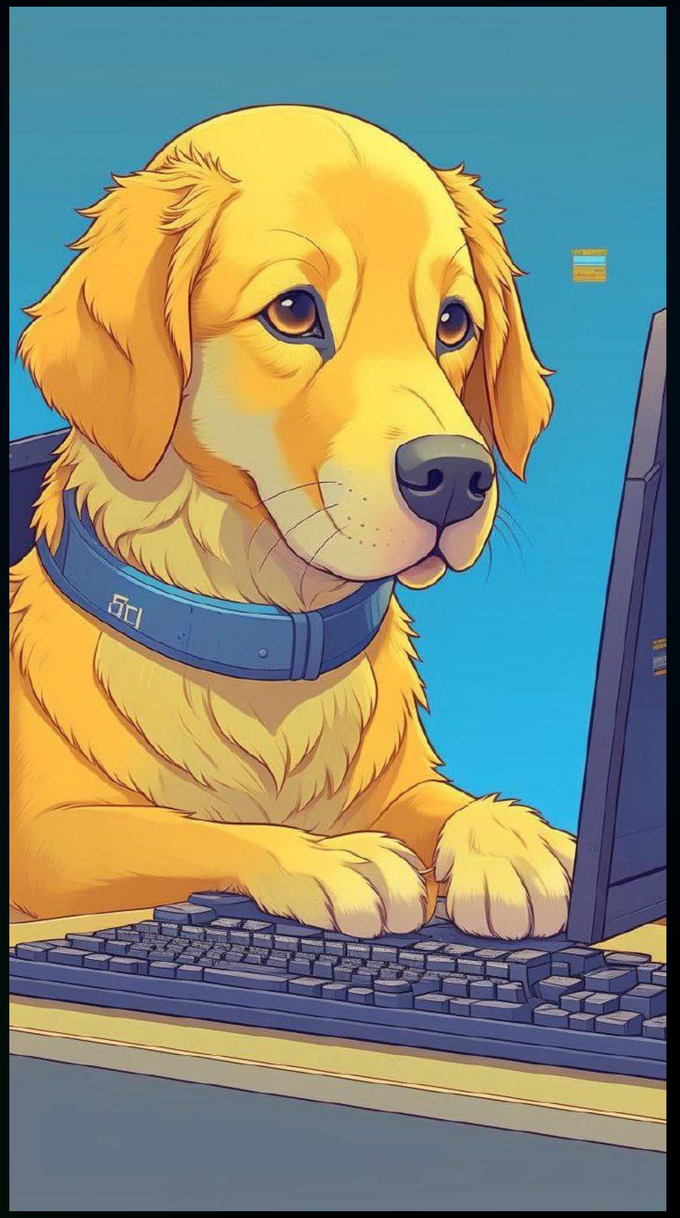
СБЕР

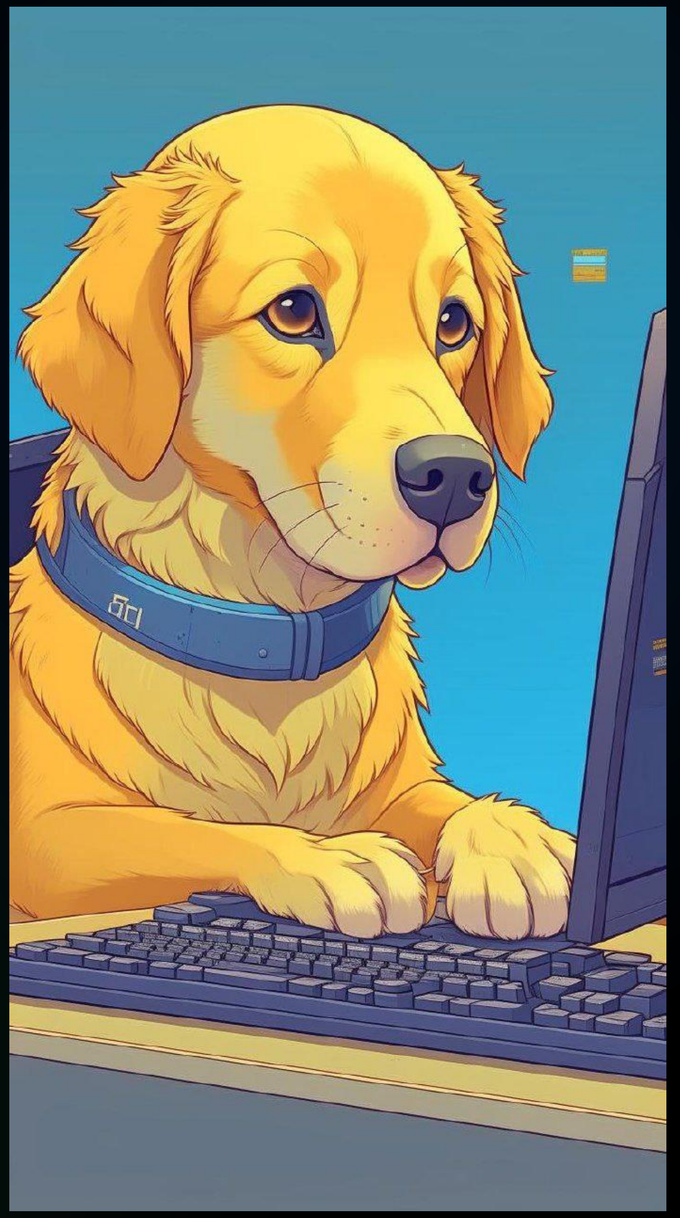
Как подружить веб-компоненты и JS-фреймворки

