Introduction To JavaScript

•••

Alex Fernandez
Senior Front-End Developer

What is JavaScript?

JavaScript is a cross-platform, object-oriented scripting language. It is small and lightweight. Inside a host environment (like a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements.

Data Types

Let's start off by looking at the building block of any language: the types. JavaScript programs manipulate values, and those values all belong to a type. JavaScript's types are:

- **Number**
- **String**
- **Boolean**
- **Function**
- **Object (Function, Array, Date, RegExp)**
- **null**
- **undefined**
- **Symbol (ES6)**

Numbers

Numbers are quantities, just like you're used to. You can do math with them.

- 4 + 4 = 8
- “4” + 4 = “44” // If you add a string to a number everything is converted into a string first.
- parseInt("4") + 4 = 8
- parseInt("SomeText4") + 4 = NaN

Strings

Strings are sequences of characters, like the letters a-z, spaces, and even numbers.

- “Red Ventures”
- “4”
- “What is your name?”

Boolean

A boolean is either true or false (both of which are keywords). Any value can be converted to a boolean according to the following rules:

1. false, 0, the empty string (‘’), NaN, null, and undefined all become false
2. all other values become true

Examples:

- Boolean(‘’); // false
- Boolean(0); // false
- Boolean(234); // true
- Boolean(10 > 9); // true

Variables

New variables in JavaScript are declared using the **var** keyword:

1. var a;
2. var name = ‘Red Ventures’;

If you declare a variable without assigning any value to it, its type is undefined.

An important difference from other languages like Java is that in JavaScript, blocks do not have scope; only functions have scope. So if a variable is defined using var in a compound statement (for example inside an if control structure), it will be visible to the entire function. However, starting with ES6, let and const declarations allow you to create block-scoped variables.

Operators

JavaScript's numeric operators are +, -, *, / and % - which is the remainder operator. Values are assigned using =, and there are also compound assignment statements such as += and -=.

The following are the same:

1. x += 5;
2. x = x + 5;

Comparisons

- > Greater than
- < Less than
- <= Less than or equal to
- >= Greater than or equal to
- === Equal to
- !== Not equal to

Examples:

1. “dog” === “dog”; // true
2. 1 === true; // true
3. 0 === true; // false

if {} else {}

An if statement is made up of the if keyword, a condition like we've seen before, and a pair of curly braces { }. If the answer to the condition is true, the code inside the curly braces will run. Otherwise, the condition is false, so only the code inside the second pair of curly braces after the else keyword will run.

```
var name = prompt('What is your name?');

if ( name.length >= 7 ) {
    console.log(‘You have a long name!’);
} else {
    console.log(‘You have a short name!’);
}
```

Functions

A function takes in inputs, does something with them, and produces an output.

Why use functions? D.R.Y - Don’t Repeat Yourself.

```
function multiply(x, y) {
	return x * y;
}

multiply(2,5); // 10

function volume(w, l, h) {
	return w * l * h;
}

volume(2, 3, 4); // 24
```

JavaScript Libraries/Frameworks

- jQuery
- MooTools
- Prototype
- Angular
- Underscore
- Mustache
- Etc

The quality of your code
will never be better than
your understanding of
the language.

Questions?

Contact me *anytime* via

TWITTER
@alexfislegend

EMAIL
afernandez@redventures.com