CHAPTER 4:

Implement a web page that displays a Pope Francis web page.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name= "author" content="John Dean">

<title>Pope Francis</title>

<style>

body {background-color: lightyellow;}

header {background-color: lightcyan;}

header > \* {

padding: 2px;

margin: 2px;

}

h2 {

color: white;

background-color: lightsteelblue;

}

aside {float: right;}

figure {margin: 15px;}

li {list-style-type: circle;}

address {display: inline;}

</style>

</head>

<body>

<header>

<h1>Pope Francis I</h1>

<h3>

Bishop of Rome, Vicar of Jesus Christ, Successor of the

Prince of the Apostles, Supreme Pontiff of the Universal Church,

Primate of Italy, Archbishop and Metropolitan of the Roman

Province, Sovereign of the State of Vatican City, Servant of

the Servants of God.

</h3>

</header>

<article>

<section>

<aside>

<figure>

<img src="pope.jpg" alt="Pope Francis with the people"

width="400" height="230">

</figure>

</aside>

<h2>Quotes:</h2>

<ul>

<li><q>We all have the duty to do good.</q></li>

<li><q>Money must serve, not rule.</q></li>

<li><q>I am a sinner.</q></li>

<li><q>Working for a just distribution of wealth

is not mere philanthropy. It is a moral obligation.</q></li>

</ul>

</section>

<section>

<h2>Interesting facts:</h2>

<ul>

<li>Worked as a bar bouncer to pay for college.</li>

<li>Washes feet of incarcerated women.</li>

<li>Lives in a small apartment behind a gas station.</li>

<li>Orders Domino's take-out every Thursday.</li>

</ul>

</section>

</article>

<footer>

Report problems to the webmaster at

<address>jorge.bergoglio@vaticancity.rlgn</address>

</footer>

</body>

</html>

CHAPTER 5:

Implement a web page that displays an invoice form.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name= "author" content="John Dean">

<title>Invoice Form</title>

<style>

table {border-collapse: collapse;}

caption {font-size: 1.5em; color: purple}

table, th, td {border: thin solid blue;}

th, td {text-align: left; padding: 5px;}

.width1 {width: 40%}

.width2 {width: 25%}

.width3 {width: 15%}

.width4 {width: 20%}

.black {background-color: black}

</style>

</head>

<body>

<table role="presentation">

<caption>Customer Order Form</caption>

<tr><td colspan="4">Order Number</td></tr>

<tr><td colspan="4">Order Date</td></tr>

<tr>

<th class="width1">Product Name</th>

<th class="width2">Color</th>

<th class="width3">Quantity</th>

<th class="width4">Price</th>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr><td class="black" colspan="4">&nbsp;</td></tr>

<tr><td colspan="4">Subtotal</td></tr>

<tr><td colspan="4">Tax</td></tr>

<tr>

<td colspan="3">Shipping</td>

<td>5.95</td>

</tr>

<tr><td colspan="4">Total</td></tr>

</table>

</body>

</html>

CHAPTER 7:

Create a child's animal alphabet web page. For extra credit, find a kids-oriented, fun web font and use it for the text in your web page.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="author" content="John Dean">

<title>Animal Alphabet Page</title>

<link rel="icon" href="../images/jd.gif">

<style>

h2 > a {text-decoration: none;}

h2 > a:hover {background-color: lightgreen;}

.table {display: table;}

.row {display: table-row;}

.row > \* {

display: table-cell;

vertical-align: middle;

text-align: center;

}

h1 {font-size: 4em;}

/\* EXTRA CREDIT: \*/

@font-face {

font-family: Kids;

src: url(../fonts/KBMaRkerFacTorY.wof) format("woff"),

url(../fonts/KBMaRkerFacTorY.ttf) format("opentype");

}

\* {font-family: Kids, serif;}

</style>

</head>

<body>

<h1><a id="top">Animal Alphabet</a></h1>

<h2>

<a href="#A">A</a> |

<a href="#B">B</a> |

<a href="#C">C</a> |

<a href="#D">D</a> |

<a href="#E">E</a>

</h2>

<div class="table">

<div id="A" class="row">

<h4>A is for Aardvark:</h4>

<span>

<img src="../images/aardvark.jpg" alt="aardvark"

width="320" height="158">

</span>

<span>

<img src="../images/underConstruction.gif" alt="under construction"

width="53" height="47">

</span>

</div>

<div id="B" class="row">

<h4>B is for Bear:</h4>

<span>

<img src="../images/bear.jpg" alt="bear" width="135" height="135">

</span>

<span>

<audio preload controls>

<source src="../audio/grizzlyBear.ogg" type="audio/ogg">

<source src="../audio/grizzlyBear.mp3" type="audio/mpeg">

</audio>

</span>

</div>

<div id="C" class="row">

<h4>C is for Cat:</h4>

<span>

<img src="../images/cat.gif" alt="cat" width="150" height="117">

</span>

<span>

<img src="../images/underConstruction.gif" alt="under construction"

