mattboldt / **typed.js** Public

A JavaScript Typing Animation Library

www.mattboldt.com/demos/typed-js/

MIT License

11.8k stars 1.5k forks

Star Watch

Code Issues 1 Pull requests 2 Actions Projects 1 Wiki Security Insights

master ...

mattboldt ... on 23 Dec 2021

View code

build passing | maintainability C | release v2.0.12 | downloads 1.8M | license MIT

Typed.js|

Live Demo | View All Demos | View Full Docs | mattboldt.com

Typed.js is a library that types. Enter in any string, and watch it type at the speed you've set, backspace what it's typed, and begin a new sentence for however many strings you've set.

Installation

Choose One

```
npm install typed.js
yarn add typed.js
bower install typed.js
```

CDN

```
<script src="https://cdn.jsdelivr.net/npm/typed.js@2.0.12"></script>
```

Setup

This is really all you need to get going.

```
// Can also be included with a regular script tag
import Typed from 'typed.js';

var options = {
  strings: ['<i>First</i> sentence.', '&amp; a second sentence.'],
  typeSpeed: 40
};

var typed = new Typed('.element', options);
```

Use with ReactJS

Hook-based function component: https://jsfiddle.net/mattboldt/60h9an7y/

Class component: https://jsfiddle.net/mattboldt/ovat9jmp/

Use with Vue.js

Check out the Vue.js component: https://github.com/Orlandster/vue-typed-js

Use it as WebComponent

Check out the WebComponent: https://github.com/Orlandster/wc-typed-js

Wonderful sites that have used (or are using) Typed.js

https://github.com/features/package-registry

https://slack.com/

https://envato.com/

https://gorails.com/

https://productmap.co/

https://www.typed.com/

https://apeiron.io

https://git.market/

https://commando.io/

http://testdouble.com/agency.html

https://www.capitalfactory.com/

http://www.maxcdn.com/

https://www.powerauth.com/

Strings from static HTML (SEO Friendly)

Rather than using the `strings` array to insert strings, you can place an HTML `div` on the page and read from it. This allows bots and search engines, as well as users with JavaScript disabled, to see your text on the page.

```
<script>
  var typed = new Typed('#typed', {
    stringsElement: '#typed-strings'
  });
</script>
```

```
<div id="typed-strings">
  <p>Typed.js is a <strong>JavaScript</strong> library.</p>
  <p>It <em>types</em> out sentences.</p>
</div>
<span id="typed"></span>
```

Type Pausing

You can pause in the middle of a string for a given amount of time by including an escape character.

```
var typed = new Typed('.element', {
  // Waits 1000ms after typing "First"
  strings: ['First ^1000 sentence.', 'Second sentence.']
});
```

Smart Backspacing

In the following example, this would only backspace the words after "This is a"

```
var typed = new Typed('.element', {
  strings: ['This is a JavaScript library', 'This is an ES6 module'],
  smartBackspace: true // Default value
});
```

Bulk Typing

The following example would emulate how a terminal acts when typing a command and seeing its result.

```
var typed = new Typed('.element', {
  strings: ['git push --force ^1000\n `pushed to origin with option force`']
});
```

CSS

CSS animations are built upon initialization in JavaScript. But, you can customize them at your will! These classes are:

```
/* Cursor */
.typed-cursor {
}

/* If fade out option is set */
.typed-fade-out {
}
```

Customization

```
var typed = new Typed('.element', {
  /**
   * @property {array} strings strings to be typed
   * @property {string} stringsElement ID of element containing string children
   */
  strings: [
    'These are the default values...',
    'You know what you should do?',
    'Use your own!',
    'Have a great day!'
  ],
  stringsElement: null,
  /**
   * @property {number} typeSpeed type speed in milliseconds
   */
  typeSpeed: 0,

  /**
   * @property {number} startDelay time before typing starts in milliseconds
   */
  startDelay: 0,

  /**
   * @property {number} backSpeed backspacing speed in milliseconds
   */
  backSpeed: 0,

  /**
   * @property {boolean} smartBackspace only backspace what doesn't match the previou
   */
  smartBackspace: true,

  /**
   * @property {boolean} shuffle shuffle the strings
   */
  shuffle: false,

  /**
   * @property {number} backDelay time before backspacing in milliseconds
   */
  backDelay: 700,

  /**
   * @property {boolean} fadeOut Fade out instead of backspace
   * @property {string} fadeOutClass css class for fade animation
   * @property {boolean} fadeOutDelay Fade out delay in milliseconds
   */
  fadeOut: false,
  fadeOutClass: 'typed-fade-out',
  fadeOutDelay: 500,

  /**
   * @property {boolean} loop loop strings
   * @property {number} loopCount amount of loops
   */
  loop: false,
  loopCount: Infinity,

  /**
   * @property {boolean} showCursor show cursor
   * @property {string} cursorChar character for cursor
   * @property {boolean} autoInsertCss insert CSS for cursor and fadeOut into HTML <h
   */
  showCursor: true,
  cursorChar: '|',
  autoInsertCss: true,

  /**
   * @property {string} attr attribute for typing
   * Ex: input placeholder, value, or just HTML text
   */
  attr: null,

  /**
   * @property {boolean} bindInputFocusEvents bind to focus and blur if el is text in
   */
  bindInputFocusEvents: false,

  /**
   * @property {string} contentType 'html' or 'null' for plaintext
```

README.md

```
    /**
     * Before it begins typing
     * @param {Typed} self
     */
    onBegin: (self) => {},

    /**
     * All typing is complete
     * @param {Typed} self
     */
    onComplete: (self) => {},

    /**
     * Before each string is typed
     * @param {number} arrayPos
     * @param {Typed} self
     */
    preStringTyped: (arrayPos, self) => {},

    /**
     * After each string is typed
     * @param {number} arrayPos
     * @param {Typed} self
     */
    onStringTyped: (arrayPos, self) => {},

    /**
     * During looping, after last string is typed
     * @param {Typed} self
     */
    onLastStringBackspaced: (self) => {},

    /**
     * Typing has been stopped
     * @param {number} arrayPos
     * @param {Typed} self
     */
    onTypingPaused: (arrayPos, self) => {},

    /**
     * Typing has been started after being stopped
     * @param {number} arrayPos
     * @param {Typed} self
     */
    onTypingResumed: (arrayPos, self) => {},

    /**
     * After reset
     * @param {Typed} self
     */
    onReset: (self) => {},

    /**
     * After stop
     * @param {number} arrayPos
     * @param {Typed} self
     */
    onStop: (arrayPos, self) => {},

    /**
     * After start
     * @param {number} arrayPos
     * @param {Typed} self
     */
    onStart: (arrayPos, self) => {},

    /**
     * After destroy
     * @param {Typed} self
     */
    onDestroy: (self) => {}
  });
```

Contributing

View Contribution Guidelines

end

Thanks for checking this out. If you have any questions, I'll be on Twitter.

If you're using this, let me know! I'd love to see it.

It would also be great if you mentioned me or my website somewhere. www.mattboldt.com

Releases 19

Typed.js v2.0.12 Latest
on 27 Mar 2021

+ 18 releases

Packages

No packages published

Used by 16.8k

+ 16,797

Contributors 32

+ 21 contributors

Environments 1

github-pages Active

Languages

● JavaScript 91.1% ● HTML 7.9% ● CSS 1.0%