItem("Fried Rice", 2.35)
13 );
14
15 function updateTotals()
16 {
17     /* update cost values */
18     var num_items;
19     var subtotal = 0;
20     for (i = 0; i < menuItems.length; i++) {
21         num_items = document.getElementsByName("quan" + i)[0].value;
22         if (parseInt(num_items) != 0) {
23             item_cost = (parseInt(num_items) * menuItems[i].cost).toFixed(
24                 2);
25             document.getElementsByName("cost")[i].value = item_cost;
26             subtotal += parseFloat(item_cost);
27         }
28     }
29     /* update subtotal, tax, and total */
30     subtotal = subtotal;
31     tax = (subtotal * 0.0625);
32     total = subtotal + tax;
33     document.getElementById("subtotal").value = subtotal.toFixed(2);
34     document.getElementById("tax").value = tax.toFixed(2);
35     document.getElementById("total").value = total.toFixed(2);
36 }
37
38 function validate_name(name)
39 {
40     /* alphabetical regular expression pattern */
41     reg_name = /^[A-Za-z]+$\/;
42     return reg_name.test(name);
43 }
44
45 function validateLastName()
```

```

46 {
47     /* remove spaces from string */
48     var last_name = document.getElementsByName('lname')[0];
49     var error_msg_lname = document.getElementById("error_lname");
50
51     /* empty last name field */
52     if (last_name.value == "") {
53         error_msg_lname.innerHTML = " Please enter last name";
54         last_name.focus();
55         return false;
56     }
57     else
58         error_msg_lname.innerHTML = "";
59
60     /* non-alphabetic last name field */
61     lname = last_name.value.replace(/\s+/g, '');
62     alpha_name = validate_name(lname);
63     if (!alpha_name) {
64         error_msg_lname.innerHTML = " The name should contain only ↵
        alphabetic letters";
65         last_name.focus();
66         return false;
67     }
68     else
69         error_msg_lname.innerHTML = "";
70
71     return true;
72 }
73
74 function validatePhone()
75 {
76     var phone = document.getElementsByName('phone')[0];
77     var error_msg_phone = document.getElementById("error_phone");
78
79     /* check for phone field */
80     if (phone.value == "") {
81         error_msg_phone.innerHTML = " Please enter a phone number";
82         phone.focus();
83         return false;
84     }
85     else
86         error_msg_phone.innerHTML = "";
87
88     /* check if phone number matches regex pattern*/
89     var reg1 = /^[0-9]{3}-[0-9]{3}-[0-9]{4}$/;
90     var reg2 = /^[([0-9]{3})]-[0-9]{3}-[0-9]{4}$/;
91     if (!(reg1.test(phone.value) || (reg2.test(phone.value)))) {

```

```

92     error_msg_phone.innerHTML = " Phone number is invalid. Try: xxx-↵
        xxx-xxxx";
93     phone.focus();
94     return false;
95 }
96 else
97     error_msg_phone.innerHTML = "";
98
99     return true;
100 }
101
102 function validateAddress(id, error_id)
103 {
104     /* remove spaces from string */
105     var field = document.getElementsByName(id)[0];
106     var error_msg = document.getElementById(error_id);
107
108     /* check for empty field */
109     if (field.value == "") {
110         error_msg.innerHTML = " Please enter a " + id;
111         field.focus();
112         return false;
113     }
114     else
115         error_msg.innerHTML = "";
116
117     return true;
118 }
119
120 /* street and city are required for deliveries, add asterisk */
121 function isDelivery()
122 {
123     document.getElementById("star1").innerHTML = "*";
124     document.getElementById("star2").innerHTML = "*";
125 }
126
127 /* street and city are not required for pickups, remove asterisk */
128 function isNotDelivery()
129 {
130     document.getElementById("star1").innerHTML = "";
131     document.getElementById("star2").innerHTML = "";
132 }
133
134 /* get X minutes from now, where X = 15 for pickups OR X = 30 for ↵
    deliveries */
135 function getETA(minutes) {
136     d = new Date();

```

```

137     return new Date(d.getTime() + minutes*60000);
138 }
139
140 /* order summary displays the following message to the user:
141  * thank you message, order type, estimated time of arrival, and total ←
    price */
142 function alertMessage(eta, orderType, totalPrice)
143 {
144     var hour = eta.getHours() > 12 ? eta.getHours() - 12 : eta.getHours();
145     var minute = eta.getMinutes();
146     var am_pm = eta.getHours() >= 12 ? "PM" : "AM";
147
148     /* SPECIAL CASE: user has submitted order without ordering anything */
149     if (totalPrice == "") {
150         totalPrice = 0;
151         totalPrice.toFixed(2);
152     }
153     alert("Thanks for your order!\nEstimated " + orderType + " time: " + (←
        hour) + ":" + (minute.toString().padStart(2, '0')) + " " + am_pm + (←
        "\nTotal amount to be paid: $" + totalPrice);
154 }
155
156 function validate()
157 {
158     var isValid = true;
159
160     /* required validations: last name and phone */
161     if (!validateLastName()) isValid = false;
162     if (!validatePhone()) isValid = false;
163
164     /* check order type */
165     var timeNeeded, orderType;
166     var is_pickup = document.getElementsByName("p_or_d")[0].checked;
167
168     /* secondary validations (delivery only): street and city */
169     if (!is_pickup) {
170         if (!validateAddress("street", "error_street")) isValid = false;
171         if (!validateAddress("city", "error_city")) isValid = false;
172         timeNeeded = 30;
173         orderType = "delivery";
174     }
175     else {
176         timeNeeded = 15;
177         orderType = "pickup";
178     }
179
180     /* if all the field are valid, output order summary to user */

```

