Unit testing native Web Components

Hello!

I am Andy Desmarais

Code Ninja/Web Geek

You can find me on all the social medias @terodox

Website: terodox.tech

Web Components

DOM native components, no framework

Basics of Web Components

Extend HTMLElement

Attach shadow

Define new custom element

```
class WebComponentExample extends HTMLElement
{


    constructor() {
        super();
        this.attachShadow({ mode: 'open' });
    }
}
customElements.define(
    'web-component-example-tag-name',
    WebComponentExample);
```

What's the problem?

We need DOM native functionality for testing, but starting a browser is slow.

jsdom is missing some pieces

Web Components need

- HTMLElement
- customElements
- Shadow DOM

These are not available in jsdom (yet)

- Invert control

○ Inject the window!?

   ◦ import { HTMLElement, customElements } from 'window';

Webpack - use externals

- module.exports = {
    ...
    externals: { window: 'window' }
  };

Rollup - use globals

- export default {
      ...
      output: {
          ...
          globals: { window: 'window' }
      }
  };

ESM - Create a window.mjs

- export const HTMLElement = window.HTMLElement;
  export const customElements = window.customElements;

WE HAVE A SEAM

Testing is now REALLY straight forward

Fake it 'til you make it

```
const window = {
    HTMLElement: class HTMLElement {
        attachShadow() {
            this.shadowRoot = document.createElement('div');
        }
    },
    customElements: {
        define: () => {}
    }
};
Object.freeze(window);
module.exports = window;
```

Examples on github

https://github.com/terodox/dotjs2019