Троицкий Роман
ПАО Сбербанк
HolyJS Spring 2024

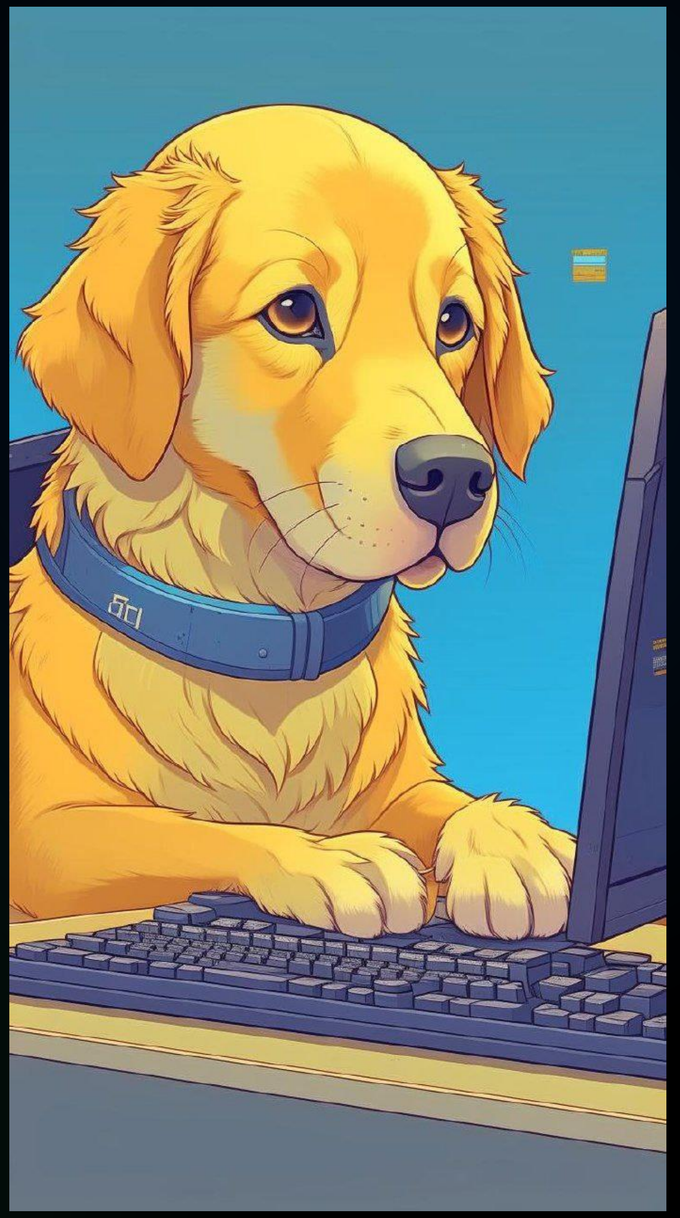
- Влюблен во фронтенд;



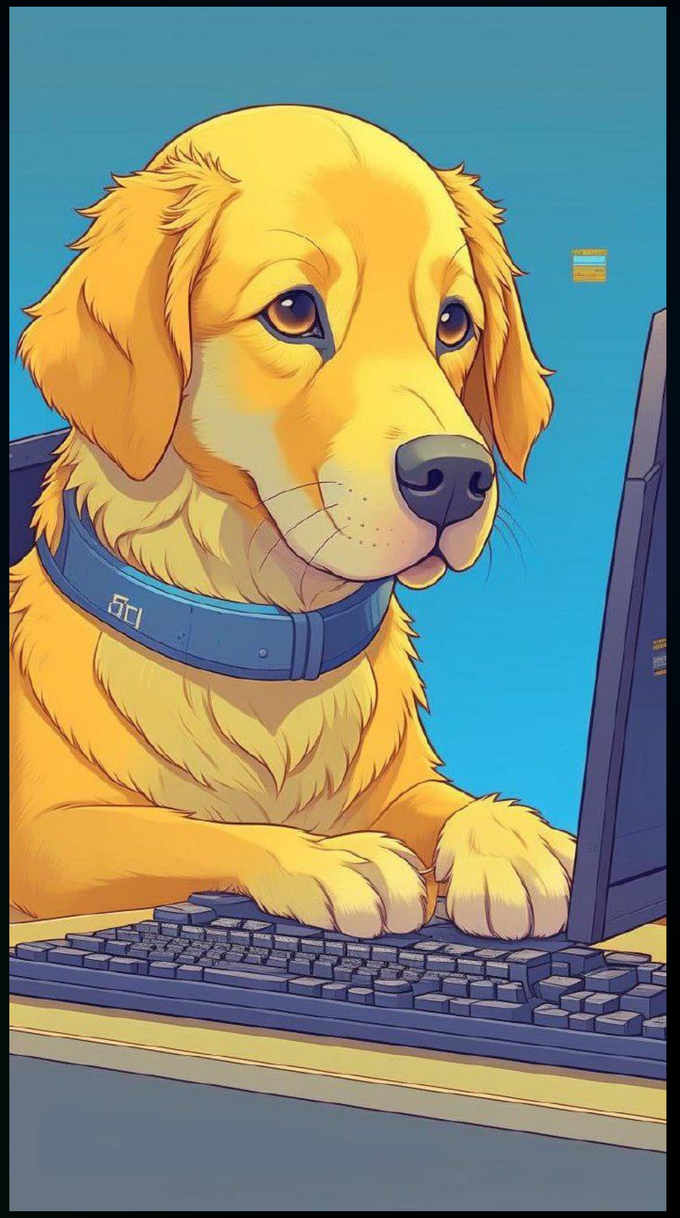
- Влюблен во фронтенд;
- Проекты на AWWWARDS, Tagline, GoldenSite, etc;



