

In computer science, data is anything that is meaningful to the computer. JavaScript provides eight different data types which are `undefined, null, boolean, string, symbol, bigint, number, and object`.

For example, computers distinguish between numbers, such as the number 12, and strings, such as "12", "dog", or "123 cats", which are collections of characters. Computers can perform mathematical operations on a number, but not on a string.

```
var ourName;
```

Variables allow computers to store and manipulate data in a dynamic fashion. They do this by using a "label" to point to the data rather than using the data itself. Any of the eight data types may be stored in a variable.

creates a variable called `ourName`. In JavaScript we end statements with semicolons. Variable names can be made up of `numbers, letters, and $ or _` but may not contain spaces or start with a number.

Use the `var` keyword to create a variable called `myName`.

Storing Values with the Assignment Operator

In JavaScript, you can store a value in a variable with the assignment operator (=).

```
myVariable = 5;
```

This assigns the **Number** value **5** to **myVariable**.

If there are any calculations to the right of the = operator, those are performed before the value is assigned to the variable on the left of the operator.

```
var myVar;  
myVar = 5;
```

First, this code creates a variable named myVar. Then, the code assigns 5 to myVar. Now, if myVar appears again in the code, the program will treat it as if it is 5.

Assign the value 7 to variable a.

Assigning the Value of One Variable to Another

After a value is assigned to a variable using the assignment operator, you can assign the value of that variable to another variable using the **assignment** operator.

```
var myVar;  
myVar = 5;  
var myNum;  
myNum = myVar;
```

```
// Setup  
var a;  
a = 7;
```

```
// Only change code below this line  
b=a;
```

The above declares a `myVar` variable with no value, then assigns it the value 5. Next, a variable named `myNum` is declared with no value. Then, the contents of `myVar` (which is 5) is assigned to the variable `myNum`. Now, `myNum` also has the value of 5.

Assign the contents of `a` to variable `b`.

```
// Setup  
var a;  
a = 7;  
var b;  
  
// Only change code below this line  
b = a;
```

Initializing Variables with the Assignment Operator

It is common to initialize a variable to an initial value in the same line as it is declared.

```
var myVar = 0;
```

Creates a new variable called `myVar` and assigns it an initial value of 0.

Define a variable `a` with `var` and initialize it to a value of 9.

```
var a = 9 ;
```

Declare String Variables

Previously you used the following code to declare a variable:

```
var myName;
```

But you can also declare a string variable like this:

"your name" is called a string literal. A string literal, or string, is a series of zero or more characters enclosed in single or double quotes.

Create two new string variables: `myFirstName` and `myLastName` and assign them the values of your first and last name, respectively.

```
// Example
var firstName = "Alan";
var lastName = "Turing";

// Only change code below this line

var myFirstName = "Beau";
var myLastName = "Carnes";
```