```

181     var eta = getETA(timeNeeded);
182     var totalPrice = document.getElementById("total").value;
183     if (isValid)
184         alertMessage(eta, orderType, totalPrice);
185
186     return isValid;
187 }
188
189 function makeSelect(name, minRange, maxRange)
190 {
191     var t= "";
192     t = "<select name='" + name + "' size='1' onchange='updateTotals()'>";
193     for (j=minRange; j<=maxRange; j++)
194         t += "<option value='" + j + "'>" + j + "</option>";
195     t+= "</select>";
196     return t;
197 }

```

---

```

1 </script>
2
3 <h1>Jade Delight</h1>
4 <div class="container">
5 <form onsubmit="return validate()">
6
7 <p>First Name: <input type="text" name='fname' placeholder="Johnny"/></p>
8 <p>Last Name<span class="star">*</span>: <input type="text" name='lname' ↵
    placeholder="Appleseed"/><span id="error_lname">&nbsp;</span></p>
9 <p>Street<span class="star" id="star1"></span>: <input type="text" name=' ↵
    street' placeholder="123 Winning Way"/><span id="error_street">&nbsp;< ↵
    /span></p>
10 <p>City<span class="star" id="star2"></span>: <input type="text" name=' ↵
    city' placeholder="Springfield"/><span id="error_city">&nbsp;</span></ ↵
    p>
11 <p>Phone<span class="star">*</span>: <input type="tel" name='phone' ↵
    placeholder="123-456-7890"/><span id="error_phone">&nbsp;</span></p>
12 <p>
13     <input type="radio" name="p_or_d" value = "pickup" checked="checked" ↵
        onchange="isNotDelivery()"/>Pickup
14     <input type="radio" name='p_or_d' value = 'delivery' onchange=" ↵
        isDelivery()"/>
15     Delivery
16 </p>
17 <table border="0" cellpadding="3">
18 <tr>
19     <th>Select Item</th>

```

```

20     <th>Item Name</th>
21     <th>Cost Each</th>
22     <th>Total Cost</th>
23 </tr>
24 <script language="javascript">
25     var s = "";
26     for (i=0; i< menuItems.length; i++)
27     {
28         s += "<tr><td>";
29         s += makeSelect("quan" + i, 0, 10);
30         s += "</td><td>" + menuItems[i].name + "</td>";
31         s += "<td> $ " + menuItems[i].cost.toFixed(2) + "</td>";
32         s += "<td>$<input type='text' name='cost' /></td></tr>";
33     }
34     document.writeln(s);
35 </script>
36 </table>
37 <p>Subtotal:
38     $<input type="text" name='subtotal' id="subtotal" />
39 </p>
40 <p>Mass tax (6.25%):
41     $ <input type="text" name='tax' id="tax" />
42 </p>
43 <p>Total: $ <input type="text" name='total' id="total" />
44 </p>
45
46 <input type = "submit" value = "Submit Order" style="position:relative; ←
    left:25%;"/>
47
48 <!-- the button below is for debugging purposes -->
49 <!-- <button onclick="validate()">Submit Button</button> -->
50
51 </form>
52 </div>
53 </body>
54 </html>

```

---

## Discussion Questions

If I had an extra two weeks for this project, I would add advanced CSS styling so that an `input [ type="text" ]` box starts out red (with an X next to it) and turns green (with a check mark next to it) as soon as someone starts typing in the input field. I would also integrate with Google Maps or another address-based service to have the form validate the street and city fields for legitimacy. Lastly, I would improve styling further and perhaps log the orders in a database so that the restaurant could one day do some analytics on its business growth and consumer trends / insights.

The thing I like best about Javascript is that the type of variables does not need to be specified and integers and floats can easily be treated as strings. I find this aspect of Javascript to be very handy. I also like that alerts are so informative and helpful for debugging. I did not use alerts in the previous assignment, but I enjoyed using them for this assignment. Event listeners are also very powerful when working with interactive webpages. I would like to learn more about hover effects and other interactive components. The weirdest thing to adjust to has been the fact that Javascript is not a compiled language. I find it so odd that a function can call another function that is defined below it. Crazy!