

Condition Statements

At the end of this chapter, you will be able to:

- Use the If statement
- Use the Else statement
- Use the Else If Statement
- Use Nested Condition Statements

Instruction 1: Use the If Statement

1. Condition statements are used in our programs to make decisions. We make decisions in our everyday lives. If a certain condition occurs, we choose an option, or else we choose another. This becomes second nature to us thus we are not too conscious about them. For example:
 - **If** the sky is dark **then** it is likely to rain.
 - **If** I buy a new mobile phone **then** I wouldn't have enough money next month.
 - **If** it is Sunday **then** I will go out or **else if** it is Monday **then** I will go to school.
2. Condition Statements are one of the basic building blocks of a computer program. We write condition statements to tell JavaScript to make decisions. For example:
 - **If** x is smaller than 5, **then** add it to the y, or else add it to z.
 - **If** the text field contains nothing, **then** pop a message box to ask the user to enter a number.
 - **If** checkbox1 is checked, **then** add the contents of x to y or **else** add the contents of x to z.
3. This is the structure of an If statement in JavaScript:

```

1. if (Condition To Test) {
2. //do the following
3. //and the following
4. }

```

In this structure, JavaScript tests if a specific condition is true. If it is, the codes in lines 2 and 3 will be executed. The “}” in line 4 denotes the end of the If block of codes. All If statements in JavaScript must start with a curly parenthesis “{” and end with a corresponding closed parenthesis “}”. Here is an example:

```

1. var test = 1
2.
3. if (test === 1) {
4.   alert("Test is 1");
5. }

```

Line 1 declares a variable called test and assigns the value 1 to test. Line 3 checks if the value of test is 1. If it is, then an alert pops up to display the text “Test is 1”.

4. In this example, we checked if the condition (`test === 1`) is true. The sign “===” is a comparison operator. Apart from testing if values are equal to each other, we may also test other conditions such as Not Equal, Less than and Greater than. To do this, we use comparison operators. This table shows a list of JavaScript comparison operators.

Meaning	Operators	Example
Equal	===	x === y
Not Equal	!==	x !== y
Less than	<	x < y
Less than or equal to	<=	x <= y
Greater than	>	x > y
Greater than or equal to	>=	x >= y
Test for more than 1 condition	&&	(x === 2) && (y === 4)
Test if value is either or something		(x === 2) (y === 4)
Test if value is not something	!	!(x === y)

5. Consider the following example. The code below checks if a toy is beyond Johnny's budget. If it is within his budget, he is allowed to buy it.

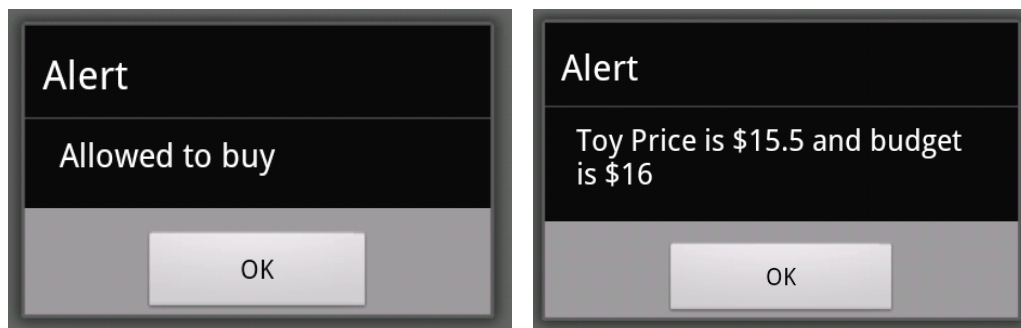
```

1 function checkCost(){
2     var toyPrice, budget;
3
4     toyPrice = 15.5;
5     budget = 16;
6
7     if (toyPrice <= budget) {
8         alert("Allowed to buy");
9     }
10    alert("Toy Price is $" + toyPrice + " and budget is $" + budget);
11 }
12 checkCost();

```

Line 2 declares 2 variables, `toyPrice` and `budget`. Line 4 and 5 assign the respective values to the variables. Line 7 checks if the toy's price is lower than or equal to `budget`. If it is, a message box pops to say "Allowed to buy". Line 11 displays the price of the toy and the budget.