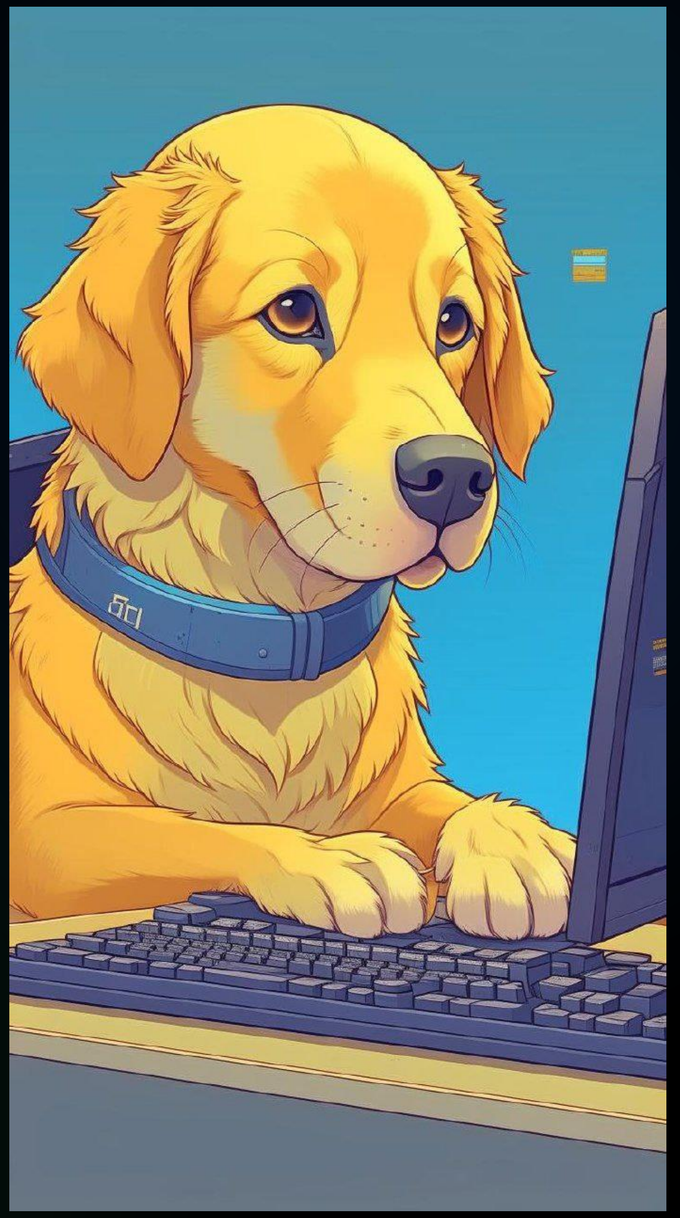
- Влюблен во фронтенд;
- Проекты на AWWWARDS, Tagline, GoldenSite, etc;
- MoscowCSS coopr;



- Влюблен во фронтенд;
- Проекты на AWWWARDS, Tagline, GoldenSite, etc;
- MoscowCSS coopr;
- Skillbox Code Expert;



- Влюблен во фронтенд;
- Проекты на AWWWARDS, Tagline, GoldenSite, etc;
- MoscowCSS coopr;
- Skillbox Code Expert;
- vzhyx.io (Q5 202X).



Комьюнити веб-разработчиков by Fusion Brain

Зачем нужны веб-компоненты


?
22
@

- Headless UI-kit

- Headless UI-kit

- Headless UI-kit
- Mitosis & etc

stencil (Workspace) [WSL: Debian] - mitosis.tsx

```tsx
export function MyComponent() {
  const state = useStore({
    name: 'Vzhyx',
  });

  return (
    <div>
      <input value={state.name} onChange={(event) => (state.name = event.target.value)} />
    </div>
  );
}
```

stencil (Workspace) [WSL: Debian] - mitosis.json

```
{
  "@type": "@builder.io/mitosis/component",
  "state": {
    "name": "Vzhyx"
  },
  "nodes": [
    {
      "@type": "@builder.io/mitosis/node",
      "name": "div",
      "children": [
        {
          "@type": "@builder.io/mitosis/node",
          "name": "input",
          "bindings": {
            "value": "state.name",
            "onChange": "state.name = event.target.value"
          }
        }
      ]
    }
  ]
}

```

stencil (Workspace) [WSL: Debian] - mitosis.json

```
{
  "@type": "@builder.io/mitosis/component",
  "state": {
    "name": "Vzhyx"
  },
  "nodes": [
    {
      "@type": "@builder.io/mitosis/node",
      "name": "div",
      "children": [
        {
          "@type": "@builder.io/mitosis/node",
          "name": "input",
          "bindings": {
            "value": "state.name",
            "onChange": "state.name = event.target.value"
          }
        }
      ]
    }
  ]
}

