

## Lesson 2: Input and Variables

---

**OBJECTIVES: In this lesson you will learn**

- How to include data in your script by using a variable
- How to declare (create) a variable
- How to name a variable (rules for identifiers)
- How to assign a value to a variable using =
- How to combine strings using + and string concatenation
- How to use the prompt statement to collect information from the user
- How to display (output) the contents (value) of a variable
- About string formatting methods

## Preparing to Program

A critical feature of programming is the ability to represent data (information). If a computer cannot process information, it is not much use. Programming languages use variables to represent information. A variable is an idea you may remember from algebra: it is a name that stands for a value. Since that value can change (vary) over time, it is called a variable. It does not mean it has to change, it just means that it may change. In programming, a variable is a container that holds information used by a program. In algebra, you learned that variables like  $x$  and  $y$  can hold numbers. In JavaScript programs, variables can also hold strings (character data) or Boolean (logical) values.

### Declaring Variables

The first step requires that you create a variable. This is called a **variable declaration**. In JavaScript, a variable declaration looks like this:

```
var myName;
```

**var** is a keyword that indicates this is a variable declaration. This line of code says that “myName” is the name of the variable. In programming, the name of a variable is called an **identifier**. All programming languages have syntax rules that restrict what can be a legal identifier. In JavaScript, the first character must be either a letter or an underscore (`_`). The remaining characters may include numbers, along with letters or an underscore. You cannot include a blank space or any other special characters like dash (`-`). You may also not use any JavaScript reserved words as an identifier (see Appendix A for a list of JavaScript reserved words). Identifiers are case sensitive, so “myName” and “MyName” are treated as different identifiers. Mis-spellings caused by the wrong case are usually difficult to spot, so always be careful when creating and using variable names.

### Assigning Values to a Variable

Since a variable is a container for a value, you need to know how to put something into that container. You give a value to a variable using this syntax:

```
myName = "Fred";
```

The equal sign is called the **assignment operator**. Notice the value is copied from the right to the left. This is opposite of how you used the equal sign in your math classes. In most programming languages, values are assigned from the right into the left. The opposite direction is an error:

```
"Fred" = myName;
//this is an error, the variable name must be on the left
```

You can also declare a variable and assign it a value at the same time:

```
var anotherName = "Sally";
```

You can also declare a variable and assign it with a numeric value. Here are some examples:

```
var number = 100;  
  
var taxRate = .0825;
```

## Using prompt and Variables

An important feature that JavaScript adds to your Web page is interactivity: you can ask users questions, and their answers can determine how the Web page is displayed. The **prompt method** is used to ask a question and to store the answer entered by the user in a variable. Here is the basic syntax for the prompt method:

### Syntax:

```
var varname = prompt("your text", "default entry")
```

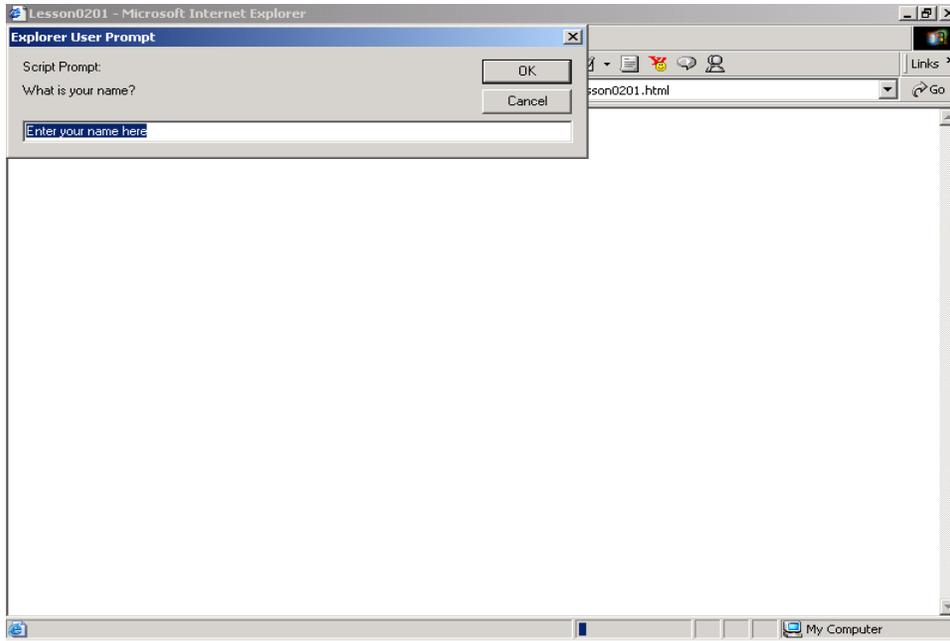
You replace *varname* with a variable name, and replace “your text” with the message displayed to the user (usually a question). The second entry, “default entry” sets a default value for the user. Often this is left as “”, which is nothing. However, it could be set to something else if you would like to have a default answer ready for the user to see. When the user sees this dialog box, enters a value and presses enter, the string the user has entered is assigned (copied) into the variable. The variable now has the value of the user’s input.

