

# JavaScript Introduction

## What is JavaScript?

- JavaScript was designed to **add interactivity to HTML pages**
- JavaScript is usually **embedded directly into HTML pages**
- JavaScript is an **interpreted** language (means that scripts execute without preliminary compilation)

## What can a JavaScript Do?

- **JavaScript gives HTML designers a programming tool** - JavaScript is a scripting language with a very simple syntax.
- **JavaScript can put dynamic text into an HTML page** – You can use JavaScript to add html code into your html page, e.g., `document.write("<h1>" + name + "</h1>")`
- **JavaScript can react to events** – You can make JavaScripts that execute when something happens, like when a page has loaded or when a user clicks on something.
- **JavaScript can read and write HTML elements** - A JavaScript can read and change the content of HTML.
- **JavaScript can be used to validate data** - A JavaScript can be used to validate form data before it is submitted to a server (e.g., make sure you've typed in a 7-digit . This saves the server from extra processing
- **JavaScript can be used to detect the visitor's browser** - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser
- **JavaScript can be used to create cookies** - A JavaScript can be used to store and retrieve information on the visitor's computer

## How to Put a JavaScript Into an HTML Page

```
<!DOCTYPE html>
<html>
  <head>
    <title>JavaScript Guidelines</title>
  </head>
  <body>

    <script>
      document.write("Hello World!");
    </script>

  </body>
</html>
```

**Note:** If we had not entered the `<script>` tag, the browser would have treated the `document.write("Hello World!")` command as pure text, and just written the entire line on the page.

### Example Explained

To insert a JavaScript into an HTML page, we use the **<script> tag**. So, the `<script >` and `</script>` tells the browser where the JavaScript starts and ends:

```
<html>
<body>
```

```
<script>
...
</script>
</body>
</html>
```

The word **document.write** is a standard JavaScript command for writing output to a page. **Document** is an object in JavaScript. There are a bunch of objects in JavaScript, like forms, windows, etc. Each object has functions (or actions) associated with it. The document object has a function called "write" that writes to the document.

By entering the document.write command between the <script> and </script> tags, the browser will recognize it as a JavaScript command and execute the code line. In this case the browser will write *Hello World!* to the page:

## JavaScript is Case Sensitive

JavaScript is case sensitive - therefore watch your capitalization closely when you write JavaScript statements, create or call variables, objects and functions.

- document.write() IS NOT the same thing as
- Document.write() IS NOT the same thing as
- document.Write() IS NOT the same thing as
- Document.Write()

## JavaScript Code

A JavaScript **statement is a command to the browser**. The purpose of the command is to **tell the browser what to do**.

This JavaScript statement tells the browser to write "Hello Dolly" to the web page:

```
document.write("Hello Dolly")
```

JavaScript code (or just JavaScript) **is a sequence of JavaScript statements**.

Each statement is **executed by the browser in the order** they are written.

### Example

```
<script >
document.write("<h1>This is a header</h1>");
document.write("<p>This is a paragraph</p>");
document.write("<p>This is another paragraph</p>");
</script>
```

## JavaScript Variables

Variables are "containers" for storing information.

A variable can have a short name, like x, or a more descriptive name, like carname.

Variables can be used to hold values (x=5 or carname = "volvo")

**THE VALUE ON THE RIGHT SIDE GOES INTO THE VARIABLE NAME ON THE LEFT!!**

Rules for JavaScript variable names:

- Variable names are **CASE SENSITIVE**

- (y and Y are two different variables)
- Variable names **MUST NOT** contain spaces
- Variable names must begin with a letter
- Variable names can only contain letters or numbers or \_
  - No special characters!!
- Variable names can be anything except words that already belong to JavaScript (e.g., document, var, write, etc.)
- Variable names are **CASE SENSITIVE**

## Example

```
<html>
  <head>
  </head>
<body>
  <script >

    var firstname;
    firstname="Fred";
    document.write(firstname + "<br />");

    firstname="Wilma";
    document.write(firstname + "<br />");

  </script>
</body>
</html>
```

## Creating JavaScript Variables

Create variables with the **var statement**:

```
var x;
var carname;
```

The variables are empty (they have no values yet).

You can assign values to the variables when you create them:

```
var x=5;
var carname="Volvo";
```

After the execution of the statements above, the variable **x** will hold the value **5**, and **carname** will hold the value **Volvo**.

