

Week 2

2.1 Variables

1. Declaring Variables

- Creating a variable in JavaScript is called "declaring" a variable.

a. You declare a JavaScript variable with the `var` keyword:

```
var carName;
```

b. After the declaration, the variable has no value (technically it has the value of undefined). To assign a value to the variable, use the equal sign:

```
var carName = "Volvo";
```

c. You can also assign a value to the variable when you declare it:

Example:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Variables</h2>

<p>Create a variable, assign a value to it,
and display it: </p>

<p id="demo"></p>

<script>
var carName = "Volvo";
document.getElementById("demo").innerHTML =
carName;
</script>

</body>
</html>
```

JavaScript Variables

Create a variable, assign a value to it, and display it:

Volvo

2. JavaScript Identifiers

- All JavaScript **variables** must be **identified** with **unique names**.
- These unique names are called **identifiers**.
- Identifiers can be short names (like x and y) or
- more descriptive names (age, sum, totalVolume).
- JavaScript identifiers are case-sensitive.

The general rules for constructing names for variables (unique identifiers) are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter
- Names can also begin with \$ and _ (but we will not use it in this tutorial)
- Names are case sensitive (y and Y are different variables)
- Reserved words (like JavaScript keywords) cannot be used as names

3. JavaScript Data Types

Note that variables may hold values that have different data types:

String This is a sequence of text known as a string. To signify that the value is a string, enclose it in single quote marks.

```
Let myVariable = 'Bob';
```

Number This is a number. Numbers don't have quotes around them.

```
Let myVariable = 10;
```

Boolean This is a True/False value. The words true and false are special keywords that don't need quote marks.

```
let myVariable = true;
```

Array This is a structure that allows you to store multiple values in a single reference.

```
let myVariable = [1, 'Bob', 'Steve', 10];
```

Object This can be anything. Everything in JavaScript is an object and can be stored in a variable. Keep this in mind as you learn.

```
let myVariable = document.querySelector('h1');
```

2.2 Operators

1. JavaScript Arithmetic Operators

Arithmetic operators perform arithmetic on numbers (literals or variables).

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

The two numbers can be literals:

```
<!DOCTYPE html>
<html>
<body>

<p>A typical arithmetic operation takes two
numbers and produces a new number.</p>

<p id="demo"></p>

<script>
var x = 100 + 50;
document.getElementById( "demo" ).innerHTML =
x;
</script>

</body>
</html>
```

A typical arithmetic operation
takes two numbers and
produces a new number.

150

The two numbers can be variables:

```
<!DOCTYPE html>
<html>
<body>

<p>A typical arithmetic operation takes two
numbers (or variables) and produces a new
number.</p>

<p id="demo"></p>

<script>
var a = 100;
var b = 100;
var x = a + b;

document.getElementById( "demo" ).innerHTML =
x;
</script>

</body>
</html>
```

A typical arithmetic operation
takes two numbers and
produces a new number.

150

2. Operators and Operands

- The numbers (in an arithmetic operation) are called operands.
- The operation (to be performed between the two operands) is defined by an operator.

Operand	Operator	Operand
100	+	50

Adding

The addition operator (+) adds numbers:

```
<!DOCTYPE html>
<html>
<body>

<h2>The + Operator</h2>

<p id="demo"></p>

<script>
var x = 5;
var y = 2;
var z = x + y;
document.getElementById( "demo" ).innerHTML =
z;
</script>

</body>
</html>
```

The + Operator

7

Subtracting

The subtraction operator (-) subtracts numbers.

```
<!DOCTYPE html>
<html>
<body>

<h2>The - Operator</h2>

<p id="demo"></p>

<script>
var x = 5;
var y = 2;
var z = x - y;
document.getElementById( "demo" ).innerHTML =
z;
</script>

</body>
</html>
```

The - Operator

3

Multiplying

The multiplication operator (`*`) multiplies numbers.

```
<!DOCTYPE html>
<html>
<body>

<h2>The * Operator</h2>

<p id="demo"></p>

<script>
var x = 5;
var y = 2;
var z = x * y;
document.getElementById( "demo" ).innerHTML =
z;
</script>

</body>
</html>
```

The * Operator

10

Dividing

The division operator (`/`) divides numbers.

```
<!DOCTYPE html>
<html>
<body>

<h2>The / Operator</h2>

<p id="demo"></p>

<script>
var x = 5;
var y = 2;
var z = x / y;
document.getElementById( "demo" ).innerHTML =
z;
</script>

</body>
</html>
```

The / Operator

2.5

Remainder

The modulus operator (`%`) returns the division remainder.

```
<!DOCTYPE html>
<html>
<body>

<h2>The % Operator</h2>

<p id="demo"></p>

<script>
var x = 5;
var y = 2;
var z = x % y;
document.getElementById( "demo" ).innerHTML =
z;
</script>

</body>
</html>
```

The % Operator

2

Incrementing The increment operator (**++**) increments numbers.