The entries inside the parentheses of a method are known as **parameters**. Remember, in the Introduction you learned about the idea of a function or method, which is a mini-program that carries out a specific task. Parameters are information that a method or function uses when carrying out its specialized tasks. You will learn more about parameters, functions, and methods in Lesson 6.

Here is an example of a prompt statement:

```
var myName = prompt("What is your name?", "Enter your name here");
```

When executed, this code displays the following dialog box:



This week in the lab you will use the prompt method to collect information from the user and make use of it in your programs.

### Using Variables With document.write

Once a variable has a value, you can use document.write to display it on a page. It is important to distinguish between the name of a variable and what its contents are. The name of a variable is its identifier. When you refer to a variable in a line of code, you use the name. Variables are stand-ins or placeholders for a value. When the code actually executes and the variable is needed to do something, its value is used. For example, given the following declaration:

```
var myName = "Sam";
```

myName is the name of the variable, and Sam is the value or contents of the variable. When you use a variable in a document.write statement, the contents of the variable are displayed on the page. For example, the statement

```
document.write(myName);
```

displays the name Sam on the page. Notice there are no quote marks surrounding myName. If you were to include quote marks, then the contents of the quote marks, not the contents of the variable would be displayed. The basic rule controlling output is this: characters inside quotes are printed; characters not inside quotes are considered variables, and the value of the variable is printed.

For example:

```
document.write("myName");
```

displays myName on the page, not the name Sam.

### Combining Strings Using Concatenation

**Concatenation** is an operation that combines strings. The + operator, when applied between two strings, combines them into one string. For example:

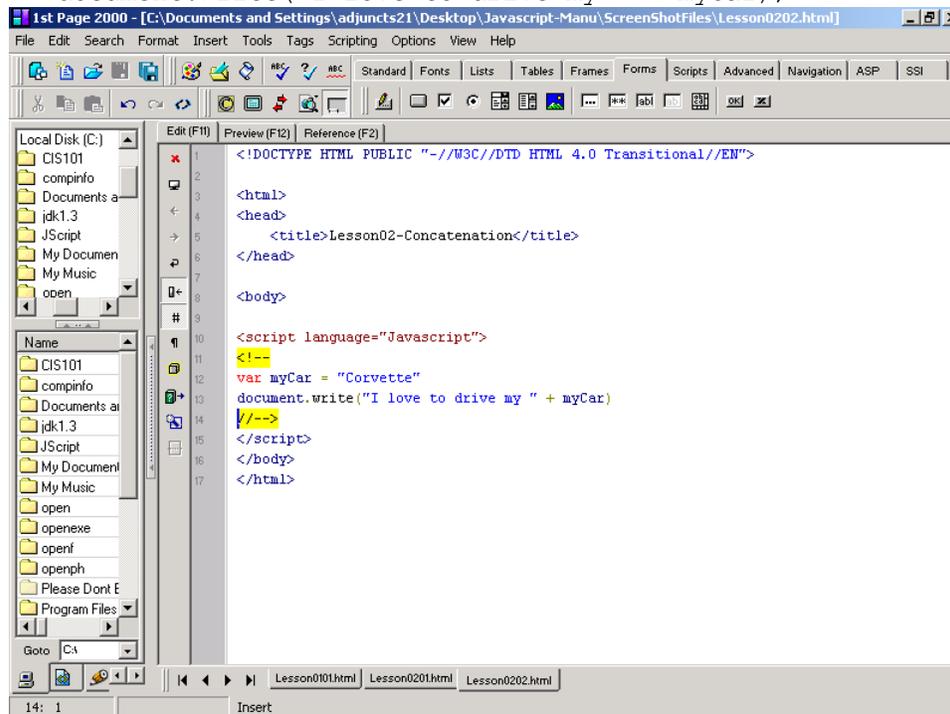
```
var part1 = "This sentence ";
var part 2 = "has ";
var part 3 = "three pieces.";
var sentence = part1 + part2 + part3;
```

Once all three parts are combined, the variable sentence has the value "This sentence has three pieces." Notice you need to include blank spaces within the string. If you omit them, they words will run together.

