

Documentation for Financial Transactions HTML Page

Contents

1	HTML	1
1.1	Preamble and head	1
2	Javascript	2
3	CSS	3
A	HTML Source	4
B	Javascript Source	8
C	CSS Source	13

1 HTML

1.1 Preamble and head

```
1 <!DOCTYPE html>
```

This line declares that the document is an HTML5 document.

```
2 <head>
```

```
3     <meta charset = "UTF-8"/>
```

```
4     <link rel="stylesheet" type="text/css" href="./style.css"/>
```

```
5     <script src="./script.js"></script>
```

```
6 </head>
```

`<head>` tags are used to contain meta information about the document. Within the `head` element:

- The first line defines the character set of the document.
- The second line defines the source of an external CSS document.
- The third line defines the source of an external Javascript document.

2 Javascript

3 CSS

A HTML Source

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset = "UTF-8"/>
5     <link rel="stylesheet" type="text/css" href="./style.css"/>
6     <script src="./script.js"></script>
7   </head>
8   <body>
9     <article id="inputFields">
10       <form onsubmit="return false" autocomplete="off">
11         <section>
12           <label for="date">Date:</label><br/>
13           <input id="date" name="date" type="date" placeholder="
14             ↪ Date" required/>
15         </section>
16         <section>
17           <label for="account">Account Number:</label><br/>
18           <input id="account" name="account" type="text"
19             ↪ placeholder="Account Number" required/>
20         </section>
21         <section>
22           <label for="type">Transaction Type:</label><br/>
23           <select id="type" name="type">
24             <option value=""></option>
25             <option value="BUY">BUY</option>
26             <option value="SELL">SELL</option>
27             <option value="DIVIDEND">DIVIDEND</option>
28             <option value="INTEREST">INTEREST</option>
29             <option value="WITHDRAW">WITHDRAW</option>
30             <option value="DEPOSIT">DEPOSIT</option>
31           </select>
32         </section>
33         <section>
34           <label for="security">Security:</label><br/>
35           <input id="security" name="security" type="text"
36             ↪ placeholder="Security" required/>
37         </section>
38         <section>
39           <label for="amount">Amount:</label><br/>
40           <input id="amount" name="amount" type="text"
41             ↪ placeholder="Unit Amount" required/>
42         </section>
43         <section>
44           <label for="dAmount">$ Amount:</label><br/>
45           <input id="dAmount" name="dAmount" type="text"
46             ↪ placeholder="$ Amount" required/>
47         </section>
```

```

48
49         <section>
50             <button id="add" type="submit" onclick="
                    ↪ addTransactionButton();">Add Transaction</button>
                    ↪ >
51             <button id="save" type="submit" hidden="true" onclick
                    ↪ ="saveChanges();">Save</button>
52             <button id="discard" type="button" hidden="true"
                    ↪ onclick="discardChanges();">Discard</button>
53         </section>
54     </form>
55 </article>
56
57 <article id="table">
58     <table>
59         <thead>
60             <tr>
61                 <th>
62                     <section>
63                         Transaction ID
64                     </section>
65                     <section class="sort">
66                         <button type="button" onclick="sortTable(0,
                                ↪ true)">^</button>
67                         <button type="button" onclick="sortTable(0,
                                ↪ false)">v</button>
68                     </section>
69                 </th>
70                 <th>
71                     <section>
72                         Date
73                     </section>
74                     <section class="sort">
75                         <button type="button" onclick="sortTable(1,
                                ↪ true)">^</button>
76                         <button type="button" onclick="sortTable(1,
                                ↪ false)">v</button>
77                     </section>
78                 </th>
79                 <th>
80                     <section>
81                         Account Number
82                     </section>
83                     <section class="sort">
84                         <button type="button" onclick="sortTable(2,
                                ↪ true)">^</button>
85                         <button type="button" onclick="sortTable(2,
                                ↪ false)">v</button>
86                     </section>
87                 </th>
88                 <th>
89                     <section>
90                         Transaction Type
91                     </section>

```

```

92         <section class="sort">
93             <button type=button onclick="sortTable(3,
94                 ↪ true)">^</button>
95             <button type=button onclick="sortTable(3,
96                 ↪ false)">v</button>
97         </section>
98     </th>
99     <th>
100         <section>
101             Security
102         </section>
103         <section class="sort">
104             <button type=button onclick="sortTable(4,
105                 ↪ true)">^</button>
106             <button type=button onclick="sortTable(4,
107                 ↪ false)">v</button>
108         </section>
109     </th>
110     <th>
111         <section>
112             Amount
113         </section>
114         <section class="sort">
115             <button type=button onclick="sortTable(5,
116                 ↪ true)">^</button>
117             <button type=button onclick="sortTable(5,
118                 ↪ false)">v</button>
119         </section>
120     </th>
121     <th>
122         <section>
123             $ Amount
124         </section>
125         <section class="sort">
126             <button type=button onclick="sortTable(6,
127                 ↪ true)">^</button>
128             <button type=button onclick="sortTable(6,
129                 ↪ false)">v</button>
130         </section>
131     </th>
132     <th>
133         <section>
134             Cost Basis
135         </section>
136         <section class="sort">
137             <button type=button onclick="sortTable(7,
138                 ↪ true)">^</button>
139             <button type=button onclick="sortTable(7,
140                 ↪ false)">v</button>
141         </section>
142     </th>
143     <th>Actions</th>
144 </tr>
145 </thead>

```