**Note:** When you assign a text value to a variable, use quotes around the value.

You can always change the value inside a variable:

```
x = 7
carname = "Mazda"
```

Before, **x** held **5**, now it holds **7**. **carname** held **"Volvo"**, now the value has been changed to **"Mazda"**

You can change the values again if you like.

# Prompt Box

A prompt box is an easy way for you to ask user to enter info into your javascript.

When a prompt box pops up, the user types something into the box, then clicks either "OK" or "Cancel".

## Syntax:

```
x = prompt("sometext", "defaultvalue");
```

where:

- x is a variable,
- sometext is the question you want to ask, and
- defaultvalue is the default answer

The best way to understand a prompt box is to try it, so in your web page do the following:

## Example :

```
<html>
<head>
</head>
<body>
<script >
  var x = prompt("What state do you live in?", "Delaware")
  document.write("<p>You currently live in " + x + "</p>")
</script>
</body>
</html>
```

## The prompt box deconstructed:

```
var x = prompt("What state do you live in?", "Delaware")
```

Start on right side of =

- Prompt creates a popup box in which the user can type in text.
- Within the first set of quotes is what shows up as the prompt's text
  - E.g., "What state do you live in?"
- In the second set of quotes is the default answer
  - "Delaware"
  - You don't have to include a default answer

Left side of =:

- a variable (an empty box) named x
- what is typed in by the user goes into the variable on the left side
- If the user doesn't type in anything, the default value ("Delaware") will end up in the variable x.

***When the user types something into the prompt box, whatever they typed in is stored in the variable x***

**Example2:**

```
<html>
<head>
</head>
<body>
<script >
  var num1 = prompt("Enter a number between 1 and 6","1")

  document.write("<h"+num1+"> A lovely header of some sort </h"+num1+">")
</script>
</body>
</html>
```

In this case, the variable num1 holds the number the user typed in. We then put that number into the string being written into the web page.

So if the user typed in 3, num1 will hold the value 3, and the line:

```
document.write("<h"+num1+"> A lovely header of some sort </h"+num1+">")
```

becomes:

```
document.write("<h3> A lovely header of some sort </h3>")
```

**Example3:**

```
<html>
<head>
</head>
<body>
<script >
  var num1 = prompt("Enter a number between 1 and 12","1")

  var promptstring = "What is " + num1 + " times 3?"
  var num2 = prompt(promptstring)
  document.write("<p> You think " + num1 + " times 3 is " + num2 + "</p>")
</script>
</body>
</html>
```

In this case, num1 holds the number that the user types in. Say the user typed in the number 5. Then the variable num1 will hold the value 5.

The string, "What is " + num1 + "times 3?"

Becomes "What is 5 times 3?"

So the variable promptstring holds, "What is 5 times 3?"

Now the line,

```
var num2 = prompt(promptstring)
```

Becomes

```
var num2 = prompt("What is 5 times 3?")
```

And whatever the user types in is what the variable num2 will hold. Say the user typed in 17. Then the variable num2 will hold 17.

So the line,

```
document.write("<p> You think " + num1 + " times 3 is " + num2 + "</p>")
```

becomes:

```
document.write("<p> You think 5 times 3 is 17</p>")
```

## Basic Math Operators:

Now, what if you wanted to actually calculate the  $3 \times 5$ , or the number that the user typed in in response to the prompt? Javascript lets you do that using basic math operations.

The basic set of mathematical operations that javascript lets you do are:

*(assume var y = 5)*

Operator	Description	Example	Result
+	Addition	var x=y+2	x=7
-	Subtraction	var x=y-2	x=3
*	Multiplication	var x=y*2	x=10
/	Division	var x=y/2	x=2.5
%	Modulus (division remainder)	var x=y%2	x=1

The following script shows all of the above operators being used:

```
<script>
    var num1 = parseInt(prompt("Enter a number","1"));
    var num2 = parseInt(prompt("Enter a second number","1"));
    document.write("<h2> Fun with numbers </h2>");
    document.write("<p>");

    var z = num1 + num2;
    document.write(num1 + " + " + num2 + " = " + z + "<br >");

    z = num1 - num2;
    document.write(num1 + " - " + num2 + " = " + z + "<br >");

    z = num1 * num2;
    document.write(num1 + " * " + num2 + " = " + z + "<br >");

    z = num1 / num2;
    document.write(num1 + " / " + num2 + " = " + z + "<br >");

    z = num1 %num2;
    document.write(num1 + " % " + num2 + " = " + z + "<br >");
    document.write("</p >");
</script>
```

## If (conditional branching)

Now, what if we only want something to happen in our code conditionally? In real life, we've got conditions all the time: if it's raining, wear a raincoat. If there's a long line, I won't get coffee. If my bank account has more than 50 dollars in it, I'll get gas for my car.