You can also use concatenation to combine strings inside quotes along with variables. For example:

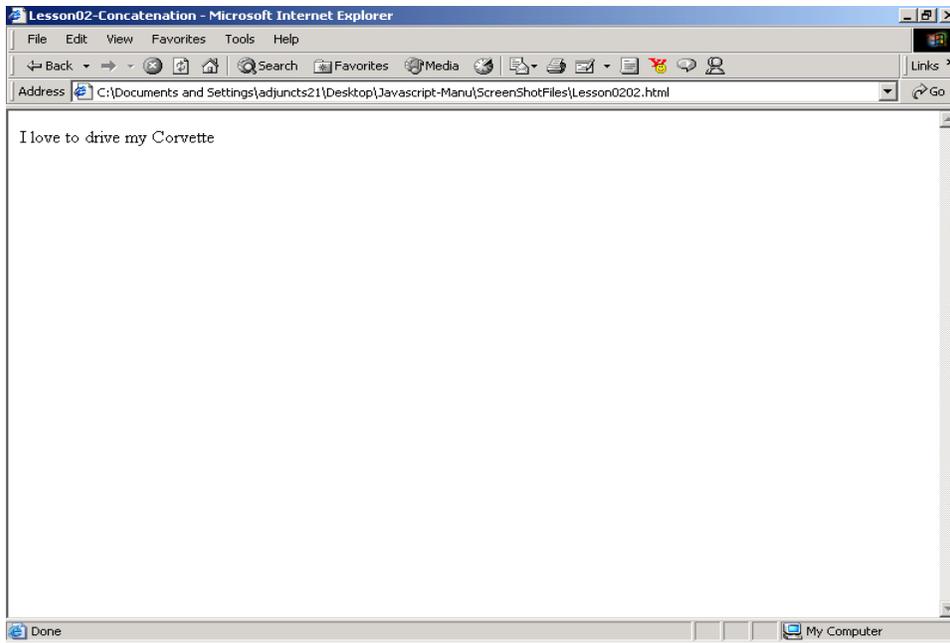
```
var myCar = "Corvette";

document.write("I love to drive my " + myCar);
```



which displays the following

## 2-6| Lesson 2: Input and Variables



You will use concatenation in this lesson to combine variables containing values input by the user in a display message.

## In the Lab

This week in the lab you will add interactivity to your Web pages by introducing variables and user input with the prompt method. This will allow you to customize the appearance of your page. We will also try out a simple version of Mad Libs, a word game that creates mangled sentences.

Start 1<sup>st</sup> Page 2000 and begin a new HTML document. Save it giving it the name lesson0201.html. Following the instructions outlined in Appendix B, place your cursor between the <BODY> ... </BODY> tags, and insert the SCRIPT tags and hiding comments by using the Scripting menu.