```

stencil (Workspace) [WSL: Debian] - mitosis.json

```
{
  "@type": "@builder.io/mitosis/component",
  "state": {
    "name": "Vzhyx"
  },
  "nodes": [
    {
      "@type": "@builder.io/mitosis/node",
      "name": "div",
      "children": [
        {
          "@type": "@builder.io/mitosis/node",
          "name": "input",
          "bindings": {
            "value": "state.name",
            "onChange": "state.name = event.target.value"
          }
        }
      ]
    }
  ]
}

```

stencil (Workspace) [WSL: Debian] - mitosis.json

```
{
  "@type": "@builder.io/mitosis/component",
  "state": {
    "name": "Vzhyx"
  },
  "nodes": [
    {
      "@type": "@builder.io/mitosis/node",
      "name": "div",
      "children": [
        {
          "@type": "@builder.io/mitosis/node",
          "name": "input",
          "bindings": {
            "value": "state.name",
            "onChange": "state.name = event.target.value"
          }
        }
      ]
    }
  ]
}

```

stencil (Workspace) [WSL: Debian] - mitosis.component.ts

```
1  @Component({
2    template: `
3      <div>
4        <input [value]="name" (change)="name = $event.target.value" />
5      </div>
6    `,
7  })
8  class MyComponent {
9    name = 'Vzhyx';
10 }
11
```

- Headless UI-kit
- Mitosis & etc
- Веб-компоненты

НЕ ПРОСТО СМЕЛО,
ЭТО БЫЛО
ПИЗДЕЦ КАК СМЕЛО


WEB COMPONENTS

stencil (Workspace) [WSL: Debian] - dropdown.html

```
47  <dropdown-menu>
48    <span slot="item">Опция 1</span>
49    <span slot="item">Опция 2</span>
50    <span slot="item">Опция 3</span>
51  </dropdown-menu>
```

stencil (Workspace) [WSL: Debian] - index.js

```
15  class WebDropdown extends HTMLElement {
16    constructor() {
17      super();
18      const template = document.getElementById('dropdown-template');
19      const templateContent = template.content;
20    }
21  }
```

stencil (Workspace) [WSL: Debian] - index.js

```
15  class WebDropdown extends HTMLElement {
16    constructor() {
17      super();
18      const template = document.getElementById('dropdown-template');
19      const templateContent = template.content;
20
21      const shadowRoot = this.attachShadow({mode: 'open'});
22      shadowRoot.appendChild(templateContent.cloneNode(true));
23    }
24  }
```

stencil (Workspace) [WSL: Debian] - index.js

```
15  class WebDropdown extends HTMLElement {
16    constructor() {
17      super();
18      const template = document.getElementById('dropdown-template');
19      const templateContent = template.content;
20
21      const shadowRoot = this.attachShadow({mode: 'open'});
22      shadowRoot.appendChild(templateContent.cloneNode(true));
23    }
24  }
```

stencil (Workspace) [WSL: Debian] - index.js

```
27  customElements.define('dropdown-menu', WebDropdown);
```


Жми меня

Жми меня

Опция 1
Опция 2
Опция 3


<body>
<template id="dropdown-template">…</template>
<dropdown-menu>
#shadow-root (open)
<style>…</style>

<button onclick="...">Жми меня</button>

<slot name="item" class="item">
flex
↪
reveal
↪
reveal
↪
reveal
</slot>

Опция 1
slot
Опция 2
slot
Опция 3
slot
</dropdown-menu>


<body>
<template id="dropdown-template">…</template>
<dropdown-menu>
#shadow-root (open)
<style>…</style>

<button onclick="...">Жми меня</button>

<slot name="item" class="item">
flex

reveal

reveal

reveal
</slot>

Опция 1
slot
Опция 2
slot
Опция 3
slot
</dropdown-menu>

stencil (Workspace) [WSL: Debian] - index.js

```
15  class WebDropdown extends HTMLElement {
16    constructor() {
17      super();
18      const template = document.getElementById('dropdown-template');
19      const templateContent = template.content;
20
21      const shadowRoot = this.attachShadow({mode: 'open'});
22      shadowRoot.appendChild(templateContent.cloneNode(true));
23    }
24  }
```


<body>
<template id="dropdown-template">…</template>
<dropdown-menu>
#shadow-root (open)
<style>…</style>

<button onclick="...">Жми меня</button>

<slot name="item" class="item">
flex

reveal

reveal

reveal
</slot>

Опция 1
slot
Опция 2
slot
Опция 3
slot
</dropdown-menu>


▼ <body>
▶ <template id="dropdown-template"> ••• </template>
▼ <dropdown-menu>
▼ #shadow-root (open)
▶ <style> ••• </style>
▼
<button onclick="...">Жми меня</button>
▼
▼ <slot name="item" class="item"> flex
↪ reveal
↪ reveal
↪ reveal
</slot>

