Web Components for Design Systems

Group 1
Perner Elke, Pollak Michael, Reichmann Oliver, Sackl Martin

Information Architecture and Web Usability

02. December 2020

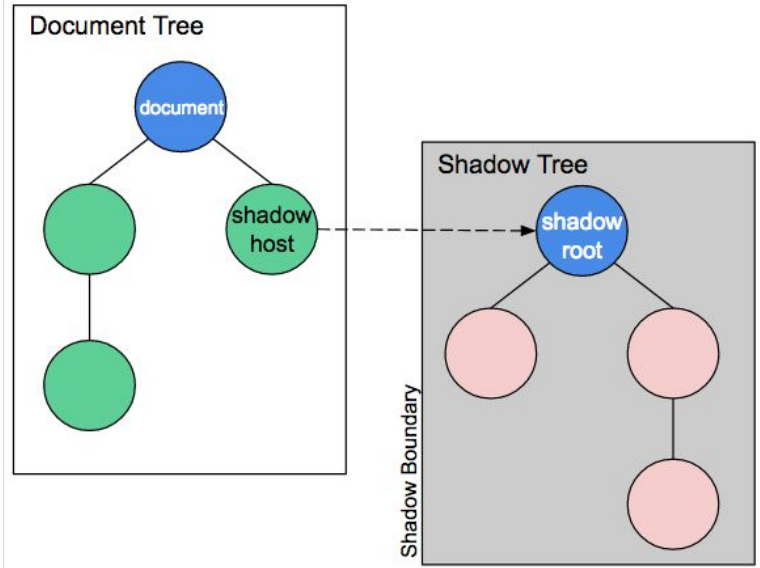
What are Web Components?

- Truly encapsulated UI in a single <tag/>.
- Reusable code block.
- No functionality or style conflicts with other elements.
- Must be custom HTML elements.
- Use of shadow DOMs.

```
<my-custom-element></my-custom-element>
```

Shadow Document Object Model

Shadow DOMs are independent code shells with their own range of validity within a normal DOM.



- Used by ~every Web Component library
- Encapsulating HTML with related CSS
- Not visible for debugger and outside scripts

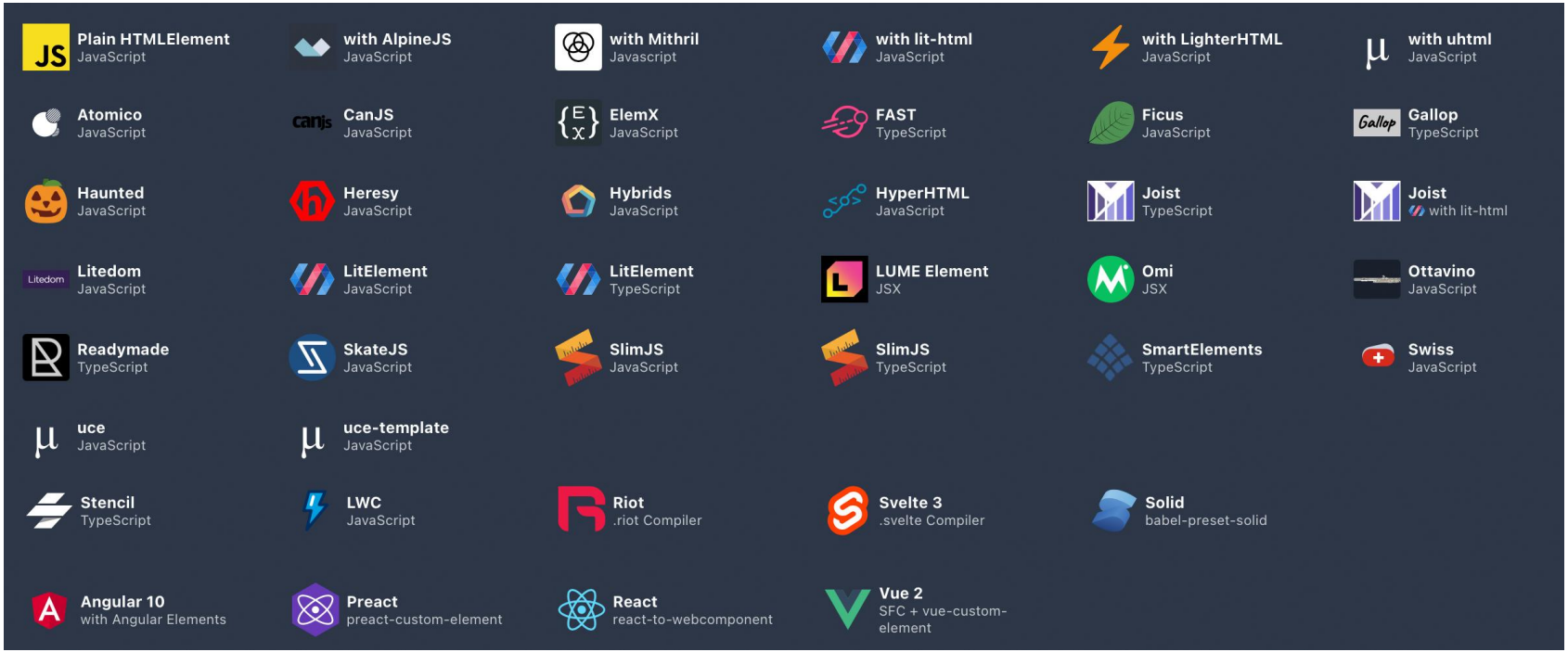
Source: developer.mozilla.org/en-US/docs/Web/Web_Components/Using_shadow_DOM