width="53" height="47">

</span>

</div>

<div id="D" class="row">

<h4>D is for Dog:</h4>

<span>

<img src="../images/dog.gif" alt="dog" width="168" height="216">

</span>

<span>

<img src="../images/underConstruction.gif" alt="under construction"

width="53" height="47">

</span>

</div>

<div id="E" class="row">

<h4>E is for Elephant:</h4>

<span>

<img src="../images/elephant.gif" alt="elephant"

width="246" height="210">

</span>

<span>

<img src="../images/underConstruction.gif" alt="under construction"

width="53" height="47">

</span>

</div>

</div>

<br>

<h4><a href="#top">Go back to the top</a></h4>

</body>

</html>

CHAPTER 9:

Implement a web page that generates a restaurant bill.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="author" content="John Dean">

<title>Restaurant Bill Calculator</title>

<style>

.table {display: table;}

.row {display: table-row;}

.row > \* {display: table-cell;}

.table {border-spacing: 5px;}

.row > :first-child {text-align: right;}

input:valid:focus {background-color: lightgreen;}

input:invalid:focus {background-color: rgb(255, 220, 220);}

</style>

<script>

// This function calculates and displays the restaurant

// bill, including menu price, tax, tip, and total.

function calculateBill(form) {

var taxBox; // tax percentage applied to price

var priceBox; // combined price as shown on menu

var tipBox; // tip percentage applied to tax + price

var totalPayment; // total bill that needs to be paid

var error = ""; // error message

var price; // numeric value extractd from the price box

var tax; // calculated tax added to the bill

var tip; // calculated tip added to the bill

taxBox = form.elements["tax-percent"];

priceBox = form.elements["price"];

tipBox = form.elements["tip-percent"];

bill = form.elements["bill"];

if (!taxBox.checkValidity()) {

error += "Invalid entry for tax percent." +

" Must be a number between " + taxBox.min + " and " +

taxBox.max + ".\n\n";

}

if (!priceBox.checkValidity()) {

error += "Invalid entry for price." +

" Must be a number between " + priceBox.min + " and " +

priceBox.max + ".\n\n";

}

if (!tipBox.checkValidity()) {

error += "Invalid entry for tip percent." +

" Must be an integer between " + tipBox.min + " and " +

tipBox.max + ".\n\n";

}

if (error != "") {

document.getElementById("bill").innerHTML = "";

// The following code deletes a blank line at the bottom of

// the alert dialog.

alert(error.substring(0, error.length - 2));

}

else {

price = priceBox.valueAsNumber;

tax = price \* (taxBox.valueAsNumber \* .01);

tip = (price + tax) \* (tipBox.valueAsNumber \* .01);

totalPayment = price + tax + tip;

document.getElementById("bill").innerHTML =

"Price: $" + price.toFixed(2) + "<br>" +

"Tax: $" + tax.toFixed(2) + "<br>" +

"Tip: $" + tip.toFixed(2) + "<br>" +

"Total: $" + totalPayment.toFixed(2);

}

} // end calculateBill

</script>

</head>

<body>

<h1>Restaurant Bill Calculator</h1>

<form class="table" id="myForm">

<div class="row">

<label for="tax-percent">Tax percent:</label>

<input type="number" id="tax-percent"

min="0" max="100" step=".01" required> %

</div>

<div class="row">&nbsp;</div>

<div class="row">

<label for="price">Price from the menu:</label>

<input type="number" id="price"

min="0" max="1000" step=".01" required>

</div>

<div class="row">

<label for="tip-percent">Tip percent:</label>

<input type="number" id="tip-percent"

min="0" max="200" step="1" required> %

</div>

</form>

<br>

<div>

<input type="button" form="myForm" value="Calculate bill"

onclick="calculateBill(this.form);">

<br><br>

<div id="bill"></div>

</div>

</body>

</html>

CHAPTER 10:

Implement a web page that generates a recipe for a root beer float.

rootBeerFloat.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="author" content="John Dean">

<title>Root Beer Float Maker</title>

<style>

.table {

display: table;

border-spacing: 10px 0;

}

.cell {

display: table-cell;

vertical-align: top;

}

fieldset {

display: inline;

vertical-align: top;

}

#recipe {

display: none;

color: white;

background-color: rgb(129,77,48); /\* root beer color \*/

border: 10px double;

width: 200px;

padding: 15px;

