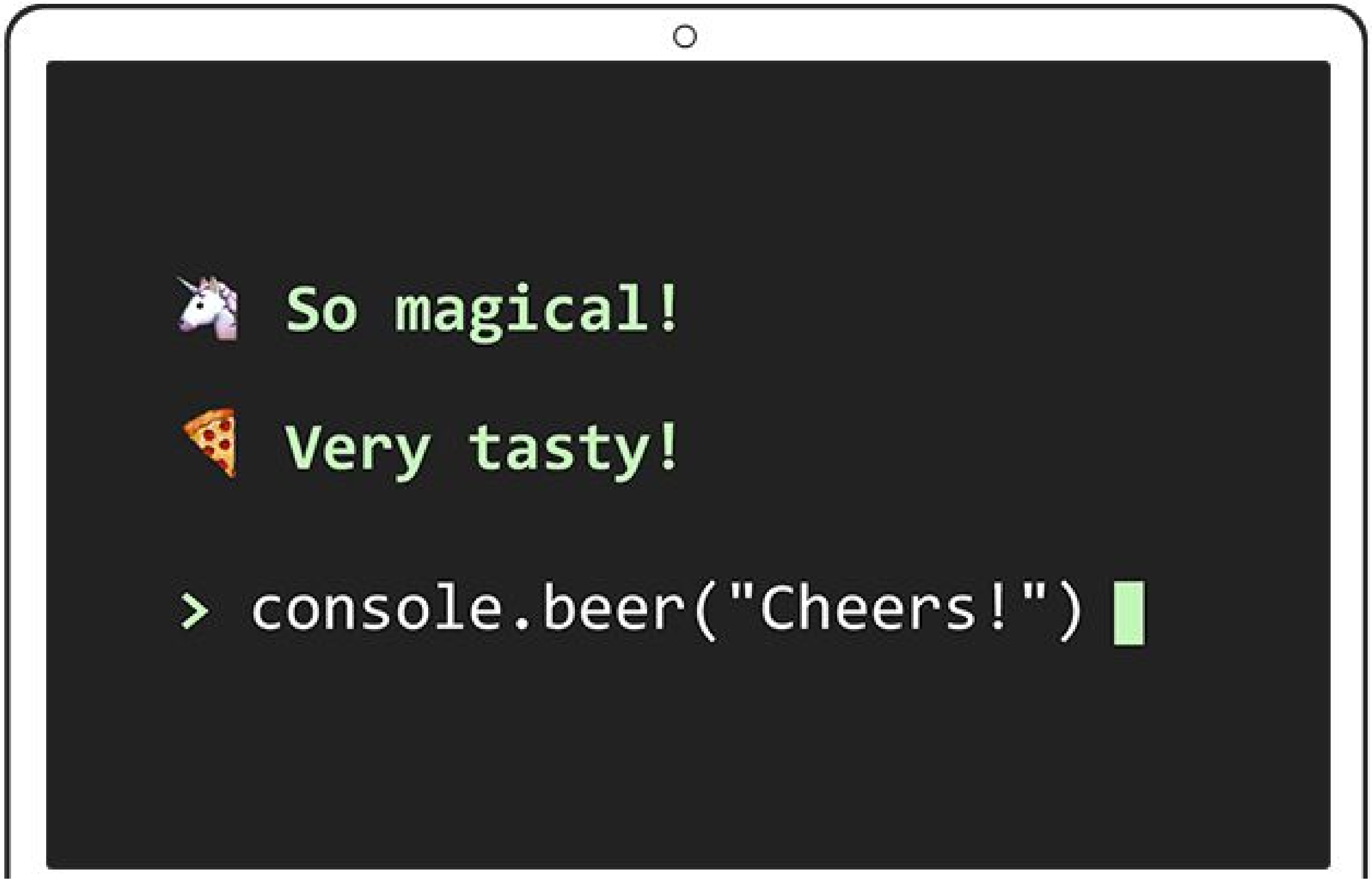
CUSTOM CONSOLE LOGGING WITH EMOJI

CODE · JUNE 20, 2016 ·

Debugging JavaScript can be painful and often means spending hours looking at boring console logs. Inspired by console.frog by Tim Holman, a little script that displays your messages with an ASCII frog, I made my own console logger — but with customizable commands and emoji. Logging to the console is finally fun again, thanks to `console.beer` 🍺 and `console.unicorn` 🦄.

Go here to head straight to the code on GitHub.



```
🦄 So magical!
🍕 Very tasty!
> console.beer("Cheers!")
```

Better Programming with Emoji

Emoji are not just silly icons to add to tweets and text messages — they can actually help you write better code and make debugging more fun:

> *We, as developers, routinely look at large amounts of text—whether it's code, production logs, commit messages, documentation, or whatever—and emoji inherently stand out in what is normally a wall of text. It's far easier to pick an emoji out of a list than a random string, and that skimmability can lead to real productivity gains.*
>
> **How Emoji Can Improve Your Code—Seriously**

You can add them as part of a comment to make it easier to find todos, or mark sections that require testing or refactoring. Or you can add them to your commit messages on GitHub to tag different types of updates or mark particularly important ones.

The Script

The idea behind this is pretty simple: all we have to do is add custom functions for each command to the

The idea behind this is pretty simple: all we have to do is add custom functions for each command to the `window.console` object. Each function logs the message and emoji to the console. To see it in action, check out the JavaScript console for this page.

```
// Define your custom commands and emoji
var commands = [
  [ "unicorn", "🦄" ],
  [ "pizza", "🍕" ],
  [ "beer", "🍺"],
  [ "poo", "💩"]
];

(function() {
  if(!window.console) return;

  // Create custom commands
  commands.forEach(function(command) {
    window.console[command[0]] = function() {

      // Get arguments as a string
      var args = Array.prototype.slice.call(arguments).toString().split(',').join(', ');

      // Log to the console with emoji
      console.log(command[1] + "  " + args);
    }
  });
})();

// Log to the console!
console.unicorn("Magical!");
console.beer("Cheers!");
console.pizza("Tasty!");
console.poo("Oh f*ck!");
```

Update: The ES6 version (2016-09-17)

Thanks to ECMAScript 6, `console.emoji` has become pretty much a one-liner. Instead of an awkward two-item array to specify the commands, I'm now using an array of objects with `emoji` and `name` keys:

```
// Define your custom commands and emoji
const commands = [
    { emoji: '🦄', name: 'unicorn' },
    { emoji: '🍕', name: 'pizza' },
    { emoji: '🍺', name: 'beer' },
    { emoji: '💩', name: 'poo' }
];

// Create custom commands
commands.forEach(({ name, emoji }) => window.console[name] = (...args) => console.log(emoji + ' ' + args.join(', ')));
```

View Code on GitHub

ABOUT THE AUTHOR

Ines Montani

I'm a digital native, programmer and front-end developer working on Artificial Intelligence and Natural Language Processing technologies. I'm the co-founder of Explosion AI and a core developer of spaCy and Prodigy.

TALKPYTHON: BUILDING A SOFTWARE BUSINESS

INTERVIEW · MARCH 9, 2019 ·

INCREMENT MAGAZINE: TRANSFORMING

SPACY'S DOCS

TECH · AUGUST 10, 2018 ·

EUROPYTHON 2018 KEYNOTE

HOW TO IGNORE MOST STARTUP ADVICE AND BUILD A DECENT SOFTWARE BUSINESS

VIDEO · JULY 26, 2018 ·

#NEWWWYEAR

CODE · DECEMBER 26, 2017