

# Date and Time Formatting

Take 2, 4/29/99

Richard Gillam

Currently, `Date` has two functions for converting the value into a string for display: `Date.prototype.toString()` and `Date.prototype.toLocaleString()`. Both are implementation dependent, and are intended to produce a string that “represent[s] the Date in a convenient, human-readable form.” There is one big problem, though: a `Date` can have both date and time components. It doesn’t say, but it would seem that both of these functions would display both the date and the time components.

Currently, the only way to display just a date or just a time is to isolate the individual fields using `getHours()`, `getMinutes()`, etc. and format them independently as numbers. The problem is that this doesn’t produce a localized solution, as `toLocaleString()` does.

For consistency, the working group’s consensus is that we should also have `toDateString()` and `toTimeString()` methods.

Therefore, I propose adding four new functions:

`Date.prototype.toLocaleDateString()`

Does exactly the same thing as `toLocaleString()`, but the resulting string only represents the “date” part of the date. (For example, if the `Date` value represented March 10, 1967, 11:15:23 AM, the output would be something along the lines of “March 10, 1967”.)

`Date.prototype.toLocaleTimeString()`

Does exactly the same thing as `toLocaleString()`, but the resulting string only represents the “time” part of the date. (For example, if the `Date` value represented March 10, 1967, 11:15:23 AM, the output would be something along the lines of “11:15:23 AM”.)

`Date.prototype.toString()`

Does exactly the same thing as `toString()`, but the resulting string only represents the “date” part of the date. (For example, if the `Date` value represented March 10, 1967, 11:15:23 AM, the output would be something along the lines of “March 10, 1967”.) The difference between this function and `toLocaleDateString()` is that this function’s result is locale-independent.

`Date.prototype.toTimeString()`

Does exactly the same thing as `toString()`, but the resulting string only represents the “time” part of the date. (For example, if the `Date` value represented March 10, 1967, 11:15:23 AM, the output would be something along the lines of

“11:15:23 AM”.) The difference between this function and `toLocaleTimeString()` is that this function’s result is locale-independent.