}

</style>

<script src="rootBeerFloat.js"></script>

</head>

<body>

<h1>Root Beer Float Maker</h1>

<div class="table">

<form class="cell">

<fieldset>

<legend>Ice Creams</legend>

<input type="checkbox" name="iceCreams" value="vanilla">vanilla<br>

<input type="checkbox" name="iceCreams" value="chocolate">chocolate<br>

<input type="checkbox" name="iceCreams" value="coffee">coffee

</fieldset>

&nbsp;&nbsp;

<fieldset>

<legend>Root Beers</legend>

<input type="radio" name="rootBeer" value="Barq's">Barq's<br>

<input type="radio" name="rootBeer" value="A&amp;W">A&amp;W<br>

<input type="radio" name="rootBeer" value="Dad's">Dad's<br>

<input type="radio" name="rootBeer" value="IBC">IBC

</fieldset>

<br><br>

<input type="button" value="I'm ready to make my float!"

onclick="process(this.form);">

<p id="recipe">

Recipe:<br>

In a glass mug, add one scoop each of these ice creams -

<span id="iceCreams"></span>.

Fill up the mug with <span id="rootBeer"></span> root beer. Enjoy!

</p>

</form>

<span class="cell">

<img src="rootBeerFloat.png"

width="240" height="315" alt="root beer float">

</span>

</div>

</body>

</html>

rootBeerFloat.js:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* rootBeerFloat.js

\* John Dean

\*

\* This file contains a function that supports the

\* root beer float web page.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Read the user's ingredient selections and display

// all the selections in the form of a recipe.

function process(form) {

var iceCreamCBs; // collection of check boxes

var iceCreams = ""; // a list of ice creams for recipe

var rootBeer; // user's selected root beer

var recipe; // recipe element

iceCreamCBs = form.elements["iceCreams"];

for (let i=0; i<iceCreamCBs.length; i++) {

if (iceCreamCBs[i].checked) {

iceCreams += ", " + iceCreamCBs[i].value;

}

} // end for

if (iceCreams.charAt(0) == ",") {

iceCreams = iceCreams.substring(2);

}

// The following value property works with Chrome

// and FireFox, but not IE.

rootBeer = form.elements["rootBeer"].value;

if (iceCreams == "" || rootBeer == "") {

alert("You must select at least one ice cream" +

" flavor, and you must select one root beer brand.");

}

else {

recipe = document.getElementById("recipe");

// Make visible the p element that stores the recipe.

recipe.style.display = "block";

document.getElementById("iceCreams").innerHTML = iceCreams;

document.getElementById("rootBeer").innerHTML = rootBeer;

}

} // end process

CHAPTER 12:

Implement a web page that displays a beryllium hydride molecule. For extra credit, implement a web page that displays an oxygen molecule.

molecule.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name= "author" content="John Dean">

<title>molecule</title>

<style>

canvas {

// border: thin solid black;

}

</style>

<script src="molecule.js"></script>

</head>

<body onload="drawBeH2();">

<!-- <body onload="drawOxygen();"> -->

<canvas id="canvas" width="790" height="510">

Sorry - This page uses <code>canvas</code> and

your browser doesn't support it.

</canvas>

</body>

</html>

molecule.js:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* molecule.js

\* John Dean

\*

\* This file draws a beryllium hydride molecule and

\* an oxygen molecule.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// This function draws the BeH2 molecule.

function drawBeH2() {

// The diameter of the hydrogen atom (should be a multiple of 2)

var H\_DIA = 40;

// Distance between window's edges, drawn rectangle, and drawn string.

var SPACER = 60;

var ctx; // the canvas object's context

ctx = document.getElementById("canvas").getContext("2d");

centerX = SPACER + 4 \* H\_DIA;

ctx.textAlign = "center";

ctx.font = "xx-large 'Times New Roman', serif";

ctx.fillText("beryllium hydride", centerX, SPACER);

ctx.fillStyle = "lightgray";

ctx.fillRect(SPACER, 2 \* SPACER, 8 \* H\_DIA, 4 \* H\_DIA);

// Set interior colors of the atoms.