TypeScript

- Built on JavaScript
- Types describe the shape of an object
- Better documentation
- Validation of code
- Web Component libraries supporting TypeScript?

Source: www.typescriptlang.org

Web Component Libraries



Source: webcomponents.dev/new/

Two Categories

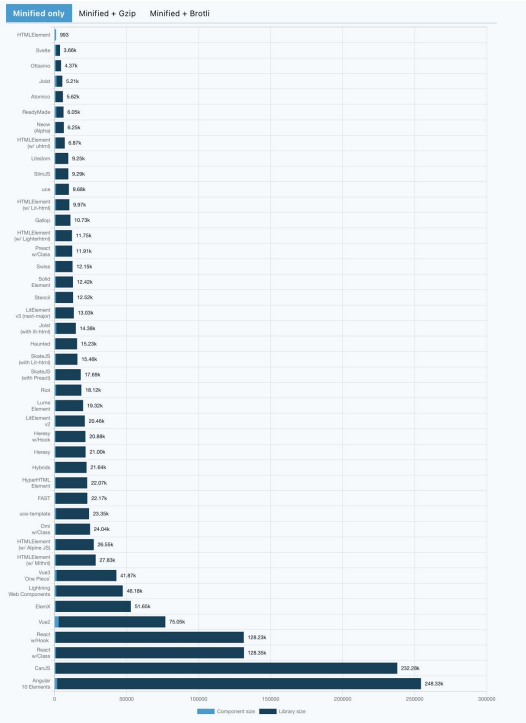
- Compile the code? → **Compiler-Based**
  - source code compiled into native Custom Element
- Else → **Class-Based**
  - object-based,
  - hook-based,
  - in general class-based