6. The output of the program is as follows:

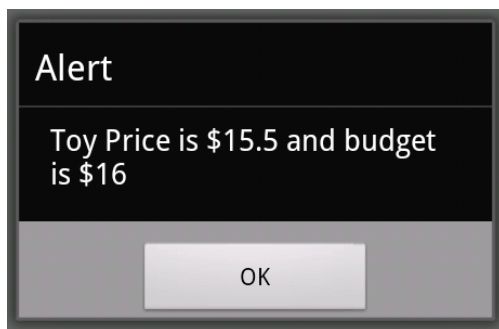


7. Assuming that the buyer's parents has decided that apart from meeting the budget, Johnny also needs to be a "good boy" to buy the toy, we amend the codes as follows:

```

1 function checkCost(){
2   var toyPrice, budget, conduct;
3
4   toyPrice = 15.5;
5   budget = 16;
6   conduct = "naughty"
7
8   if ((toyPrice <= budget) && (conduct === "good")) {
9     alert("Allowed to buy");
10  }
11  alert("Toy Price is $" + toyPrice + " and budget is $" + budget);
12 }
13
14 checkCost();

```



We added a new string variable “conduct” in line 2. In line 6 we assign the value “naughty” to conduct. In Line 8, we added a second criterion to check if Johnny has been “Good”. Run the program and note that the alert that says “Allowed to Buy” no longer pops up.

8. Johnny’s parents have decided to add a new criterion – If Johnny scores 100 marks for his test, then he can buy the toy even if he is not a good boy and even if the toy is beyond his budget. The following codes reflect this:

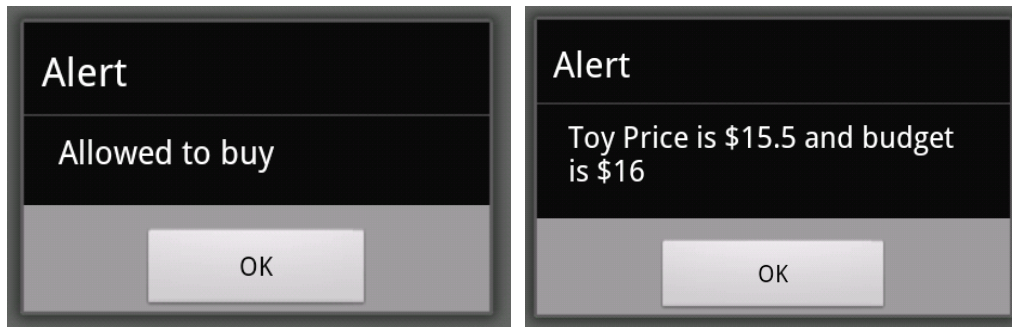
```

1 function checkCost(){
2   var toyPrice, budget, conduct, testScore;
3
4   testScore = 100;
5   toyPrice = 15.5;
6   budget = 16;
7   conduct = "naughty";
8
9   if ((testScore === 100) || ((toyPrice <= budget) && (conduct ===
  "good")))) {
10    alert("Allowed to buy");
11  }
12  alert("Toy Price is $" + toyPrice + " and budget is $" + budget);
13 }
14
15 checkCost();

```

We added a new variable `testScore` in line 2. Line 4 assigns 100 to `testScore`. In line 9, we check if `testScore === 100`. We also checks if `toyPrice` is less than or equal to `budget` and if

`conduct` is good. Notice that the checks on `toyPrice` and `conduct` are enclosed in an additional set of brackets to denote that both of these conditions have to be met at the same time.



Thus, if Johnny's parents change the conditions to say that being a "Good" boy is a necessary criteria to receive his toy whether or not he scores 100 for his test, then the `If` statement becomes:

```
if (((testScore === 100) || (toyPrice <= budget)) && (conduct = "Good")) {
```

If the conditions are changed to say that Johnny may buy his toy as long as one of these conditions is met:

- His test score is 100
- The toy's price is within the budget
- His conduct is good

Then the `if` statement becomes:

```
if ((testScore === 100) || (toyPrice <= budget) || (conduct === "Good")) {
```

If Johnny is allowed to buy his toy as long as he scores between 80 and 100 for his test, then the statement becomes:

```
if ((testScore >= 80) And (testScore <= 100)) {
```