```
136             <tbody id="tableBody">
137             </tbody>
138         </table>
139     </article>
140 </body>
141 </html>
```

B Javascript Source

```
1 function addTransactionButton() {
2     var data = getData();
3     if(data) {
4         var date = data[0];
5         var account = data[1];
6         var type = data[2];
7         var security = data[3];
8         var amount = data[4];
9         var dAmount = data[5];
10        var costBasis = data[6];
11
12        var id = generateId();
13
14        addTransaction(id, date, account, type, security, amount, dAmount,
15            ↪ costBasis);
16    }
17 }
18 function addTransaction(id, date, account, type, security, amount, dAmount
19     ↪ , costBasis) {
20     var tableBody = document.getElementById('tableBody');
21     var newRow = tableBody.insertRow(0);
22     newRow.classList += "bodyRow";
23     var actionsContent = "<button type='button' onclick='editRow(this)''>
24     ↪ Edit</button> <button type='button' onclick='deleteRow(this)''>
25     ↪ Delete</button>";
26     var rowContents = [id, date, account, type, security, amount, dAmount,
27     ↪ costBasis, actionsContent];
28
29     for(var i = 0; i < rowContents.length; i++) {
30         var newCell = newRow.insertCell(i);
31         newCell.innerHTML = rowContents[i];
32         if(i == 0) {
33             newCell.classList += "idCell";
34         }
35     }
36 }
37
38 function getData() {
39     var date = document.getElementById("date").value;
40     var account = document.getElementById("account").value;
41     var type = document.getElementById("type").value;
42     var security = document.getElementById("security").value;
43     var amount = document.getElementById("amount").value;
44     var dAmount = document.getElementById("dAmount").value;
45
46     amount = Number(amount);
47
48     if(dAmount[0] == '$') {
49         dAmount = dAmount.substr(1);
50     }
51 }
```

```

48     dAmount = Number(dAmount);
49
50     if(validate(date, account, type, security, amount, dAmount)) {
51         var costBasis = calculateCostBasis(amount, dAmount);
52         dAmount = '$' + dAmount.toFixed(2);
53
54         return [ date, account, type, security, amount, dAmount, costBasis
55                 ↪ ];
56     }
57     else return false;
58 }
59 function validate(date, account, type, security, amount, dAmount) {
60     if(!validateDate(date)) return false;
61     if(!validateAccount(account)) return false;
62     if(!validateType(type)) return false;
63     if(!validateSecurity(security)) return false;
64     if(!validateAmount(amount)) return false;
65     if(!validateDAmount(dAmount)) return false;
66
67     return true;
68 }
69
70 function validateDate(date) {
71     realDate = new Date();
72     inputDate = document.getElementById('date').valueAsNumber;
73
74     if(isNaN(inputDate)) {
75         alert('Error: No date specified');
76         return false;
77     }
78
79     if(realDate.valueOf() < inputDate) {
80         alert('Error: Date is in the future');
81         return false;
82     }
83
84     return true;
85 }
86
87 function validateAccount(account) {
88     if(account == '') {
89         alert('Error: Missing Account Number');
90         return false;
91     }
92
93     return true;
94 }
95
96 function validateType(type) {
97     if(type == '') {
98         alert('Error: Missing Transaction Type');
99         return false;
100    }

```

```

101
102     return true;
103 }
104
105 function validateSecurity(security) {
106     if(security == '') {
107         alert('Error: Missing Security');
108         return false;
109     }
110
111     return true;
112 }
113
114 function validateAmount(amount) {
115     if(amount == '') {
116         alert('Error: Missing Amount');
117         return false;
118     }
119
120     if(isNaN(amount)) {
121         alert('Error: Invalid Amount');
122         return false;
123     }
124
125     return true;
126 }
127
128 function validateDAmount(dAmount) {
129     if(dAmount == '') {
130         alert('Error: Missing $ Amount');
131         return false;
132     }
133
134     if(isNaN(dAmount)) {
135         alert('Error: Invalid $ Amount');
136         return false;
137     }
138
139     return true;
140 }
141
142 function generateId() {
143     var id = '';
144     var idLength = 6;
145
146     var characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
147     var charactersLength = characters.length;
148
149     var unique = false;
150
151     while(!unique) {
152         for(var i = 0; i < idLength; i++) {
153             id += characters.charAt(Math.floor(Math.random() *
                ↪ charactersLength));

```