Опция 1 slot
Опция 2 slot
Опция 3 slot
</dropdown-menu>

stencil (Workspace) [WSL: Debian] - index.js

```
1  class WebInput extends HTMLElement {
2    constructor() {
3      const template = document.getElementById('input-template');
4      ...
5
6      const shadowRoot = this.attachShadow({mode: 'open'});
7      ...
8    }
9  }
```

stencil (Workspace) [WSL: Debian] - index.js

```
1  class WebInput extends HTMLElement {
2    constructor() {
3      const template = document.getElementById('input-template');
4      ...
5
6      const shadowRoot = this.attachShadow({mode: 'open'});
7      ...
8    }
9  }
```

stencil (Workspace) [WSL: Debian] - index.js

```
1  class WebInput extends HTMLElement {
2    constructor() {
3      const template = document.getElementById('input-template');
4      ...
5
6      const shadowRoot = this.attachShadow({mode: 'open'});
7      ...
8    }
9  }
```

stencil (Workspace) [WSL: Debian] - index.js

```
1  class WebInput extends HTMLElement {
2    constructor() {
3      const template = document.getElementById('input-template');
4      ...
5
6      const shadowRoot = this.attachShadow({mode: 'open'});
7      ...
8    }
9  }
```


Введите имя

stencil (Workspace) [WSL: Debian] - 3_shadow_dom.html

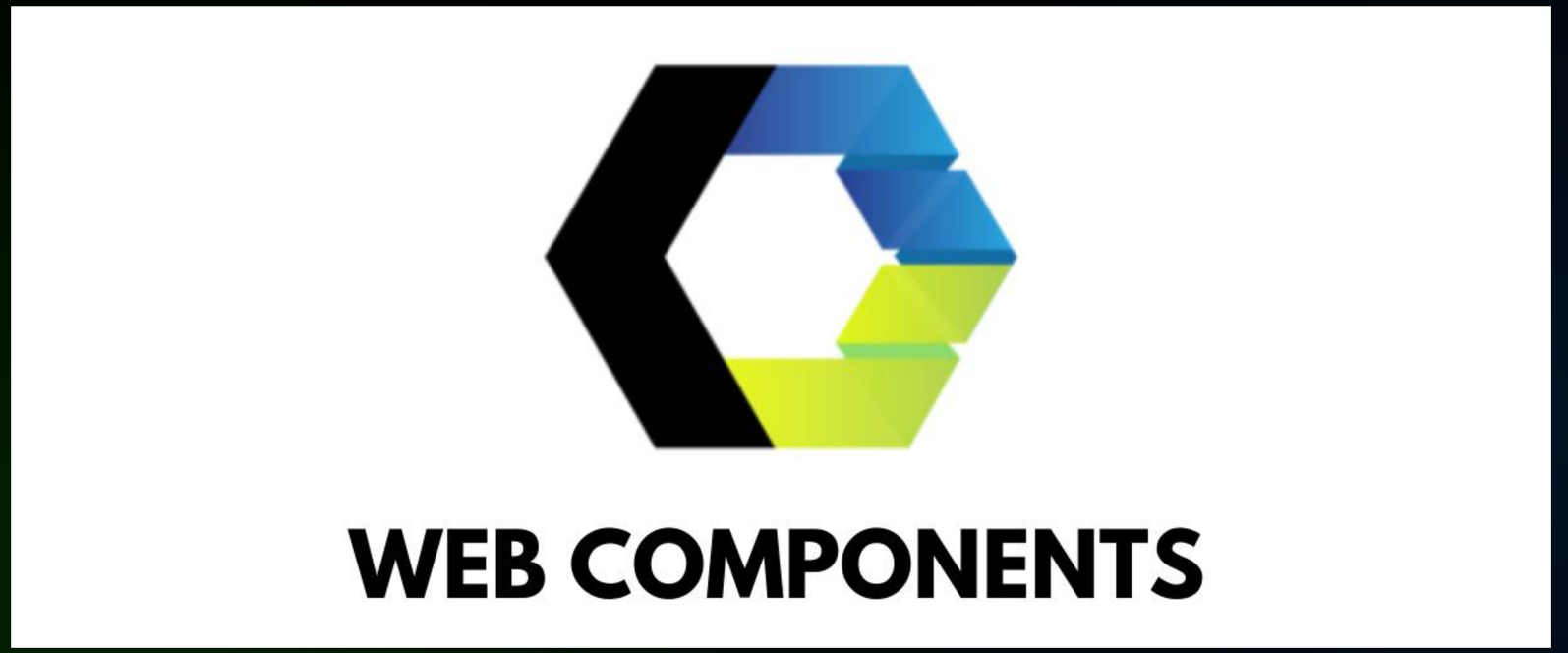
```
29  static formAssociated = true;
```