9. Create a new project, C7.
10. Follow Instruction1 in Chapter 4 to include jQuery and jQuery Mobile libraries into your project.
11. Create the following user interface. This application calculates the total salary for a sales person. In this example, a sales person's salary is calculated based on his basic salary as well as the number of items he sold. However, he receives a commission only if he sells more than the minimum he needs to sell to receive a commission.
12. Edit `index.html`:

Basic Salary:

2000

Items Sold:

8

Commission Per Item Sold:

1.5

Minimum to sell for Commission:

5

Calculate

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <!--
5       Customize the content security policy in the meta tag below as
6       needed. Add 'unsafe-inline' to default-src to enable inline JavaScript.
7       For details, see http://go.microsoft.com/fwlink/?LinkID=617521
8       -->
9       <meta http-equiv="Content-Security-Policy" content="default-src
10        'self' data: gap: https://ssl.gstatic.com 'unsafe-eval'; style-src
11        'self' 'unsafe-inline'; media-src *">
12
13       <meta http-equiv="content-type" content="text/html;
14        charset=UTF-8" />
15       <meta name="format-detection" content="telephone=no">
16       <meta name="msapplication-tap-highlight" content="no">
17       <meta name="viewport" content="user-scalable=no, initial-
18        scale=1, maximum-scale=1, minimum-scale=1, width=device-width">
19       <link rel="stylesheet" type="text/css" href="css/index.css">
20       <link rel="stylesheet" href="css/jquery.mobile-1.4.5.css">
21
22       <script src="lib/jquery-1.11.2.min.js"></script>
23       <script src="lib/jquery.mobile-1.4.5.min.js"></script>
24       <script src="scripts/common.js"></script>
25
26       <title>C7</title>
27     </head>
28     <body>
29       <div data-role="main" class="ui-content">
30         <form name="salaryform" id="salaryform">
31           <div class="ui-field-contain">
32
33             <div data-role="fieldcontainer">
34               <label for="txtBasicsalary">Basic
35               Salary:</label>
36               <input type="text" name="txtBasicSalary"
37               id="txtBasicSalary">

```

```

31         </div>
32
33         <div data-role="fieldcontainer">
34             <label for="txtItemssold">Items Sold:</label>
35             <input type="text" name="txtItemsSold"
id="txtItemsSold">
36         </div>
37
38         <div data-role="fieldcontainer">
39             <label for="txtCommission">Commission Per Item
Sold:</label>
40             <input type="text" name="txtCommission"
id="txtCommission">
41         </div>
42
43         <div data-role="fieldcontainer">
44             <label for="txtMinimum">Minimum to sell for
Commission:</label>
45             <input type="text" name="txtMinimum"
id="txtMinimum">
46         </div>
47
48             <input type="button" value="Calculate"
id="btnCalculate">
49         </div>
50     </form>
51 </div>
52
53     <script type="text/javascript" src="cordova.js"></script>
54     <script type="text/javascript"
src="scripts/platformOverrides.js"></script>
55     <script type="text/javascript" src="scripts/index.js"></script>
56 </body>
57 </html>

```

13. Edit index.js:

```

1 (function () {
2
3     $(document).ready(function () {
4         $("#btnCalculate").bind("click", function () {
5             calculateSalary();
6         });
7     });
8
9
10    function calculateSalary() {
11        var basicSalary, itemsSold, commission, minimum, totalSalary;
12
13        basicSalary = parseFloat($("#txtBasicSalary").val());
14        itemsSold = parseFloat($("#txtItemsSold").val());
15        commission = parseFloat($("#txtCommission").val());
16        minimum = parseFloat($("#txtMinimum").val());
17
18        totalSalary = basicSalary;

```

```

19
20     if (itemsSold > minimum) {
21         totalSalary = totalSalary + (itemsSold * commission);
22     }
23
24     alert("Your total salary is $" + totalSalary);
25 }
26
27 } )();

```

- a. Lines 3 to 7 assigns the task to be done when the btnCalculate button is pressed. The calculateSalary() function will be executed.
 - b. Lines 13 to 16 assign the values that the users entered in the text fields to the basicSalary, itemsSold, commission and minimum variables.
 - c. Line 18 assigns the value of basicSalary to totalSalary.
 - d. Line 20 checks if the itemsSold value is greater than minimum. We will only need to calculate the employee's commission rate if he is eligible. He is not eligible for commission if he did not sell more than the minimum.
 - e. Line 21 calculates his total commission by multiplying itemsSold with commission and adding it to his totalSalary.
 - f. On line 24, we pop the totalSalary in an alert.
14. Run the program with the values entered in the text fields in point 12. The total salary for this sales person is \$2012 inclusive of commission.



Instruction 2: Use the Else Statement

1. The Else statement works together with the If statement to tell JavaScript what it needs to do if the condition is false. The structure of an else statement in JavaScript looks like this:


