



OBJECTIVES

- Reading and writing JavaScript.
- You will also learn how to give a web browser instructions you want it to follow.

STATEMENTS

- A script is a series of instructions that a computer can follow one-by-one.
- Each individual instruction or step is known as a **statement**.
- Statements should end with a semicolon.
- Examples:

```
var today= new Date();  
greeting = 'Welcome';  
document.write(greeting) ;
```
- Some **statements** are surrounded by **curly braces**;
- these are known as **code blocks**.
- The closing curly brace is not followed by a semicolon.

COMMENTS

- You should write **comments** to explain what your code does.
- They help make your code easier to read and understand.
- This can help you and others who read your code.

- **SINGLE-LINE COMMENTS**

```
var today = new Date(); // Create a new date object
var hour Now = today.getHours(); // Find the current hour
```

- **MULTI-LINE COMMENTS**

```
/* This script displays a greeting to the user based upon the current time.
It is an example from JavaScript & jQuery book */
```

VARIABLES

- **WHAT IS A VARIABLE?**

- A script will have to temporarily store the bits of information it needs to do its job. It can store this data in **variables**.
- How to declare a variable?

```
var quantity;
```

VARIABLE KEYWORD VARIABLE NAME

VARIABLES

- How to assign variable a value?

The diagram illustrates the components of an assignment statement. The word "quantity" is labeled as the "VARIABLE NAME" with a bracket underneath. The equals sign "=" is labeled as the "ASSIGNMENT OPERATOR" with a vertical line above it. The number "3" is labeled as the "VARIABLE VALUE" with a bracket underneath. The semicolon ";" is not explicitly labeled but is part of the statement.

DATA TYPES

- Numeric data type
- String data type
- Boolean data type

```
var price;  
var quantity;  
var total;  
  
price = 5;  
quantity = 14;  
total = price * quantity;  
  
var el = document.getElementById('cost');  
el.textContent = '$' + total;
```

VARIABLES

- Using a variable to store a string.

```
var username;  
var message;  
username = 'Molly';  
message = 'See our upcoming range';  
  
var elName = document.getElementById('name');  
elName.textContent = username;  
var elNote = document.getElementById('note');  
elNote.textContent = message;
```

VARIABLES

- Using quotes inside a string

```
var title;  
var message;  
title = "Molly's Special Offers";  
message = '<a href=\"sale.html\">25% off!</a>';  
  
var elTitle = document.getElementById('title');  
elTitle.innerHTML = title;  
var elNote = document.getElementById('note');  
elNote.innerHTML = message;
```

VARIABLES

- Using a variable to store a boolean

```
var inStock;
var shipping;
inStock = true;
shipping = false;

var elStock = document.getElementById('stock');
elStock.className = inStock;

var elShip = document.getElementById('shipping');
elShip.className = shipping;
```

VARIABLES

- Shorthand for creating variables

```
var price = 5;
var quantity = 14;
var total = price * quantity;
```

```
var price = 5, quantity = 14;
var total = price * quantity;
```

```
var price, quantity, total;
price = 5;
quantity = 14;
total = price * quantity;
```

```
var el = document.getElementById('cost');
el.textContent = '$' + total;
```

VARIABLES

- Rules for naming variables
- The name must begin with a letter, dollar sign (\$), or an underscore (_). It must not start with a number.
- The name can contain letters, numbers, dollar sign (\$), or an underscore (_). Note that you must not use a dash(-) or a period (.) in a variable name.
- You cannot use **keywords** or reserved words.
 - **Keywords** are special words that tell the interpreter to do something.

VARIABLES

- Rules for naming variables
- All variables are case sensitive, so **score** and **Score** would be different variable names, **but** it is bad practice to create two variables that have the same name using different cases.
- Use a name that describes the kind of information that the variable stores.
- If your variable name is made up of more than one word, use a capital letter for the first letter of every word *after* the first word. (**Camel Case**)

ARRAYS

- An array is a special type of variable. It doesn't just store one value; it stores a **list of values**.

```
var colors;  
colors = ['white', 'black', 'custom'];  
  
var el = document.getElementById('colors');  
el.textContent = colors[0];
```

EXPRESSIONS

- An **expression** evaluates into (results in) a single value. Broadly speaking there are two types of expressions.
 - Expressions that just assign a value to a variable

```
var color = 'beige';
```

- Expressions that use two or more values to return a single value

```
var area = 3 * 2;
```


OPERATORS

- Expressions rely on things called **operators**; they allow programmers to create a single value from one or more values.
- Assignment Operators
- Arithmetic Operators
- String Operators
- Comparison Operators
- Logical Operators

SUMMARY

- A script is made up of a series of statements. Each statement is like a step in a recipe.
- Scripts contain very precise instructions. For example, you might specify that a value must be remembered before creating a calculation using that value.
- Variables are used to temporarily store pieces of information used in the script.
- Arrays are special types of variables that store more than one piece of related information.
- JavaScript distinguishes between numbers (0-9), strings (text), and Boolean values (true or false).
- Expressions evaluate into a single value.
- Expressions rely on operators to calculate a value.