stencil (Workspace) [WSL: Debian] - 3_shadow_dom.html

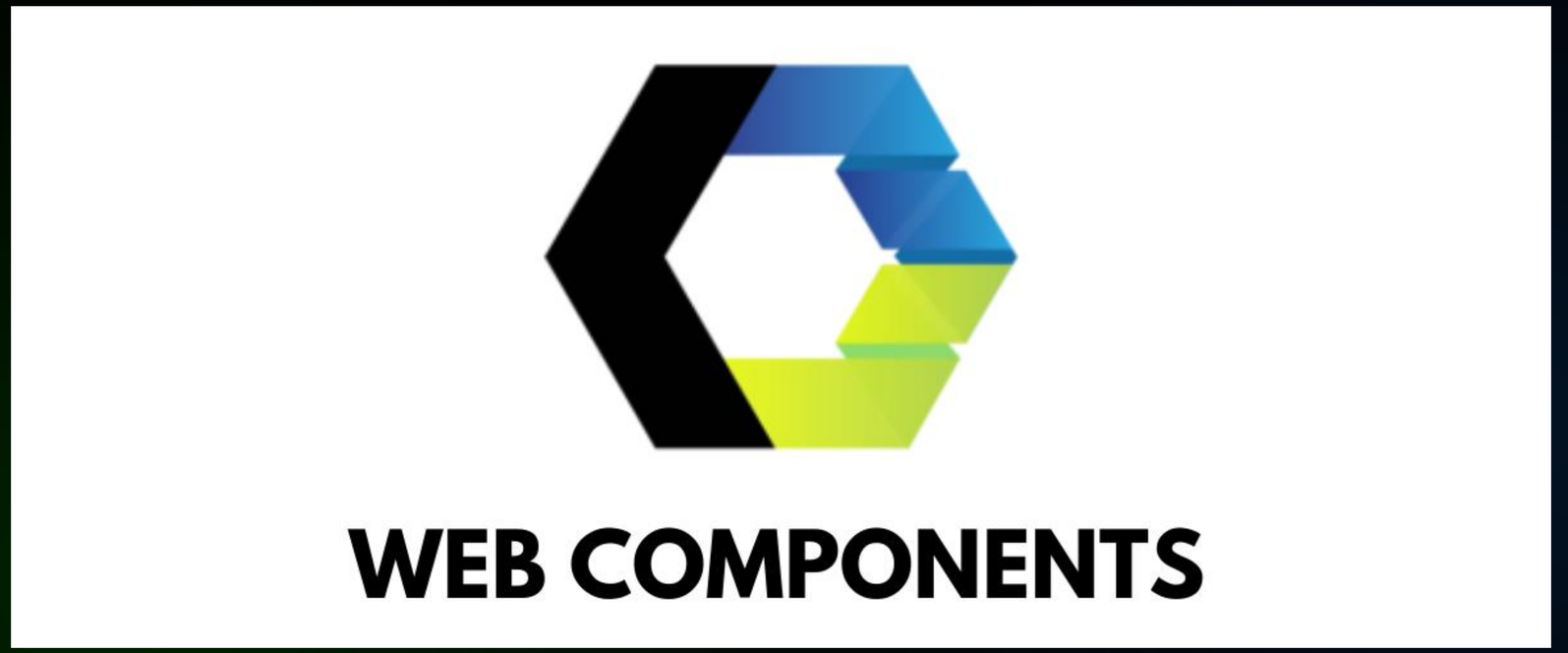
```
29  static formAssociated = true;
```

stencil (Workspace) [WSL: Debian] - 3_shadow_dom.html

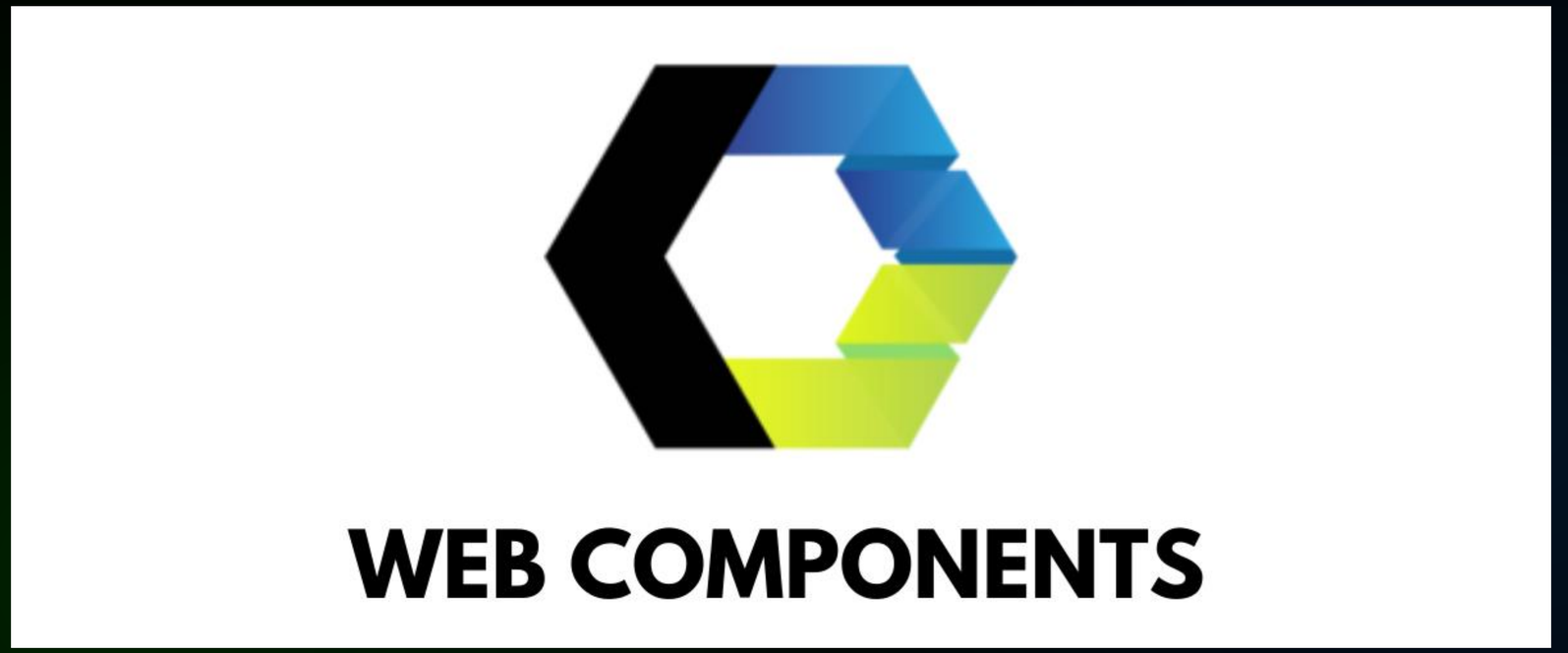
```
38  get value() { return this.value_; }
39  set value(v) { this.value_ = v; }
```



