Working with Functions

1. Open the guestbook.html file.
2. Locate the text [ADD CODE HERE] and replace it with the following form. You will use this form to add guests to the <textarea> element.

<form action="" method="post" name="newGuest">

<p>

Guest

<input type="text" name="guestName" style="width: 240px" />&nbsp; <br />

Relationship

<input type="text" name="relationship" style="width: 210px" />

<input type="button" value="Add Guest" /></p>

<p><textarea name="guests"></textarea></p>

</form>

1. Add the following function definition to the script section. The statements within the function definition build a text variable named guestInfo, which contains the guest name and relationship. This string is then assigned as the value of the <textarea> element.

<script type="text/javascript">

function addGuest() {

var guestInfo = document.newGuest.guestName.value + “, “;

guestInfo + = document.newGuest.relationship.value ;

document.newGuest.guests.value = guestInfo;

}

</script>

Calling Functions

Next you will add a function call to the onclick event handler of the Add Guest button.

1. Locate the button control in the form and add the following onclick event handler:

onclick=”addGuest()”

1. Save and run the file.

Returning a value from the function

1. Return to the guestbook.html file.
2. Modify the addGuest() function definition so that it includes 2 parameters: name and relationship. Then modify the statements that build the guestInfo variable so that they reference the parameters instead of the form values. Also replace the last statement that assigns the guestInfo variable to the <textarea> element with a statement that returns the variable to the function. Your modified code should look like:

function addGuest(name, relationship) {

var guestInfo = name + ", ";

guestInfo += relationship ;

return guestInfo;

}

1. Modify the onclick event handler in the button control so it includes 2 statements. The first statement in the event handler calls the addGuest() function and passes to it the values of the name and relationship fields. The second statement then assigns the returned value to the <textarea> field. Your modified code should look like:

onclick="var newGuest=addGuest(document.newGuest.guestName.value, document.newGuest.relationship.value); document.newGuest.guests.value= newGuest"

1. Save the file and run it. Should run the same way.

Working with global variables

1. Open index.html.
2. Add the following global variables to a script section in the head of the document.

<script type="text/javascript">

var guestsCost = 0;

var limousinesCost = 0;

var liveMusicCost = 0;

var flowersCost = 0;

var totalEstimate = 0;

</script>

1. Save the file.

Working with strings and escape characters

1. Open the guestbook.html file.
2. Modify the statement in the addGuest() method that assigns the relationship argument to the guestInfo variable so that it also includes a carriage return escape sequence as follows:

guestInfo +=relationship + “\r”;

1. Modify the last statement in the button elements onclick event handler so that it adds the new guest string to the existing value in the <textarea> element, as follows:

onclick="var newGuest=addGuest(document.newGuest.guestName.value, document.newGuest.relationship.value); document.newGuest.guests.value=document.newGuest.guests.value + newGuest"

1. Save the file and check that you can add multiple guests.

Arithmetic Operators

Next you will modify the index.html file so that it calculates the cost of a wedding.

1. Open the index.html file.
2. Locate the text [ADD TEXT HERE] and replace it with the following form. This form allows users to enter the number of guests and limousines for a wedding. Notice that the numGuests and numLimousines textboxes use onchange() events to call the calcGuests and calcLimousines() functions, respectively. You will add these functions next.

<form action="" name="details">

<table>

<tr>

<td>

<tr>

<td>Guests<br />($65 each)</td>

<td> <input type="text"

name="numGuests" size="3"

onchange="calcGuests()" /></td>

</tr>

<tr>

<td>Limousines<br />($125 each)</td>

<td>

<input type="text"

name="numLimousines" size="3"

onchange="calcLimousines()" /></td>

</tr>

</table>

</form>

1. Locate the text [ADD ESTIMATE HERE] and replace it with the following form. This form simply displays he calculated estimate.

<form action="" name="estimate">

<p>

Estimated total cost:

<input type="text" name="cost" size="5"

style="border-style: none; border-color: inherit;

border-width: medium; background-color: Transparent" text="0" /></p>

</form>

1. Add the following calcGuests() function to the end of the script section. The first statement subtracts the current guest cost from the totalEstimate variable. The second and third statement calculate the new guest cost and the fourth statement assigns the new estimate to the text box in the estimate form.

function calcGuests() {

totalEstimate -= guestsCost;

guestsCost = document.details.numGuests.value \* 65;

totalEstimate += guestsCost;

document.estimate.cost.value = "€" + totalEstimate;

}

1. Add the following calcLimousines() function to the end of the script section. This function contains the same statements as the calcGuests() function except that it calculates the limousine cost instead.

function calcLimousines() {

totalEstimate -= limousinesCost;

limousinesCost = document.details.numLimousines.value \* 125;

totalEstimate += limousinesCost;

document.estimate.cost.value = "€" + totalEstimate;

}

1. Save the file and test it by entering values in the guests and limousines text boxes.

Comparison and Conditional Operators

1. Next you will add fields and code to the wedding planner form that allow users to select live music and flowers. Conditional operators in associated functions for each field will determine whether to add or subtract the cost of each item.
2. Open the index.html file.
3. Add the following elements and fields to the end of the table in the details form. Radio buttons allow users to select whether or not to include live music and flowers. The radio buttons use onclick() event handlers to call associated functions for each of the radio buttons.

