Wordle for Good

This project allows you to create a Wordle game for your website, using web components.

How to use

You can generate your wordle game using the Wordle for Good Generator. Or you can follow this documentation to understand how it works.

1. Add the script below to your website:

```
<script
  defer
  src="https://apps.4sitestudios.com/wordle-for-good/wordle-for-good.js"
></script>
```

2. You need to add the `wordle-for-good` tag where you want to display the game:

```
<wordle-for-good
  word="104-101-97-114-116"
  bg-color="#fafafa"
  text-color="#333333"
  tile-border-color="#888888"
  tile-bg-color="#ffffff"
  tile-text-color="#333333"
  key-bg-color="#818283"
  key-text-color="#ffffff"
  key-text-size="1.3rem"
  height="690px"
  title="Wordle For Good"
  success-selector="#success"
  failure-selector="#failure"
></wordle-for-good>
```

You have some options to customize the wordle game. See below for more information.

Attributes

The wordle game has one main attribute:

- `word` - **REQUIRED** The value of this attribute is the result of a 5 letter word, separated by dashes. For example, if you want to use the word "heart", you need to convert it to ASCII and separate each (lowercase) letter with a dash. The result will be `104-101-97-114-116`. You can use this ASCII Table to convert your word to ASCII. If not specified, the game will not render and will throw an error in the console. If you want an easy way to generate the word, you can use the Wordle for Good Generator.

Other Optional Attributes

- `bg-color` - The background color of the game. If not specified, we will use `#fafafa`.
- `text-color` - The text color of the game. If not specified, we will use `#333333`.
- `tile-border-color` - The border color of the tiles. If not specified, we will use `#888888`.
- `tile-bg-color` - The background color of the tiles. If not specified, we will use `#ffffff`.
- `tile-text-color` - The text color of the tiles. If not specified, we will use `#333333`.
- `key-bg-color` - The background color of the keyboard keys. If not specified, we will use `#818283`.
- `key-text-color` - The text color of the keyboard keys. If not specified, we will use `#ffffff`.
- `key-text-size` - The text size of the keyboard keys. If not specified, we will use `1.3rem`.
- `height` - The height of the game. If not specified, we will use `690px`.
- `title` - The title of the game. If not specified, the game will have no Title, but the sharing title will be "Wordle For Good".
- `success-selector` - The selector of the element that will be shown when the user wins the game. If not specified, we will use our own algorithm to guess the element: We will move up the DOM tree until we find an adjacent sibling of the game.
- `failure-selector` - The selector of the element that will be shown when the user loses the game. If not specified, we will use the same element as the `success-selector`.

Development

Your code must be on the `src` folder. Style files must be on the `src/styles` folder. The `./index.html` file is used to test your code. The `dist` folder is generated automatically. When you are ready to deploy, run `npm run build` to generate the `dist` folder, then use the `dist/wordle-for-good.js` file in your website.

Install Dependencies

1. `npm install`

Deploy

1. `npm run build` - Builds the project

It's going to create a `dist` folder, where you can get the `dist/wordle-for-good.js` file and publish it.

Hot Module Reloading

1. `npm run dev` - Starts the server with hot reloading enabled

Recommended IDE Setup

- VS Code + Volar

Type Support For `.vue` Imports in TS

Since TypeScript cannot handle type information for `.vue` imports, they are shimmed to be a generic Vue component type by default. In most cases this is fine if you don't really care about component prop types outside of templates. However, if you wish to get actual prop types in `.vue` imports (for example to get props validation when using manual `h(...)` calls), you can enable Volar's Take Over mode by following these steps:

1. Run `Extensions: Show Built-in Extensions` from VS Code's command palette, look for `TypeScript and JavaScript Language Features`, then right click and select `Disable (Workspace)`. By default, Take Over mode will enable itself if the default TypeScript extension is disabled.
2. Reload the VS Code window by running `Developer: Reload Window` from the command palette.

You can learn more about Take Over mode here.

Demo

https://apps.4sitestudios.com/wordle-for-good/test/index.html

It's currently published on 4Site's APP AWS EC2 Server: https://apps.4sitestudios.com/wordle-for-good/wordle-for-good.js