```

1. if (Condition To Test) {
2.     //do the following
3.     //and the following
4. }
5. else {
6.     //do this instead
7. }

```

2. In this structure, JavaScript tests if a specific condition is true in line 1. If it is, the codes in lines 2 and 3 will be executed. Line 5 executes if the condition in line 1 is false. Here is an example:

```

1  function testElse(){
2  var test;
3  test = 5;
4
5  if (test === 1){
6  alert("Test is 1");
7  }
8  else {
9  alert("Test is not 1");
10 }
11
12 }
13 testElse();

```

- a. Line 2 declares a variable called `test`.
 - b. Line 3 assigns the value 5 to `test`.
 - c. Line 5 checks if the value of `test` is 1. If it is, then an alert pops up to display the text “Test is 1” or else an alert pops up to display the text “Test is not 1”.
3. We will use the example of Johnny and his toy to illustrate the `If...else...` statement.

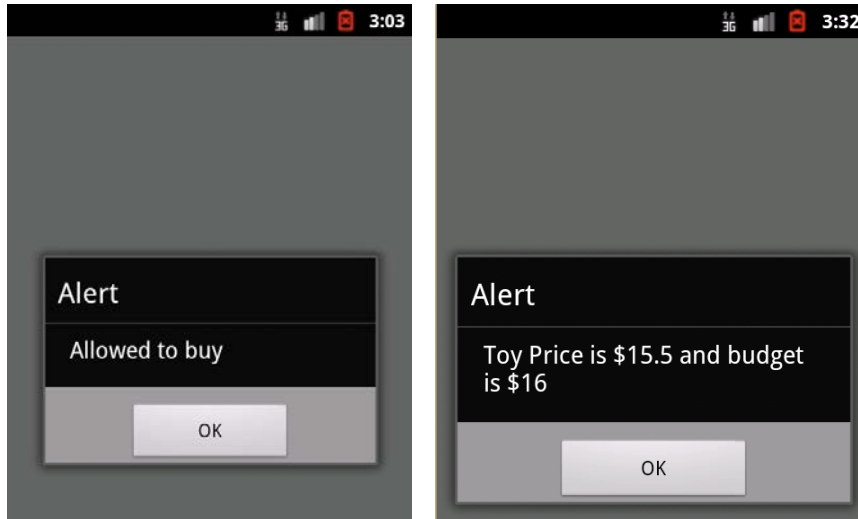
```

1  function checkCost(){
2  var toyPrice, budget, conduct, testScore;
3
4  testScore = 100;
5  toyPrice = 15.5;
6  budget = 16;
7  conduct = "Naughty";
8
9  if ((testScore >= 80) && (testScore <= 100)) {
10      alert("Allowed to buy");
11  }
12  else {
13      alert("Not allowed to buy");
14  }
15  alert("Toy Price is $" + toyPrice + " and budget is $" +
    budget);
16  }

```

```
17      checkCost();
```

- a. In this example, line 9 checks if Johnny's test score is between 80 and 100. If it is, a message box pops up to say "Allowed to buy". If not, another message box pops up to say "Not Allowed to buy".



4. We will edit the salary calculator in Instruction 1 to use a else statement as follows:

```

1 (function () {
2
3     $(document).ready(function () {
4         $("#btnCalculate").bind("click", function () {
5             calculateSalary();
6         });
7     });
8
9
10    function calculateSalary() {
11        var basicSalary, itemsSold, commission, minimum, totalSalary;
12
13        basicSalary = parseFloat($("#txtBasicSalary").val());
14        itemsSold = parseFloat($("#txtItemsSold").val());
15        commission = parseFloat($("#txtCommission").val());
16        minimum = parseFloat($("#txtMinimum").val());
17
18        totalSalary = basicSalary;
19
20        if (itemsSold > minimum) {
21            totalSalary = totalSalary + (itemsSold * commission);
22        }
23        else {
24            alert("You are not eligible for comission payout");
25        }
26
27        alert("Your total salary is $" + totalSalary);
28    }
29

```

