

```

119 // JavaScript statements include conditionals and loops using the syntax
120 // of C, C++, Java, and other languages.
121 function abs(x) { // A function to compute the absolute value
122   if (x >= 0) { // The if statement...
123     return x; // executes this code if the comparison is true.
124   } // This is the end of the if clause.
125   else { // The optional else clause executes its code if
126     // the comparison is false.
127   } // curly braces optional when 1 statement per clause.
128 } // Note return statements nested inside if/else.
129 }
130
131 function factorial(n) { // A function to compute factorials
132   var product = 1; // Start with a product of 1
133   while(n > 1) { // Repeat statements in {} while expr in () is true
134     product *= n; // Shortcut for product = product * n;
135     n--; // Shortcut for n = n - 1
136   } // End of loop
137   return product; // Return the product
138 }
139 factorial(4) // => 24: 1*4*3*2
140
141 function factorial2(n) { // Another version using a different loop
142   var i, product = 1; // Start with 1
143   for(i=2; i <= n; i++) // Automatically increment i from 2 up to n
144     product *= i; // Do this each time. {} not needed for 1-line loops
145   return product; // Return the factorial
146 }
147 factorial2(5) // => 120: 1*2*3*4*5
148
149 // Define a constructor function to initialize a new point object
150 function Point(x,y) { // By convention, constructors start with capitals
151   this.x = x; // this keyword is the new object being initialized
152   this.y = y; // Store function arguments as object properties
153 } // No return is necessary
154 // Use a constructor function with the keyword "new" to create instances
155 var p = new Point(1, 1); // The geometric point (1,1)
156 // Define methods for Point objects by assigning them to the prototype
157 // object associated with the constructor function.
158 Point.prototype.r = function() {
159   return Math.sqrt( // Return the square root of x^2 + y^2
160     this.x * this.x + // This is the Point object on which the method...
161     this.y * this.y // ...is invoked.
162   );
163 };
164 // Now the Point object p (and all future Point objects) inherits the method r()
165 p.r() // => 1.414...
166
167 <script>
168 function moveon() {
169 // Display a modal dialog to ask the user a question
170 var answer = confirm("Ready to move on?");
171 // If they clicked the "OK" button, make the browser load a new page
172 if (answer) window.location = "http://google.com";
173 }
174 // Run the function defined above 1 minute (60,000 milliseconds) from now.
175 setTimeout(moveon, 60000);
176 </script>
177

```

```

178 // Display a message in a special debugging output section of the document.
179 // If the document does not contain such a section, create one.
180 function debug(msg) {
181 // Find the debugging section of the document, looking at HTML id attributes
182 var log = document.getElementById("debuglog");
183 // If no element with the id "debuglog" exists, create one.
184 if (!log) {
185   log = document.createElement("div"); // Create a new <div> element
186   log.id = "debuglog"; // Set the HTML id attribute on it
187   log.innerHTML = "<h1>Debug Log</h1>"; // Define initial content
188   document.body.appendChild(log); // Add it at end of document
189 }
190 // Now wrap the message in its own <pre> and append it to the log
191 var pre = document.createElement("pre"); // Create a <pre> tag
192 var text = document.createTextNode(msg); // Wrap msg in a text node
193 pre.appendChild(text); // Add text to the <pre>
194 log.appendChild(pre); // Add <pre> to the log
195
196
197 function hide(e, reflow) { // Hide the element e by scripting its style
198   if (reflow) { // If 2nd argument is true
199     e.style.display = "none" // hide element and use its space
200   }
201   else { // Otherwise
202     e.style.visibility = "hidden"; // make e invisible, but leave its space
203   }
204 }
205 function highlight(e) { // Highlight e by setting a CSS class
206 // Simply define or append to the HTML class attribute.
207 // This assumes that a CSS stylesheet already defines the "hilitte" class
208 if (!e.className) e.className = "hilitte";
209 else e.className += " hilitte";
210 }

```