We can do this with JavaScript as well. In javascript I can say:

```
if (bankaccount > 50)
{
    document.write("<p> Go ahead and get gas </p>");
}
```

A couple of things to note about the if condition. First, the if condition must have something it is evaluating to decide whether to do the code associated with the if condition. In other words, something must be true for us to do the code associated with the if. In our real life examples, the first condition was, "is it raining?" The answer is either yes, or no. If the answer is yes, then you should wear a raincoat. In the code above, the condition was, does the variable called bankaccount hold a value that is greater than 50? Again, the answer is either yes, or no, and if the answer is yes, we write the paragraph, "Go ahead and get gas". In javascript, that condition must be in parentheses.

Next, the code we want to happen if the condition is true must go inside of { and }. There can be more than one thing we want executed by the browser if something is true. It all must go inside the { and } associated with that if statement. So, in the above example I could have had:

```
if (bankaccount > 50)
{
    document.write("<p> You've got "+bankaccount+" in your bank account. </p>");
    document.write("<p> Go ahead and get gas </p>");
}
```

Here is an example of using the if condition with the prompt box:

```
<body>
  <script>
    var answer = prompt("How many pieces of chocolate do you want?");
    if (answer > 10)
    {
        document.write("<p> I see diabetes in your future! </p>");
    }
  </script>
</body>
```

In this case, the user is asked to type in a number in answer to the question, "how many pieces of chocolate do you want?" That number goes into the variable answer. If the variable answer then holds a number greater than 10 (and ONLY if the variable answer holds a number greater than 10), javascript

writes to the web document the paragraph, "I see diabetes in your future!". Otherwise (if the number in the variable answer is less than or equal to 10), nothing is written to the web document.

==

Here is another example of using the if condition:

```
<body>
  <script>
    var answer = prompt("Do you like puppies?");
    if (answer == "yes")
    {
        document.write("<p>Me too! I love puppies, especially puppy bellies!</p>");
    }
  </script>
</body>
```

Did you notice her the odd == in the if condition? In JavaScript (and all programming languages) we use one = for assignment, e.g.,

```
var x = 3;
```

assigns the value of 3 to the variable x.

But what if I wanted to check to see if x is 3 in my if condition? If I just said,

```
if (x = 3)
```

I just assigned the value of 3 to x (x now holds 3). I didn't check to see if x does in fact equal 3. So to check to see if x actually does equal 3, I'd use the double-equal, or ==:

```
if (x == 3)
```

Think of the == as a question: Does x == 3? Yes or no?

Or, if you prefer, just remember that in if conditions you ALWAYS use == to check to see if two things are the same.

## Other Comparators:

Other comparators you can use in the if condition are the following:

**(Assume**

```
var x=3;
```

```
var y = 7 );
```



Operator	English	Example	Result
==	Equal To	x == y	false
!=	Not Equal To	x != y	true
<	Less Than	x < y	true
>	Greater Than	x > y	false
<=	Less Than or Equal To	x <= y	True
>=	Greater Than or Equal To	x >= y	False

A quick note: for greater than or equal to, the > always comes before the =. Same with less than or equal to.

## Conditional if/else:

In real life, we've often got an "otherwise" option. For instance, if it's raining, bring your umbrella. Otherwise just grab your jacket. If you've got 50 dollars in your bank account, get gas in the car. Otherwise wait until payday. In JavaScript, we have the else option. So in JavaScript, you would have:

```

if (bankaccount > 50)
{
    document.write("<p> You've got enough money to get gas now!</p>");
}
else
{
    document.write("<p> Wait until payday.</p>");
}

```

The above script states that if the variable bankaccount holds a value greater than 50, the paragraph, "You've got enough money to get gas now!" is written out. Otherwise (and only otherwise!) the paragraph "Wait until payday" is written to the web page.