1. Custom Elements.



1. Custom Elements.
2. Shadow DOM.



1. Custom Elements.
2. Shadow DOM.
3. HTML <template> и <slot>.

Плюсы

- Нативность;

Плюсы

- Нативность;
- Инкапсуляция;

Плюсы

- Нативность;
- Инкапсуляция;
- Переиспользование;

Плюсы

- Нативность;
- Инкапсуляция;
- Переиспользование;
- Агностицизм;

Плюсы

- Нативность;
- Инкапсуляция;
- Переиспользование;
- Агностицизм;
- Меньше зависимостей;

Плюсы

- Нативность;
- Инкапсуляция;
- Переиспользование;
- Агностицизм;
- Меньше зависимостей;
- Перформанс.

Минусы

- Всплытие и прослушка событий;

Минусы

- Всплытие и прослушка событий;
- Пропсы - строки;

Минусы

- Всплытие и прослушка событий;
- Пропсы - строки;
- Связывание данных;

Минусы

- Всплытие и прослушка событий;
- Пропсы - строки;
- Связывание данных;
- Много работы с DOM Api;

Минусы

- Всплытие и прослушка событий;
- Пропсы - строки;
- Связывание данных;
- Много работы с DOM Api;
- Доступность;

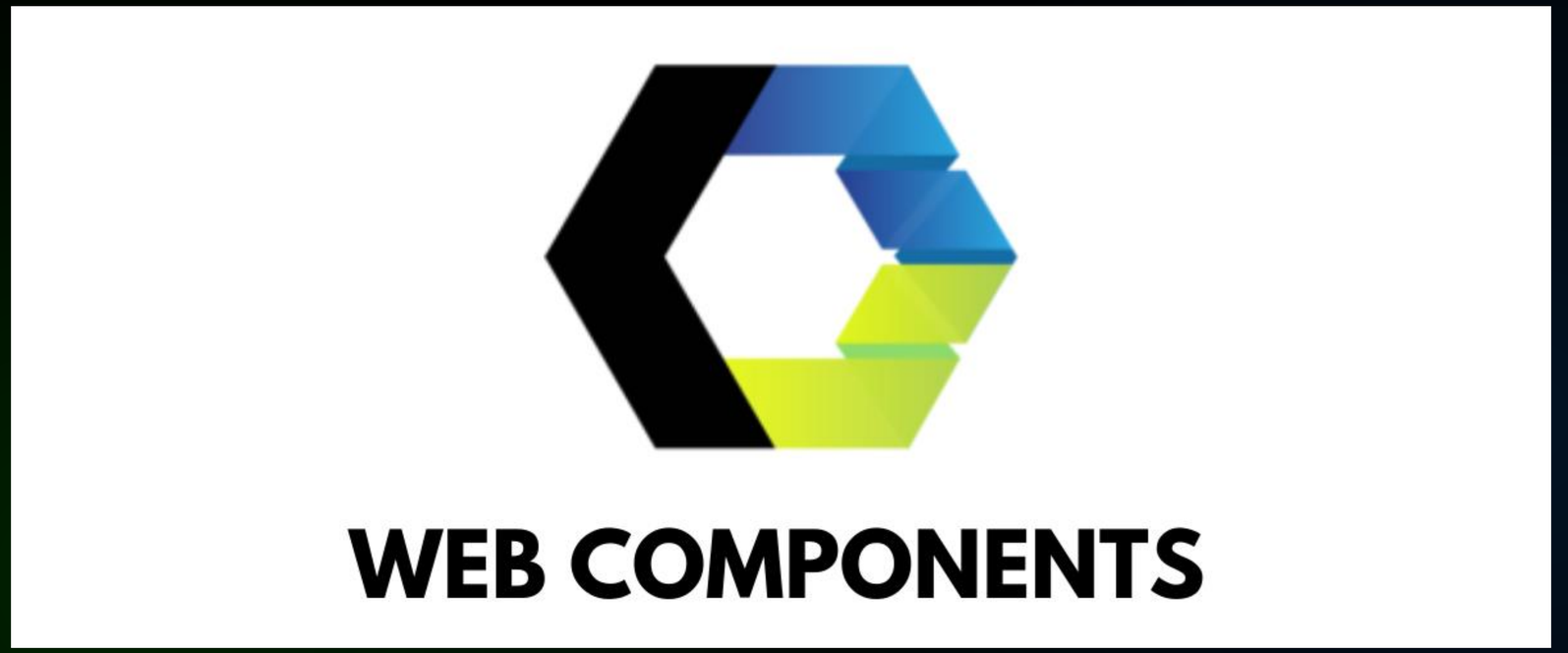
Минусы

- Всплытие и прослушка событий;
- Пропсы - строки;
- Связывание данных;
- Много работы с DOM Api;
- Доступность;
- SEO;

Минусы

- Всплытие и прослушка событий;
- Пропсы - строки;
- Связывание данных;
- Много работы с DOM Api;
- Доступность;
- SEO;
- Низкая популярность.