```
30 } )();
```

- a. An else condition has been added in lines 23, 24 and 25 to pop the alert to say that the sales person is not eligible for commission if itemSold is less than or equal to minimum.

Instruction 3: Use the Else If Statement

2. The `else if` statement works together with the `if` statement to tell JavaScript that it needs to consider another condition if the previous condition statement is not true. The structure of an `else if` statement in Excel VBA looks like this:

```
1. if (Condition-1 To Test) {
2.     //do the following
3.     //and the following
4. } else if (Condition-2 To Test) {
5.     //do this instead
6. } else if (Condition-3 To Test) {
7.     //do this instead
8. }
9. else {
10.    //do this if all else fails
11.}
```

3. In the structure above, JavaScript tests if a specific condition is true in line 1. If it is, the codes in lines 2 and 3 will be executed. If Line 1 is not true, line 4 will be tested. If lines 1 and 4 are not true, then line 6 will be tested. If all 3 tests on line 1, 4 and 6 fail, then lines 9 and 10 will be executed.
4. Here is an example:

```
1      function elseIfTest(){
2          var test;
3          test = 5;
4
5          if (test === 5) {
6              alert("Test is 5");
7          }
8          else if (test === 2) {
9              alert("Test is 2");
10         }
11         else if (test === 3) {
12             alert("Test is 3");
13         }
14         else {
15             alert("I don't know what test is");
16         }
17     }
18     elseIfTest();
```

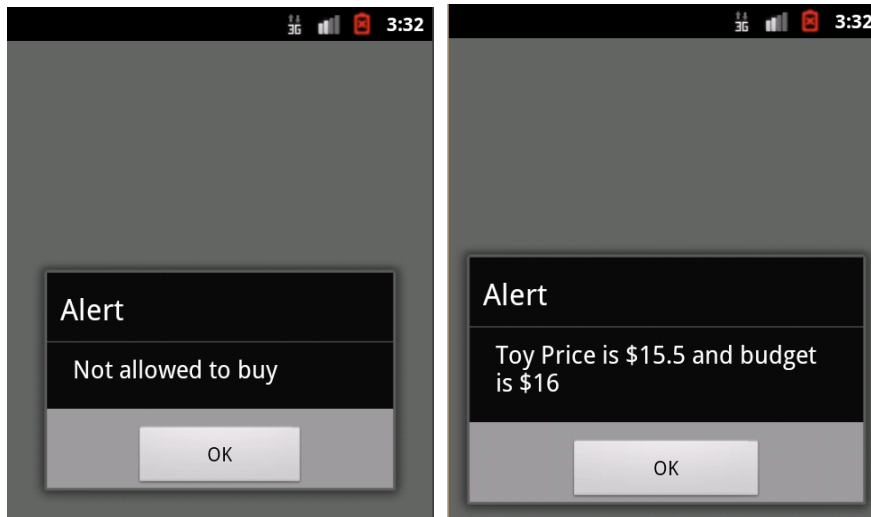
- a. Line 2 declares a variable called `test`.

- b. Line 3 assigns the value 5 to test.
 - c. Line 5 checks if test is equal to 5. If it is, an alert pops to say that “Test is 5”.
 - d. Or else, line 8 checks if test is equal to 2. If it is, an alert pops to say that “Test is 2”.
 - e. Or else, line 11 checks if test is equal to 3. If it is, an alert pops to say that “Test is 3”.
 - f. Or else line 15 pops an alert to say “I don’t know what test is”.
5. We will use the example of Johnny and his toy to illustrate the `if...else...if` statement. Let’s assume that Johnny’s parents have set a new set of criteria as follows:
- If Johnny scores between 80 and 100 for his test, then he may purchase a toy no matter how expensive it is - but he must be a good boy.
 - If he scores less than 80 for his test and he is a good boy, he may purchase a toy within the budget set for him.
 - If he has been naughty, then he cannot buy a toy.