Parameters for Ranking

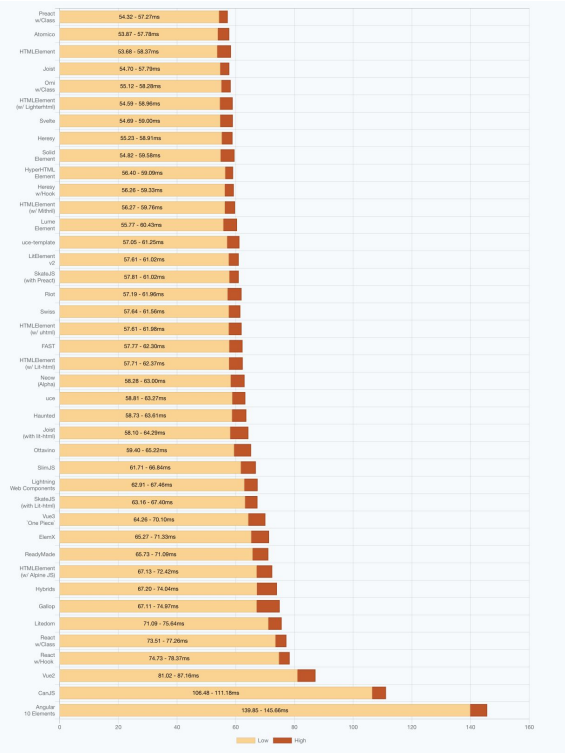
- User Base — ranking based on npm and github (downloads, forks, …)
- Bundle Size — webcomponents.dev
- TypeScript Support — individual web component website
- Performance in Browser — webcomponents.dev
- Provides Component Library — **individual web component website**
- Good Documentation — ranking from 1-10 by us (subjective)

Bundle Size and Performance

- Smallest: HTMLElements
  993 kb
- Biggest: Angular
  248330 kb



- Fastest: Preact
  55 ms
- Slowest: Angular
  150 ms



Sources: webcomponents.dev/blog/all-the-ways-to-make-a-web-component

Ranking of Web Component Libraries

	Name	TypeScript Support	Provides Component Library	Size [1] (minified, in kb)	Performance [1] (ms)	User Base [2]	Documentation
	HTMLElements			993	57.27	10	10
Compiler -Based	**Stencil**	Yes	No	12,520	-	2	9
	Lightning Web Components	Yes	Yes	46,180	67.46	6	10
	Riot	Preprocessor	No	18,120	61.96	1	8
	Svelte	Preprocessor	Yes	3,660	59.00	6	7
Class -Based	**LitElement**	Yes	No	13,030	61.02	9	10
	FAST	Yes	Yes	22,170	62.30	3	7
	Ottavino	No	No	4,370	65.22	4	5

1) statistics from: www.webcomponents.dev/blog/all-the-ways-to-make-a-web-component
2) statistics from: github and npm

Class-Based

	FAST	LitElements	Ottavino
Stars on Github	5,000	3,800	64
Weekly Downloads on npm	37,400	30,000	8
Size*	22,170 kb	13,030 kb	4,370 kb

* statistics from: www.webcomponents.dev/blog/all-the-ways-to-make-a-web-component

Code Example - OK/Cancel Button

HTML Element and LitElement

https://www.youtube.com/watch?v=DwhcgEnRJJo&feature=youtu.be

Compiler based

	Stencil	Lightning	Riot	Svelte
Stars on Github	8,500	893,000	14,300	39,400
Weekly Downloads on npm	37,400	2,400	6,100	104,000
Size*	12,520 kb	46,180 kb	18,120 kb	3,660 kb

* statistics from: www.webcomponents.dev/blog/all-the-ways-to-make-a-web-component

Code Example - OK/Cancel Button

Stencil

https://www.youtube.com/watch?v=ng4eQVTxwX8&feature=youtu.be

Recommendation

- **Stencil** easy and simple
  - documentation is very good
  - npm installation - works perfect
  - easy coding & integration

- **LitElements** easy and simple, again
  - Prebuilt starter package
    https://github.com/PolymerLabs/lit-element-starter-js
  - Perfect documentation for beginners
  - Close to HTMLElements but more readable

Design System from Web Components

- CSS: hard for big systems.
- Web components: Shadow DOMs.
- Problem : sometimes we want style from outside →build design system
- Combining custom web components:
  - consistent and repeatable custom design system
- BUT: Shadow DOM

Design System from Web Components

- Solution: Constructable Stylesheets
- CSS file(s) from design system: adopted by any Web Component
- Sheets shared
- For development: Storybook → open source Sandbox tool
- Many components libraries available (see full survey paper)

Questions

Not Recommended

- **Lightning** complex and uncomfortable
  - npm installation - nearly impossible,
  - npm publish necessary

- **FAST** complex and uncomfortable
  - No starter documentation.
  - Forces prebuilt elements.