Here's another example:

```

<script>
    var age = prompt("What is your age?");
    if (age >= 21)
    {
        document.write("<p>Woo Woo. You're over 21 so you can drink legally.</p>");
    }
    else
    {
        document.write("<p>You're under 21 so don't even try drinking alcohol.</p>");
    }
</script>

```

In this case, the user types in a value in response to the prompt, “What is your age?”. The value goes into the variable age.

Then the script checks to see if the variable age is greater than or equal to 21. If the answer is yes, the paragraph, “Woo woo. You’re over 21 so you can drink legally.” is written to the web page. Otherwise (and ONLY if the value inside of the variable age is not greater than or equal to 21) the paragraph, “You’re under 21 so don’t even try drinking alcohol” is written out. With if branches we always only do the first branch that is true.

## if/else if/ else

You can have more than one option with an if condition. In real life, you might say, “If he’ll eat the broccoli, feed him the broccoli. Otherwise, if he’ll eat an apple, give him an apple. Otherwise just give him a cookie.” In this case, you do the first condition that is true. The child doesn’t get an apple if he’ll eat broccoli, and he doesn’t get a cookie if he’ll eat an apple. He only gets a cookie if he won’t eat broccoli or an apple.

Another example is if you’ve got over 50 in your bank account, you’ll fill your tank with gas. Otherwise, if you’ve got over 25 in your bank account, you’ll fill half a tank. Otherwise, you’ll just wait until payday to fill your tank.

With JavaScript, you’d write this as:

```
if (bankaccount > 50)
{
    document.write("<p> You’ve got enough money to get gas now!</p>");
}
else if (bankaccount > 25)
{
    document.write("<p> Fill only half of the tank for now..</p>");
}
else
{
    document.write("<p> Wait until payday.</p>");
}
```

Here we have a branching if condition. The browser will go through and check the first condition: is the value inside of bankaccount > 50? If so, the browser writes out the paragraph, “You’ve got enough money to get gas now!”. Only if your bankaccount doesn’t have over 50 in it does the browser check to see if the bankaccount amount is > 25. If this is true, then the browser writes out the paragraph, “Fill only half of the tank for now.” If, however, your bankaccount amount is not greater than 50 and it is not greater than 25, then and only then does the browser write out, “Wait until payday.”

Another example is:

```
<body>
  <h1> Calculate Your Grade </h1>
  <script>
    var tests = parseInt(prompt("Exam grade?"));
    var labs = parseInt(prompt("Lab grade?"));
    var total = tests + labs / 100;
    if (total >= 90)
    {      document.write("<p>You get an 'A' </p>");
    }
    else if (total >= 80)
    {      document.write("<p>You get a 'B' </p>");
    }
    else if (total >= 70)
    {      document.write("<p>You get a 'C' </p>");
    }
    else if (total >= 60)
    {      document.write("<p>You get a 'D' </p>");
    }
    else
    {      document.write("<p>You fail.</p>");
    }
    document.write("<p><em> Have a nice summer.</em></p>");
  </script>
</body>
```

In this case, you are prompted to enter your exam grade, and that grade is converted to an integer ( a number) and the number goes into tests. You're then asked to enter your lab grade, and when you type that in, it is converted to a number and that number goes into the labs variable. Then your total grade is calculated and stored in the variable total.

Now, if total holds a value greater than 90, the browser writes out the paragraph, "You get an 'A'". The if, with all its branches, is done, and we drop down outside all of the if conditions and write out the paragraph, "Have a nice summer!"

If, however the total does not hold a value greater than 90, then we check to see if the total is greater than 80. If it is, we write out, "You get a 'B'" and drop down past all the if conditions to print out, "Have a nice summer."

If, however, the total is not greater than 90, and not greater than 80, then and only then do we check to see if it's greater than 70. If it is, we write out, "You get a 'C'" and then write out, "Have a nice summer."

If total is not greater than 90, and it is not greater than 80, and it is not greater than 70, then we check to see if it is greater than 60. If it is we write out, "You get a 'D'" and then write "Have a nice summer."

Finally, if the total is not greater than 90, and it is not greater than 80, and it is not greater than 70, and it is not greater than 60, then (and only then) do we write you, "You fail." And then write out, "Have a nice summer."