```

1      function checkCost(){
2      var toyPrice, budget, conduct, testScore;
3
4      testScore = 100;
5      toyPrice = 15.5;
6      budget = 16;
7      conduct = "Naughty";
8
9      if ((testScore >= 80) && (testScore <= 100) && (conduct ==
"Good")) {
10         alert("Allowed to buy");
11     }
12     else if ((testScore < 80) && (toyPrice <= budget) && (conduct ==
"Good")){
13         alert("Allowed to buy");
14     }
15     else {
16         alert("Not allowed to buy");
17     }
18     alert("Toy Price is $" + toyPrice + " and budget is $" +
budget);
19 }
20 checkCost();

```



6. For our salary calculation program, we will like to change the policy is to pay a different commission based on the number of items the employee sold.
 - Between 1 and 3 pieces, the employee receives \$1.20 commission per item sold.
 - Between 4 and 6 pieces, the employee receives \$1.40 commission per item sold.
 - For 7 pieces and above, the employee receives \$2.00 commission per item sold.
7. Change the interface to allow the user to enter commission rates based on 1 to 3, 4 to 6 and above 7 pieces sold.

Basic Salary:

2000

Items Sold:

8

Commission Per Item Sold (1 to 3 pieces):

1.2

Commission Per Item Sold (4 to 6 pieces):

1.4

Commission Per Item Sold (7 pieces and above):

2

Calculate

8. Edit index.html:

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <!--
5        Customize the content security policy in the meta tag below as
6        needed. Add 'unsafe-inline' to default-src to enable inline JavaScript.
7        For details, see http://go.microsoft.com/fwlink/?LinkID=617521
8      -->
9      <meta http-equiv="Content-Security-Policy" content="default-src
10 'self' data: gap: https://ssl.gstatic.com 'unsafe-eval'; style-src
11 'self' 'unsafe-inline'; media-src *">
12
13      <meta http-equiv="content-type" content="text/html;
14 charset=UTF-8" />
15      <meta name="format-detection" content="telephone=no">
16      <meta name="msapplication-tap-highlight" content="no">
17      <meta name="viewport" content="user-scalable=no, initial-
18 scale=1, maximum-scale=1, minimum-scale=1, width=device-width">
19      <link rel="stylesheet" type="text/css" href="css/index.css">
20      <link rel="stylesheet" href="css/jquery.mobile-1.4.5.css">
21
22      <script src="lib/jquery-1.11.2.min.js"></script>
23      <script src="lib/jquery.mobile-1.4.5.min.js"></script>
24      <script src="scripts/common.js"></script>
25
26      <title>C7</title>
27    </head>
28    <body>
29      <div data-role="main" class="ui-content">
30        <form name="salaryform" id="salaryform">
31          <div class="ui-field-contain">
32
33            <div data-role="fieldcontainer">
34              <label for="txtBasicsalary">Basic
35              Salary:</label>
36              <input type="text" name="txtBasicSalary"
37              id="txtBasicSalary">
38            </div>
39
40            <div data-role="fieldcontainer">
41              <label for="txtItemssold">Items Sold:</label>
42              <input type="text" name="txtItemsSold"
43              id="txtItemsSold">
44            </div>
45
46            <div data-role="fieldcontainer">
47              <label for="txtCommission13">Commission Per
48              Item Sold (1 to 3 pieces):</label>
49              <input type="text" name="txtCommission13"
50              id="txtCommission13">
51            </div>
52
53            <div data-role="fieldcontainer">

```

```

45     <label for="txtCommission46">Commission Per
Item Sold (4 to 6 pieces):</label>
46     <input type="text" name="txtCommission46"
id="txtCommission46">
47     </div>
48
49     <div data-role="fieldcontainer">
50     <label for="txtCommission7">Commission Per Item
Sold (7 pieces and above):</label>
51     <input type="text" name="txtCommission7"
id="txtCommission7">
52     </div>
53
54     <input type="button" value="Calculate"
id="btnCalculate">
55     </div>
56     </form>
57 </div>
58
59     <script type="text/javascript" src="cordova.js"></script>
60     <script type="text/javascript" src="scripts/platformOverrides.js"></script>
61     <script type="text/javascript" src="scripts/index.js"></script>
62 </body>
</html>

```

9. Edit index.js:

```

1 (function () {
2
3     $(document).ready(function () {
4         $("#btnCalculate").bind("click", function () {
5             calculateSalary();
6         });
7     });
8
9
10    function calculateSalary() {
11        var basicSalary, itemsSold, commission, minimum, totalSalary;
12
13        basicSalary = parseFloat($("#txtBasicSalary").val());
14        itemsSold = parseFloat($("#txtItemsSold").val());
15
16        commission13 = parseFloat($("#txtCommission13").val());
17        commission46 = parseFloat($("#txtCommission46").val());
18        commission7 = parseFloat($("#txtCommission7").val());
19
20        totalSalary = basicSalary
21
22        if ((itemsSold >= 1) && (itemsSold <= 3)) {
23            commission = commission13;
24        }
25        else if ((itemsSold >= 4) && (itemsSold <= 6)) {
26            commission = commission46;
27        }