ctx.fillStyle = "yellow";

ctx.beginPath();

ctx.arc(SPACER + 1.5 \* H\_DIA, 2 \* SPACER + 2 \* H\_DIA,

H\_DIA / 2, 0, 2 \* Math.PI);

ctx.fill();

ctx.beginPath();

ctx.arc(SPACER + 6.5 \* H\_DIA, 2 \* SPACER + 2 \* H\_DIA,

H\_DIA / 2, 0, 2 \* Math.PI);

ctx.fill();

ctx.fillStyle = "aqua";

ctx.beginPath();

ctx.arc(SPACER + 4 \* H\_DIA, 2 \* SPACER + 2 \* H\_DIA,

H\_DIA, 0, 2 \* Math.PI);

ctx.fill();

ctx.strokeRect(SPACER, 2 \* SPACER, 8 \* H\_DIA, 4 \* H\_DIA);

ctx.beginPath();

ctx.arc(SPACER + 1.5 \* H\_DIA, 2 \* SPACER + 2 \* H\_DIA,

H\_DIA / 2, 0, 2 \* Math.PI);

ctx.stroke();

ctx.beginPath();

ctx.arc(SPACER + 6.5 \* H\_DIA, 2 \* SPACER + 2 \* H\_DIA,

H\_DIA / 2, 0, 2 \* Math.PI);

ctx.stroke();

ctx.beginPath();

ctx.arc(SPACER + 4 \* H\_DIA, 2 \* SPACER + 2 \* H\_DIA,

H\_DIA, 0, 2 \* Math.PI);

ctx.stroke();

// Draw molecular bond lines.

ctx.lineWidth = 4;

ctx.beginPath();

ctx.moveTo(SPACER + 2 \* H\_DIA, 2 \* SPACER + 2 \* H\_DIA);

ctx.lineTo(SPACER + 3 \* H\_DIA, 2 \* SPACER + 2 \* H\_DIA);

ctx.stroke();

ctx.beginPath();

ctx.moveTo(SPACER + 5 \* H\_DIA, 2 \* SPACER + 2 \* H\_DIA);

ctx.lineTo(SPACER + 6 \* H\_DIA, 2 \* SPACER + 2 \* H\_DIA);

ctx.stroke();

} // end drawBeH2

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// EXTRA CREDIT:

// This function draws the oxygen molecule.

function drawOxygen() {

// The radius of the hydrogen atom

var RADIUS = 100;

// Distance between window's edges, drawn rectangle, and drawn string.

var SPACER = 40;

// the angle in radians up to the top covalent bond

var ANGLE = .35;

// horizontal distance from center of atom to edge of bond

var xOffset;

// vertical distance from center of atom to edge of bond

var yOffset;

var ctx; // the canvas object's context

ctx = document.getElementById("canvas").getContext("2d");

centerX = SPACER + 3.5 \* RADIUS;

ctx.textAlign = "center";

ctx.font = "xx-large 'Times New Roman', serif";

ctx.fillText("oxygen", centerX, SPACER);

ctx.fillStyle = "lightgray";

ctx.fillRect(SPACER, 2 \* SPACER, 7 \* RADIUS, 4 \* RADIUS);

// Set interior color of the atoms.

ctx.fillStyle = "green";

ctx.beginPath();

ctx.arc(SPACER + 2 \* RADIUS, 2 \* SPACER + 2 \* RADIUS,

RADIUS, 0, 2 \* Math.PI);

ctx.fill();

ctx.beginPath();

ctx.arc(SPACER + 5 \* RADIUS, 2 \* SPACER + 2 \* RADIUS,

RADIUS, 0, 2 \* Math.PI);

ctx.fill();

// Draw black lines for rectangle and atom borders.

ctx.fillStyle = "black";

ctx.strokeRect(SPACER, 2 \* SPACER, 7 \* RADIUS, 4 \* RADIUS);

ctx.beginPath();

ctx.arc(SPACER + 2 \* RADIUS, 2 \* SPACER + 2 \* RADIUS,

RADIUS, 0, 2 \* Math.PI);

ctx.stroke();

ctx.beginPath();

ctx.arc(SPACER + 5 \* RADIUS, 2 \* SPACER + 2 \* RADIUS,

RADIUS, 0, 2 \* Math.PI);

ctx.stroke();

// Draw molecular bond lines.

ctx.lineWidth = 10;

xOffset = RADIUS - Math.round(Math.cos(ANGLE) \* RADIUS);

yOffset = Math.round(Math.sin(ANGLE) \* RADIUS);

ctx.beginPath();

ctx.moveTo(SPACER + 3 \* RADIUS - xOffset,

2 \* SPACER + 2 \* RADIUS - yOffset);

ctx.lineTo(SPACER + 4 \* RADIUS + xOffset,

2 \* SPACER + 2 \* RADIUS- yOffset);

ctx.stroke();

ctx.beginPath();

ctx.moveTo(SPACER + 3 \* RADIUS - xOffset,

2 \* SPACER + 2 \* RADIUS + yOffset);

ctx.lineTo(SPACER + 4 \* RADIUS + xOffset,

2 \* SPACER + 2 \* RADIUS + yOffset);

ctx.stroke();

} // end drawOygen