<tr>

<td>Live music<br />

($500)</td>

<td>

<input type="radio" name="music"

onclick="addMusic()" />Yes

<input type="radio" name="music"

checked="checked" onclick="removeMusic()" />No

</td>

</tr>

<tr>

<td>Flowers<br />

($400)</td>

<td>

<input type="radio" name="flowers"

onclick="addFlowers()" />Yes

<input type="radio" name="flowers"

checked="checked" onclick="removeFlowers()" />No </td>

</tr>

1. Add the following global variables above the calcGuests() function. These variables will be used to determine whether the user has selected live music and flowers.

var liveMusic = false;

var flowers = false;

1. Add the following functions to the end of the script section.
2. The addMusic() function uses a conditional operator to determine whether the liveMusic variable is set to false. If it is, then the liveMusicCost variable is assigned a value of 500. If not, then it is assigned a value of 0. The liveMusicCost variable is then assigned to the totalEstimate variable with an addition assignment operator. The last 2 statements assign the liveMusic variable a value of true and the value of the totalEstimate variable to the textbox in the estimate form.
3. The removeMusic function uses the exact same syntax as the addMusic() function, except that it assigns a value of -500 to the liveMusicCost variable, which causes the addition assignment expression to subtract the value from the totalEstimate variable.

function addMusic() {

(liveMusic == false)

? liveMusicCost = 500 : liveMusicCost = 0;

totalEstimate += liveMusicCost;

liveMusic = true;

document.estimate.cost.value = "€" + totalEstimate;

}

function removeMusic() {

(liveMusic == true)

? liveMusicCost = -500 : liveMusicCost = 0;

totalEstimate += liveMusicCost;

liveMusic = false;

document.estimate.cost.value = "€" + totalEstimate;

}

1. Add the following addFlowers() and removeFlowers() functions to the end of the script section. These functions are identical to the addMusic() and removeMusic() functions, except they update the total estimate to include flower costs instead of music costs.

function addFlowers() {

(flowers == false)

? flowersCost = 400 : flowersCost = 0;

totalEstimate += flowersCost;

flowers = true;

document.estimate.cost.value = "€" + totalEstimate;

}

function removeFlowers() {

(flowers == true)

? flowersCost = -400 : flowersCost = 0;

totalEstimate += flowersCost;

flowers = false;

document.estimate.cost.value = "€" + totalEstimate;

}

1. Save the files and test them.

Exercise 1

Many companies normally charge a shipping and handling fee for purchases. Create a web page that allows a user to enter a purchase price in a text box; include a JavaScript function that calculates shipping and handling. Add functionality to the script that adds a minimum shipping and handling fee of 1.50 for any purchase that is less than or equal to 25.00. For any orders over 25.00, add 10% to the total purchase price for shipping and handling, but do not include the 1.50 minimum shipping and handling fee. The formula for calculating a percentage is price \* percent / 100. (use a conditional operator)

After you determine the total cost of the order (purchase plus shipping and handling) display it in an alert dialog box.

Exercise 2

The formula for calculating body mass index (BMI) is weight \* 703 / height2 . For example, if you weigh 200 pounds and are 72 inches tall, then you can calculate your body mass index with this expression: (200 \* 703) / (72 \* 72).

Create a web page that contains three text boxes: one for your weight in pounds, one for your height in inches, and one that will contain the BMI result. Create a script with a function named calcBMI() that performs the calculation using the values in the weight and height boxes and assign the result to the BMI text box. Convert the value to an integer by using the parseInt() function. Reference the text boxes from within the function by using the document object, form name, and name and value attributes of each text box (in other words don’t use function arguments). Perform the calculation by calling the function from an onclick() in a button element.

Exercise 3

Create a function called calcPay that takes as parameters the number of hours worked and an hourly rate (taken in through 2 text boxes initially) and returns the pay. According to the company’s rules, no employee may be paid for more than 60 hours at which point the hours are capped. This validation is done within the function. The pay returned from the function is displayed in a textarea box.

Exercise 4

Create a simple calculator using functions. The calculator must take any two numbers from the user and perform 5 basic arithmetic operations i.e addition, subtraction, multiplication, division and modulus".

.