```

```

28     else if (itemsSold >= 7) {
29         commission = commission7;
30     }
31     else {
32         commission = 0;
33     }
34
35     totalSalary = totalSalary + (itemsSold * commission);
36     alert("Your total salary is $" + totalSalary);
37 }
38
39 } )();

```

- a. Lines 16, 17 and 18 copy the values in the commission text fields to its corresponding variables.
 - b. Line 22 checks if the value of itemsSold is between 1 and 3. If it is, assign the value of commission13 to commission.
 - c. Line 25 checks if itemsSold is between 4 and 6. If it is, assign the value of commission46 to commission.
 - d. Line 28 checks if itemsSold is greater than or equal to 7. If it is, assign the value of commission7 to commission.
 - e. Line 31 catches the case where itemsSold is 0. commission is 0 if itemsSold is 0.
 - f. Line 35 calculates TotalSalary by adding the salary without commissions to the commission calculated.
10. Run the program by entering the values in point 6 into the text fields to get the following total salary:

The screenshot shows a web application interface. At the top, a light gray alert box displays the text "localhost:4400 says:" followed by "Your total salary is \$2016" and an "OK" button. Below the alert box, the form contains three input fields. The first field is labeled "Items Sold:" and contains the value "8". The second field is labeled "Commission Per Item Sold (1 to 3 pieces):" and contains the value "1.2". The third field is labeled "Commission Per Item Sold (4 to 6 pieces):" and contains the value "1.4".

Instruction 4: Use Nested Condition Statements

1. A new policy has been included so that commissions are not paid to employees who had a disciplinary action taken against them.
2. Make the following change to the user interface in Instruction 3.

Basic Salary:

2000

Items Sold:

10

Disciplinary Action Taken?

☒ Yes ☐ No

Commission Per Item Sold (1 to 3 pieces):

1.2

Commission Per Item Sold (4 to 6 pieces):

1.4

Commission Per Item Sold (7 pieces and above):

2

Calculate

3. Add the follow radio button control to index.html:

```

1 <div data-role="fieldcontainer">
2   <fieldset data-role="controlgroup" data-type="horizontal">
3     <legend>Disciplinary Action Taken?</legend>
4     <label for="Yes">Yes</label>
5     <input type="radio" name="rdoDiscipline" id="Yes" value="TRUE">
6
7     <label for="No">No</label>
8     <input type="radio" name="rdoDiscipline" id="No" value="FALSE">
9   </fieldset>
10 </div>

```

- a. This adds a radio button group that accepts either a “Yes” or “No” input. When “Yes” is pressed, the value “TRUE” is stored in the form variable discipline. When “No” is pressed, the value “FALSE” is stored in the form variable discipline.

4. Edit index.js:

```

1 (function () {
2
3     $(document).ready(function () {
4         $("#btnCalculate").bind("click", function () {
5             calculateSalary();
6         });
7     });
8
9
10    function calculateSalary() {
11        var basicSalary, itemsSold, commission, minimum, totalSalary;
12
13        basicSalary = parseFloat($("#txtBasicSalary").val());
14        itemsSold = parseFloat($("#txtItemsSold").val());
15
16        commission13 = parseFloat($("#txtCommission13").val());
17        commission46 = parseFloat($("#txtCommission46").val());
18        commission7 = parseFloat($("#txtCommission7").val());
19
20        totalSalary = basicSalary
21
22        if (getRadioValue("rdoDiscipline") == "FALSE") {
23
24            if ((itemsSold >= 1) && (itemsSold <= 3)) {
25                commission = commission13;
26            }
27            else if ((itemsSold >= 4) && (itemsSold <= 6)) {
28                commission = commission46;
29            }
30            else if (itemsSold >= 7) {
31                commission = commission7;
32            }
33            else {
34                commission = 0;
35            }
36        }
37        else {
38            commission = 0;
39        }
40
41        totalSalary = totalSalary + (itemsSold * commission);
42        alert("Your total salary is $" + totalSalary);
43    }
44
45 } )();

```

- a. Line 22 checks for the value of discipline. The value of discipline is either TRUE or FALSE based on whether the user pressed “Yes” or “No”. Note that the function `getRadioValue()` was developed in Chapter 6 and is now part of `common.js`.