```
<!DOCTYPE html>
<html>
<body>

<h2>The ++ Operator</h2>

<p id="demo"></p>

<script>
var x = 5;
x++ ;
var z = x;
document.getElementById( "demo" ).innerHTML =
z;
</script>

</body>
</html>
```

The ++ Operator

6

Decrementing The decrement operator (**--**) decrements numbers.

```
<!DOCTYPE html>
<html>
<body>

<h2>The -- Operator</h2>

<p id="demo"></p>

<script>
var x = 5;
x-- ;
var z = x;
document.getElementById( "demo" ).innerHTML =
z;
</script>

</body>
</html>
```

The -- Operator

4

Exponentiation The exponentiation operator (******) raises the first operand to the power of the second operand.

```
<!DOCTYPE html>
<html>
<body>

<h2>The ** Operator</h2>

<p id="demo"></p>

<script>
var x = 5;
document.getElementById( "demo" ).innerHTML =
x ** 2;
</script>

</body>
</html>
```

The ** Operator

25

3. Comparison Operators

Assignment operators assign values to JavaScript variables.

Operator	Description
<code>==</code>	equal to
<code>!=</code>	not equal
<code>></code>	greater than
<code><</code>	less than
<code>>=</code>	greater than or equal to
<code><=</code>	less than or equal to

`==` equal to

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Comparison</h2>

<p>Assign 5 to x, and display the value of
the comparison (x == 5):</p>

<p id="demo"></p>

<script>
var x = 5;
document.getElementById("demo").innerHTML =
(x == 5);
</script>

</body>
</html>
```

JavaScript Comparison

Assign 5 to x, and display the value of the comparison (x == 5):

true

! = not equal

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Comparison</h2>

<p>Assign 5 to x and display the value of
the comparison (x != 8).</p>

<p id="demo"></p>

<script>
var x = 5;
document.getElementById("demo").innerHTML =
(x != 8);
</script>

</body>
</html>
```

JavaScript Comparison

Assign 5 to x and display the value of the comparison (x != 8):

true

> greater than

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Comparison</h2>

<p>Assign 5 to x and display the value of
the comparison (x > 8).</p>

<p id="demo"></p>

<script>
var x = 5;
document.getElementById("demo").innerHTML =
(x > 8);
</script>

</body>
</html>
```

JavaScript Comparison

Assign 5 to x and display the value of the comparison (x > 8):

true

< =

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Comparison</h2>

<p>Assign 5 to x and display the value of
the comparison (x <= 8).</p>

<p id="demo"></p>

<script>
var x = 5;
document.getElementById("demo").innerHTML =
(x <= 8);
</script>

</body>
</html>
```

JavaScript Comparison

Assign 5 to x and display the value of the comparison (x <= 8):

true

4. Logical Operators

Operator	Description
& &	logical and
	logical or
!	logical not

Logical And

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Comparison</h2>

<p>The AND operator (&&) returns true if both
expressions are true, otherwise it returns
false.</p>

<p id="demo"></p>

<script>
var x = 5;
document.getElementById("demo").innerHTML =
(x < 10 && y > 1) + "<br>" +
(x < 10 && y < 1);
</script>

</body>
</html>
```

JavaScript Comparison

The AND operator (&&) returns true if both expressions are true, otherwise it returns false.

true
false

Logical Or

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Comparison</h2>

<p>The OR operator (||) returns true if one or both
expressions are true, otherwise it returns false.</p>

<p id="demo"></p>

<script>
var x = 6;
var y = 3;

document.getElementById("demo").innerHTML =
(x == 5 || y == 5) + "<br>" +
(x == 6 || y == 0) + "<br>" +
(x == 0 || y == 3) + "<br>" +
(x == 6 || y == 3);
</script>

</body>
</html>
```

JavaScript Comparison

The OR operator (||) returns true if one or both expressions are true, otherwise it returns false.

false
true
true
true

Logical Not

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Comparison</h2>

<p>The NOT operator (!) returns true for false
statements and false for true statements.</p>

<p id="demo"></p>

<script>
var x = 6;
var y = 3;

document.getElementById("demo").innerHTML =
!(x === y) + "<br>" +
!(x > y);
</script>

</body>
</html>
```

JavaScript Comparison

The NOT operator (!) returns true for false statements and false for true statements.

true
false

Sources:

JavaScript Arithmetic Operators. (n.d.). Retrieved February 3, 2021 from https://www.w3schools.com/js/js_arithmetic.asp

JavaScript Comparison and Logical Operators. (n.d.). Retrieved February 3, 2021 from https://www.w3schools.com/js/js_comparisons.asp

JavaScript Operators. (n.d.). w3schools. Retrieved February 3, 2021 from https://www.w3schools.com/js/js_operators.asp

JavaScript Variables. (n.d.). w3schools. Retrieved February 3, 2021 from https://www.w3schools.com/js/js_variables.asp