Now type in *exactly* the following code:

```
<script language="Javascript">

<!--

var name=prompt("Please enter your name:"," your name");

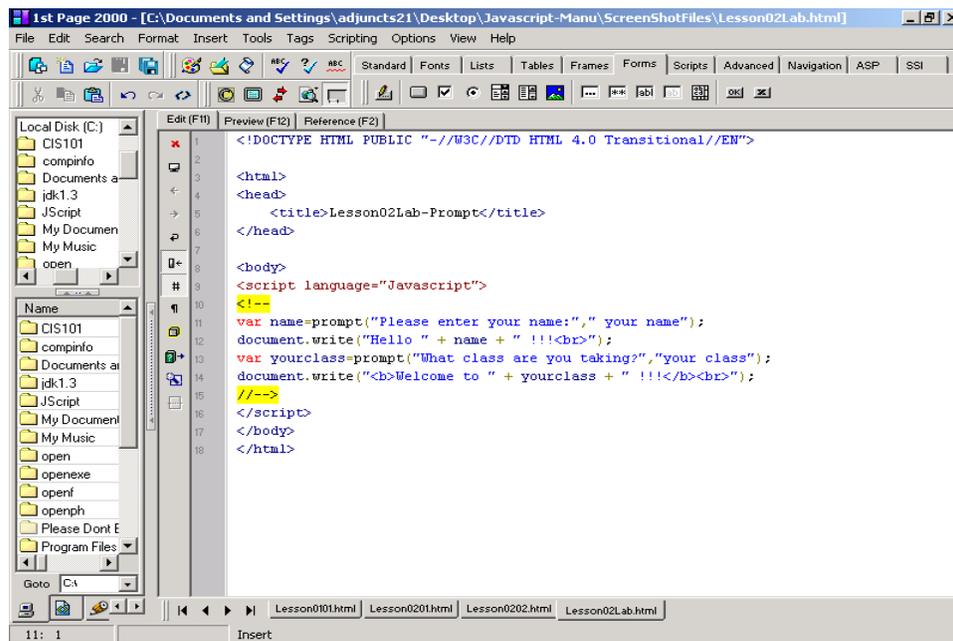
document.write("Hello " + name + " !!!<br>");

var yourClass=prompt("What class are you taking?","your class");

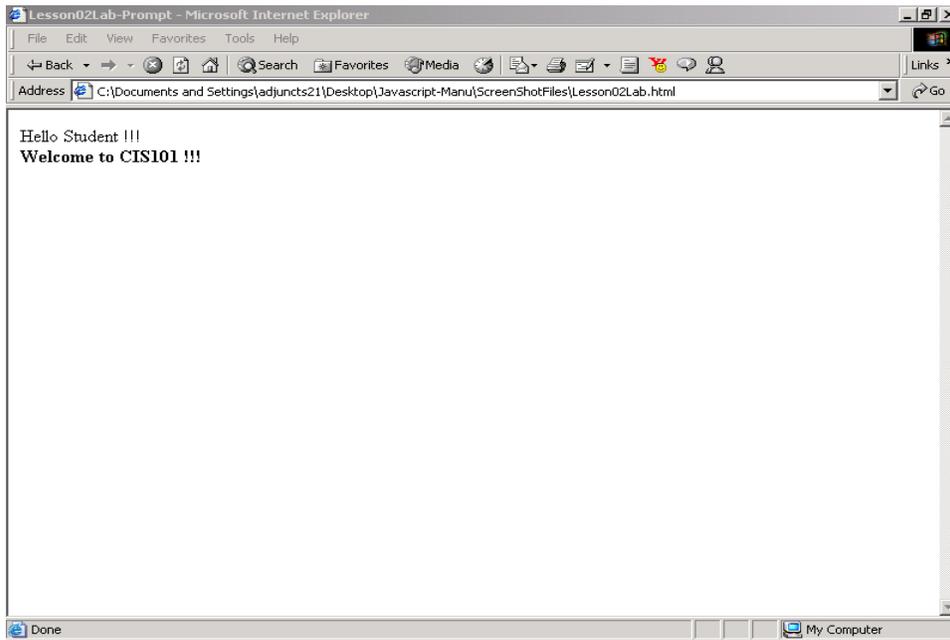
document.write("<b>Welcome to " + yourClass + " !!!</b><br>");

//-->

</script>
```



After you have typed in the above code, run it using the preview button. If you have any errors, correct your code until you have output that looks like the following:



### Student Modifications

Make the following modifications to the code:

- In the prompt method set the value of the second parameter to "", which means empty. This takes away the default entry. Run the code again, and do not input a value, just hit the enter key. What is displayed?
- Create another variable called myMovie. Ask the user to input their favorite movie, and store their answer in myMovie. Write a document.write statement that displays the variable.
- Use HTML format tags to change the format of a variable. Remember, the HTML tags need to be inside quote marks, so you must use concatenation to do this.

### Having Some Fun With Variables

Save your work from the previous exercise. Start a new HTML document in 1<sup>st</sup> Page 2000, add the SCRIPT tags using the Scripting menu, and save this file as lesson0202.html.

A Mad Lib is a popular party activity where a potentially humorous story is written down, with blanks in the place of some important words. Before reading the story, the storyteller asks others present to fill in those blanks. Those selecting the words are only told the type of word required, and have no other information about the story. This lack of context in selecting words can result in an entertaining story when the words are plugged in the appropriate places.