```

154     }
155
156     unique = true;
157     for(var i = 0; i < document.getElementsByClassName('idCell').
        ↪ length; i++) {
158         if(document.getElementsByClassName('idCell')[i].innerText ==
            ↪ id) {
159             unique = false;
160             break;
161         }
162     }
163 }
164 return id;
165 }
166
167 function calculateCostBasis(amount, dAmount) {
168     costBasis = '$' + (dAmount / amount).toFixed(2);
169     return costBasis;
170 }
171
172 function deleteRow(button) {
173     var row = button.parentElement.parentElement;
174     document.getElementById("tableBody").removeChild(row);
175
176     if(document.getElementsByClassName('editing').length == 0) {
177         document.getElementById('add').removeAttribute('hidden');
178         document.getElementById('save').setAttribute('hidden', true);
179         document.getElementById('discard').setAttribute('hidden', true);
180     }
181 }
182
183 function editRow(button) {
184     if(document.getElementsByClassName('editing').length > 0)
185         document.getElementsByClassName('editing')[0].classList = "bodyRow
            ↪ ";
186
187     var row = button.parentElement.parentElement;
188     var rowContent = row.getElementsByTagName('td');
189     row.classList = "bodyRow editing";
190
191     document.getElementById('date').value = rowContent[1].innerText;
192     document.getElementById('account').value = rowContent[2].innerText;
193     document.getElementById('type').value = rowContent[3].innerText;
194     document.getElementById('security').value = rowContent[4].innerText;
195     document.getElementById('amount').value = rowContent[5].innerText;
196     document.getElementById('dAmount').value = rowContent[6].innerText;
197
198     document.getElementById('add').setAttribute('hidden', true);
199     document.getElementById('save').removeAttribute('hidden');
200     document.getElementById('discard').removeAttribute('hidden');
201 }
202
203 function saveChanges() {
204     data = getData();

```

```

205     if(data) {
206         rowToEdit = document.getElementsByClassName('editing')[0];
207         cellsToEdit = rowToEdit.getElementsByTagName('td');
208
209         for(var i = 0; i < data.length; i++) {
210             cellsToEdit[i + 1].innerHTML = data[i];
211         }
212         rowToEdit.classList = "bodyRow";
213     }
214
215     document.getElementById('add').removeAttribute('hidden');
216     document.getElementById('save').setAttribute('hidden', true);
217     document.getElementById('discard').setAttribute('hidden', true);
218 }
219
220 function discardChanges() {
221     document.getElementsByClassName('editing')[0].classList = "bodyRow";
222
223     document.getElementById('add').removeAttribute('hidden');
224     document.getElementById('save').setAttribute('hidden', true);
225     document.getElementById('discard').setAttribute('hidden', true);
226 }
227
228 function sortTable(column, ascending) {
229     var tableBody = document.getElementById('tableBody');
230     var rows = document.getElementsByClassName('bodyRow');
231
232     var sorting = true;
233     while(sorting) {
234         sorting = false;
235         for(var i = 0; i < (rows.length - 1); i++) {
236             rowA = rows[i].getElementsByTagName('td')[column];
237             rowB = rows[i + 1].getElementsByTagName('td')[column];
238
239             var swap = false;
240
241             if(ascending && rowA.innerHTML.toLowerCase() > rowB.innerHTML.
                ↪ toLowerCase()) swap = true;
242             else if(!ascending && rowA.innerHTML.toLowerCase() < rowB.
                ↪ innerHTML.toLowerCase()) swap = true;
243
244             if(swap) {
245                 sorting = true;
246                 rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
247             }
248         }
249     }
250 }

```

C CSS Source

```
1  body {
2      font-size: 14px;
3  }
4
5  #inputFields {
6      padding: 10px 0;
7      overflow-x: auto;
8  }
9
10 form {
11     min-width: 1900px;
12 }
13
14     form > section {
15         width: 14%;
16         display: inline-block;
17     }
18
19     input,
20     select {
21         min-width: 100px;
22         width: 80%;
23     }
24
25     button {
26         width: 40%;
27     }
28
29 #table {
30     max-height: 80vh;
31     overflow: auto;
32 }
33
34 table {
35     width: 100%;
36     margin: auto;
37     border-collapse: collapse;
38 }
39     th {
40         min-width: 200px;
41         width: 10%;
42         position: sticky;
43         background: white;
44         top: 0;
45     }
46
47     th > section {
48         width: 80%;
49         display: inline-block;
50         padding: 0;
51         margin: 0;
52     }
```

```
53
54 .sort {
55     width: 10%;
56 }
57
58 .sort > button {
59     padding: 0;
60     border: 0;
61     display: block;
62     width: 100%;
63 }
64
65 .editing {
66     background-color: yellow;
67 }
68
69 #table,
70 table,
71 td,
72 th {
73     box-shadow: 1px 1px black, inset 1px 1px black;
74 }
```