If discipline is “FALSE”, then lines 24 to 35 will execute to determine the commission.

- b. If discipline is “TRUE”, then lines 37 to 39 will execute to assign 0 to commission.

- Run the Program to find the total salary of the employee based on his disciplinary action status.

localhost:4400 says:
Your total salary is \$2000

OK

Items Sold.
10

Disciplinary Action Taken?
Yes No

Commission Per Item Sold (1 to 3 pieces):
1.2

Exercise

- Write a program that allocates the correct grades based on the score that the student received for his test. Grades are allocated according to the following table:

Score	Grade
0 - 49	F
50 - 59	D
60 - 69	C
70 - 79	B
Above 80	A

Score:
79

Get Grade

Alert
B

OK

- What will be popped in the alert in this program?

```

1  function exercise2(){
2      var x, y, z;
3      x = 20;
4      y = 30;
5      z = 50;
6
7      if ((x - 10) < y){
8          alert(y - 5);
9      }
10     else {
11         alert(z - 5);
12     }
13 }
14 exercise2();

```

3. What will be popped in the alert in this program?

```

1  function exercise3(){
2      var x, y, z;
3      x = 20;
4      y = 30;
5      z = 50;
6
7      if ((x - 10) < y) {
8          if (y - 5 > x) {
9              alert (z - x);
10         }
11         else {
12             alert (z - 5);
13         }
14     }
15 }
16 exercise3();

```

4. What will be popped in the alert in this program?

```

1  function exercise4(){
2      var x, y, z;
3      x = 20;
4      y = 30;
5      z = 50;
6
7      if (((x - 10) < y) && (y < z)) {
8          if ((y - 5 > x) && (x - 10 > z)) {
9              alert (z - x);
10         }
11         else {
12             alert(y - z);
13         }
14     }
15     else {
16         alert (z - 5);
17     }

```

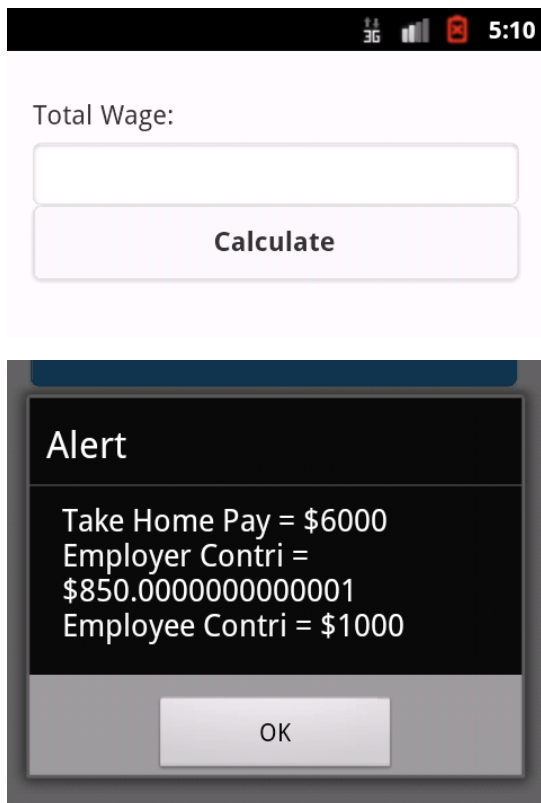
```

18  }
19  exercise4();

```

5. Write a program that calculates CPF (Central Provident Fund) contribution based on an employee's total wage and the current prevailing employer and employee contributions. Note the following rules for CPF contribution rates calculation:

- There is a total wage ceiling of \$5,000 for CPF Contribution. i.e. Neither you nor your employer needs to contribute the wage amount that you have earned beyond \$5,000 to the CPF.
- Employer's contribution is 17% of wages up to \$5,000.
- Employee's contribution is 20% of wages up to \$5,000.
- Take Home Wage = Total Wage – Employee's Contribution



Total Wage:

Calculate

Alert

Take Home Pay = \$6000
 Employer Contr = \$850.00000000000001
 Employee Contr = \$1000

OK

Codes

Codes for this chapter can be found by searching VC7-Condition-Statements on Github.com.