Strange things about javascript:

- Variable naming: cannot use dash in a variable name (it’s interpreted like minus)
- Compound assignment statements:
  - `var firstName = "Ray", lastName = "Harris";`
- Using the “new” keyword (page 63), examples that create new objects and store them in variables
  - `var today = new Date(); // creates a Date object named today`
  - `var states = new Array(); // creates an Array object named states`
- Object chaining (page 63)
  - `alert( document.getElementById("rate").value );`
  - `alert( window.location.toLowerCase() );`
- “number” is a class (like String) and so you can use number methods (pg 67):
  - `var balance = 2384.55678; // creates a number object named balance`
  - `alert ( balance.toFixed(2) ); // displays 2384.56`
  - `// balance is still 2384.55678`
  - `balance = parseFloat( balance.toFixed(2) ); // balance is now 2384.56`
- How to use the disabled property to enable a text box (pg 69)
  - `document.getElementById("salesTax").disabled = false;`
- How to use the focus method to move the cursor to a text box (pg 69)
  - `document.getElementById("investment").focus();`
- Strange way you can define a function (using var) (pg 77)
  ```javascript
  var calculateTax = function ( subtotal, taxRate ) {
    var tax = subtotal * taxRate;
    return tax;
  };
  ```
- Strange way that you can assign functions to events:

```html
<html>
<head>
  <title>JavaScript Event Handler</title>
  <script type="text/javascript">
  // Returns a reference to the HTML object that has the specified id.
  var $ = function ( id ) {
    return document.getElementById( id );
  }
  // A regular function definition
  var display_click = function () {
    alert( "You clicked the button." );
  };
  // This is the event handler for the onload event of the page.
  // It is executed after the page is loaded and the DOM has been built.
  window.onload = function () {
    // This statement assigns the event handler named display_click
    // to the onclick event of the XHTML object named btnDisplay
    $('"btnDisplay").onclick = display_click;
  };
</script>
</head>
<body>
  <p><input type="button" value="Display Message" id="btnDisplay" /></p>
</body>
</html>
```
Creating your own objects:

As with any objects in JavaScript, you can add properties to Objects dynamically:

```javascript
var robot = { }; // a new empty object
robot.name = "Zephyr";
robot.model = "Guard";
robot.hasJetpack = true;
```

You can also add functions dynamically to an object. The following code extends the previous simple example by adding a method to the robot object. We first define the function and then add it to the object:

```javascript
function strikeIntruder() {
    alert("ZAP!");
}
robot.attack = strikeIntruder();
```

Or more compactly, like this:

```javascript
robot.attack = function strikeIntruder() {
    alert("ZAP!");
};
```

Or even more compactly, like this:

```javascript
var robot = {
    name: "Zephyr ",
    model: "Guard",
    hasJetpack: true,
    attack: function() { alert("ZAP!"); }
};
```

You can have objects inside of other objects like this:

```javascript
var robot = {
    name: null,
    model: "Guard",
    hasJetpack: true,
    attack: function() { alert("ZAP!"); },
    sidekick: {
        name: "Spot",
        model: "Dog",
        hasJetpack: false,
        attack: function() { alert("CHOMP!"); }
    }
};
```
Declaring Arrays

```javascript
var firstArray = []; // array with no elements
var secondArray = ["red", "green", "blue"];  
var thirdArray = [5, 6];  
// mixed types is OK
var fourthArray = [23.2, "green", 4];
```

JavaScript provides associative arrays. These two expressions are equivalent (the property name can be used like an array index):

```javascript
object.property
object["property"]
```

This means you can use an "id" instead of an integer index, like this:

```javascript
<script>
var customers = {}; // {} // new Object(); // new Array(); // all work
customers["Tom Doe"] = {age: 42, title: "manager"};
customers["Sylvia Cheung"] = {age: 56, title: "secretary"};
customers["George Speight"] = {age: 33, title: "guard"};

for (var client in customers) {
    console.log("Customer " + client + " is a " +
            customers[client].age + " year old " + customers[client].title);
    console.log(" ");
}
</script>
```