### **Side note: parseInt**

*parseInt makes whatever you typed in be an integer. It forces what you've typed in to be a number. In computer science there is a difference between numbers and words. So 24 and "24" are two different things. If I want to make sure that a number is, in fact, a number, I can use parseInt. So, for example, in the following line:*

```
var tests = parseInt(prompt("Exam grade?"));
```

Is exactly the same as:

```
var tests = prompt("Exam grade?");
```

Only by adding parseInt, I've made sure that what the user types in gets converted into a number. So *when we prompt the user to enter a number, we should always use parseInt around it to turn it into a number.*

### **Random numbers**

Often in programming, we want to generate a random number. Games often use random numbers to represent the rolling of dice, for instance, or to represent a random card in a deck of cards, or a random location on a board. In tests, we may use a random number to randomly choose the question that comes next.

JavaScript has a built-in method for randomly generating a number. It is the following:

```
var randnum = Math.floor(Math.random() * range);
```

What this does: It generates a random number between 0 and *range* and puts that random number in the variable *randnum*

To break it down,

```
var randnum = Math.floor(Math.random() * 9)
```

`Math.random()` generates a random number between 0 and 1 (e.g., 0.4).

This number is multiplied by the range:

```
Math.random() * 9
```

gives us 3.6

```
Math.floor(3.6)
```

rounds down to the nearest integer. or 3

Finally, you need a variable to put all this in. I called my variable randnum, so in essence,

```
var randnum = 3
```

Here's an example:

```
<script>
  var guess = Math.floor(Math.random() * 11);
  var person = parseInt(prompt("Enter a number between 0 and 10));
  if (guess == person)
  {      document.write("<p>You guessed right!</p>");
  }
  else
  {      document.write("<p>You're wrong</p>");
  }
</script>
```

In this script, the computer generates a random number between 0 and 10 and that random number goes into the variable guess. Then the user is asked to enter a number between 0 and 10, that number is converted into a number (by parseInt), and then goes into the variable person. If the value inside the variable guess is the same as the value inside the variable person, the browser writes, "You guessed right!". Otherwise it writes out the paragraph, "You're wrong".

Here are some other examples of using random numbers:

```
<body>
  <h1> Magic 8 Ball</h1>
  <p> Ask your question and the magic 8 ball will predict your answer! </p>
  <script>
    var ans = prompt("What do you wish to ask the magic 8 ball?");
    var randnum = Math.floor(Math.random() * 8);
    if (randnum == 0)
    {      document.write("<p> There is no chance this will happen.</p>")
    }
    else if (randnum == 1)
    {      document.write("<p> It is highly unlikely! </p>");
    }
    else if (randnum == 2)
    {      document.write("<p> Probably not, but you never know.</p>");
    }
    else if (randnum == 3)
    {      document.write("<p> Maybe. Maybe not.</p>");
    }
    else if (randnum == 4)
    {      document.write("<p> This may happen if you are lucky.</p>");
    }
  </script>
```

```

else if (randnum == 5)
{
    document.write("<p> Your chances are good.</p>");
}
else if (randnum == 6)
{
    document.write("<p> This will almost certainly happen.</p>");
}
else if (randnum == 7)
{
    document.write("<p> Definitely.</p>");
}
document.write("<p><em> Thank you for consulting the magic 8 ball!</em></p>");
</script>

```

Here's another script:

```

<body>
<h1> Heads or Tails? </h1>
<script>
    var ans = prompt("Heads or Tails?");
    if (ans == "Heads")
    {
        var numans = 0;
    }
    else if (ans == "Tails")
    {
        var numans = 1;
    }
    var compguess = Math.floor(Math.random() * 2);
    if (compguess == 0)
    {
        var compan = "Heads";
    }
    else
    {
        var compan = "Tails";
    }
    if (numans == compguess)
    {
        document.write("<p> You guessed correctly! You both guessed "+ans+"!</p>")
    }
    else
    {
        document.write("<p> You guessed incorrectly. <br>")
        document.write("You guessed "+ans+" and the computer guessed "+compan+".</p>")
    }
    document.write("<p><em> Play again sometime!.</em></p>")
</script>
</body>

```

And finally:

```
<body>
  <h1> Page Colors:</h1>
  <script>
    var randcolor = Math.floor(Math.random() * 3);
    if (randcolor == 0)
    {      document.write("<p style = \"color: #0000FF\"> This is blue </p>");
    }
    else if (randcolor == 1)
    {      document.write("<p style = \"color: #FF0000\"> This is red </p>");
    }
    else if (randcolor == 2)
    {      document.write("<p style = \"color: #00FF00\"> This is green </p>");
    }
  </script>
</body>
```