**Differences between Node.js and the Browser**

Both the browser and Node.js use JavaScript as their programming language. Building apps that run in the browser is a completely different thing than building a Node.js application. Despite the fact that it's always JavaScript, there are some key differences that make the experience radically different.

From the perspective of a frontend developer who extensively uses JavaScript, Node.js apps bring with them a huge advantage: the comfort of programming everything - the frontend and the backend - in a single language.

You have a huge opportunity because we know how hard it is to fully, deeply learn a programming language, and by using the same language to perform all your work on the web - both on the client and on the server, you're in a unique position of advantage.

## Data type examples

Data types can vary based on size, length and use depending on the coding language. Here are some examples of the data types listed above that you might encounter when programming:

### Integer

Integers are digits that account for whole numbers only. Some integer examples include:

* *425*
* *65*
* *9*

### Character

Characters are letters or other figures that programmers might combine in a string. Examples of characters include:

* *a*
* ^
* *!*

### Date

Programmers can include individual dates, ranges or differences in their code. Some examples might be:

* *2009-09-15*
* *1998-11-30 09:45:87*
* *SYSDATETIME ()*

### Long

Long data types are whole numbers, both positive and negative, that have many place values. Examples include:

* *-398,741,129,664,271*
* *9,000,000,125,356,546*

### Short

Short data types can be up to several integers, but they are always less than long data. Examples include:

* *-27,400*
* *5,428*
* *17*

### Floating point (real)

Float data types might look like this:

* *float num1 = 1.45E2*
* *float num2 = 9.34567*

Similar but often longer in length, an example of the floating-point double might be:

* *double num2 = 1.87358497267482791E+222*
* *double num2 = 3.198728764857268945*

The floating-point double type can provide more accurate values, but it also may require additional memory to process.

### String

### Strings are a combination of figures that includes letters and punctuation. In some code, this might look like this:

* *String a = new String("Open")*
* *String b = new String("The door")*
* *String c = new String("Say Hello!")*

These can be independent commands, or they can work together.

### Boolean

Boolean data can help guide the logic in a code. Here are some examples of how you might use this:

* *bool baseballIsBest = false;*
* *bool footballIsBest = true;*

Depending on the program, the code may direct the end-user to different screens based on their selection.

**Related:** [**How To Use Boolean Search Strings To Find Great Resumes**](https://www.indeed.com/career-advice/career-development/boolean-search-strings)

### Nothing

Nothing means a code has no value, but the programmer coded something other than the digit 0. This is often "Null," "NaN" or "Nothing" in code. An example of this is:

* *Dim option = Nothing*
* *Program.WriteWords(x Is Nothing)*

### Void

The void data type in coding functions as an indicator that code might not have a function or a response yet. This might appear as:

* *int function\_name (void)*

*Please note that none of the products mentioned in this article are affiliated with Indeed.*