The following JavaScript code uses variables and the prompt method to create a Mad Lib.

Enter the following code:

```
<html>
<head>
  <title>Simple Mad Lib</title>
</head>
<body>
<script language="Javascript">
<!--
var name = prompt("Give me a name: ", "");
var verb = prompt("Give me a past-tense verb: ", "");
var adjective= prompt("Give me an adjective: ", "");
var sentence = name + " " + verb +
  " to the museum, and the monkey was " + adjective + ".<p>";
document.write(sentence);
//-->
</script>
</body>
</html>
```

After entering this code, run it a few times with different words. Check your blank spaces, and be sure the words are spaced properly.

### Student Modifications

You have used embedded HTML to change the format of string output. JavaScript also has **string formatting methods** that can simplify your output formatting.

For example, add the following code after the first document.write statement (line 17):

```
document.write(sentence.bold());
```

Type this code in and run it. You will see it displays sentence in bold. This method performs the task of adding <B> to the front of sentence and </B> to the end of sentence.

Here are some additional string methods:

```
string.blink()
```

```
string.fontcolor(colorValue)
```

## 2-10| Lesson 2: Input and Variables

```
string.fontSize(integer1to7)
string.italics()
string.big()
string.small()
```

Try some or all of these methods in your Mad Lib. For example, to use italics, replace string with sentence, i.e.

```
document.write(sentence.italics());
```

If you want to use fontcolor, you will need to include a color as a parameter. Select a color from the list in Appendix C. If you want to use fontsize, you will need to include a number between 1 and 7 as a parameter.

## Key Terms and Definitions

- **variable declaration** – a statement that creates and names a new variable.
- **var** – JavaScript keyword used to create a new variable in a variable declaration.
- **identifier** – the name of a part of your program, like a variable, function or method. In JavaScript, an identifier must begin with either an underscore or a letter, and may only contain letters, numbers or an underscore.
- **assignment operator** – equal sign (=), used to give (assign) a value to a variable. Values are always assigned from right to left.
- **prompt method** – JavaScript method that asks the user for input and stores the answer in a variable.
- **parameters** – data inside the parentheses portion of a function or method. Functions use values in parameters when carrying out their specific tasks.
- **concatenation** – the process of combining strings using the + operator.
- **string formatting methods** – methods that are used to change the format (appearance) of a string.

## Lesson 2 Summary

You learned how to declare and name a variable using the JavaScript rules for identifiers. You also assigned values to a variable with the assignment operator. You queried the user for input, and stored the user's response in a variable with the prompt method. You learned how to combine strings using concatenation and the + operator. You also learned how to display the value of a variable with document.write. Finally you learned how to use string formatting methods to alter the appearance of a string.

## Lesson 2 Exercises

2\_1. Write a JavaScript program that uses three variables and three prompt statements. Ask the user to enter their first name, middle name, and last name in separate prompt statements. Then use string concatenation to display the name in the following format:

last name, first name middle name

So if Thomas Francis Jones is the name that is entered, your program will display:  
Jones, Thomas Francis

Be careful of spaces. Be sure to include any needed spaces within your quote marks.

2\_2. Write a JavaScript program that asks the user to input their name, what city they were born in, and the month of their birthday. Then display that information using `document.write`.

2\_3. Make your own Mad Lib:

In this exercise you will create a Web page that serves as an interactive Mad Lib program. Your page will contain JavaScript code that prompts the user for words to fill in the blanks in a story, and then stores those words in variables. After having read in all of the words, your code should then display the story in the Web page, using the values of the variables where appropriate.

For example, here is a start to a Mad Lib:

It was a **adjective** kind of day when **person's name** walked out into the street. The sky was a deep **color**, and **same name** was walking his new pet **animal** ...

Making the following substitutions:

adjective = smarmy

person's name = Chris

color = mauve

animal = gnu

The story would read:

It was a **smarmy** kind of day when **Chris** walked out into the street. The sky was a deep **mauve**, and Chris was walking his new pet **gnu** ...

The content of the story can be anything that you like -- be creative! Your story must meet the

following conditions, however.

It must be at least two paragraphs long.

It must have at least six missing words.

At least one of the missing words must be used multiple times in the story. For example, the person's name was used twice in the sample story above.

The page should have a title, centered at the top, that includes your name.