ВОТ МНЕ ЛИЧНО ЭТО ИНТЕРЕСНО
ЗА ДРУГИХ СКАЗАТЬ НЕ МОГУ

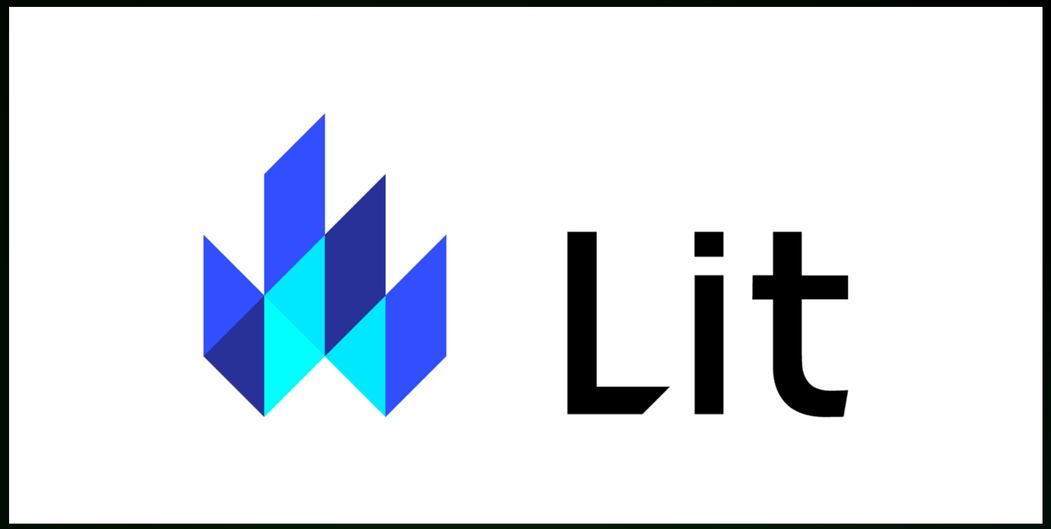


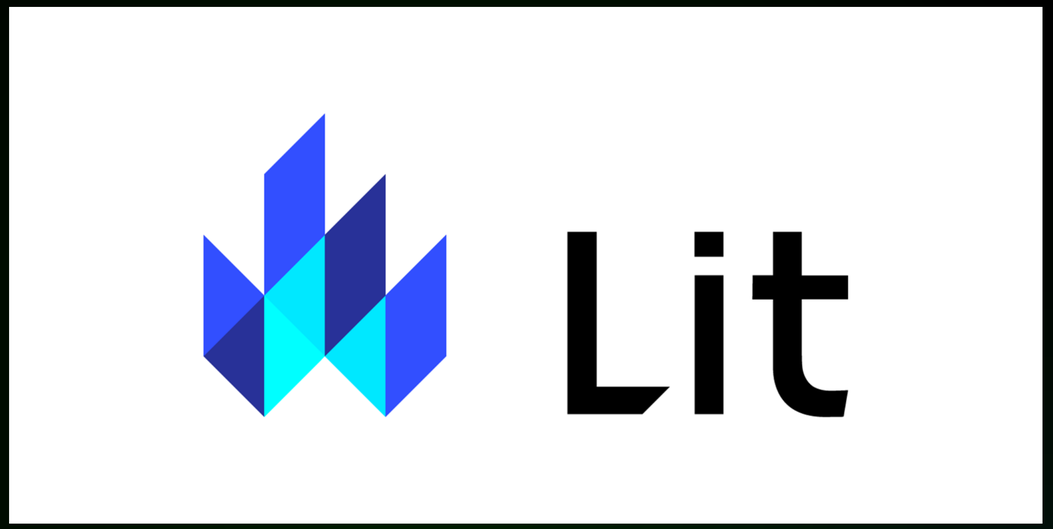
1. Custom Elements.
2. Shadow DOM.
3. HTML <template> и <slot>.
4. Declarative Shadow DOM


YouTube


SPACEX


Adobe

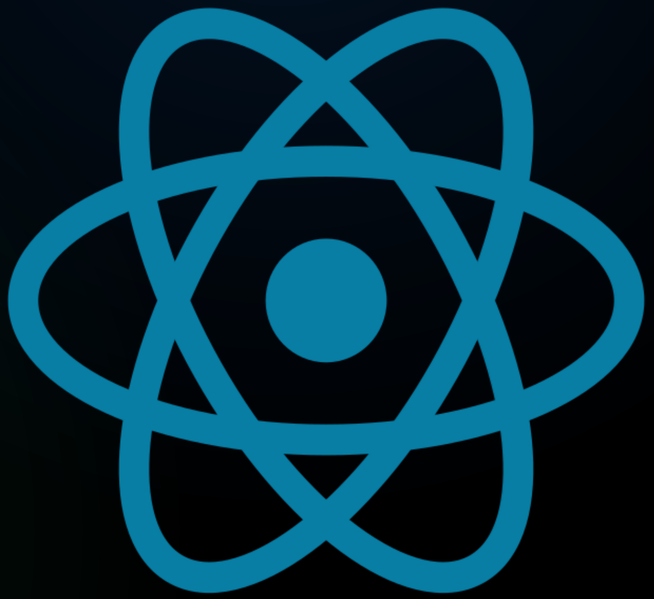

Lit


Lit


stencil

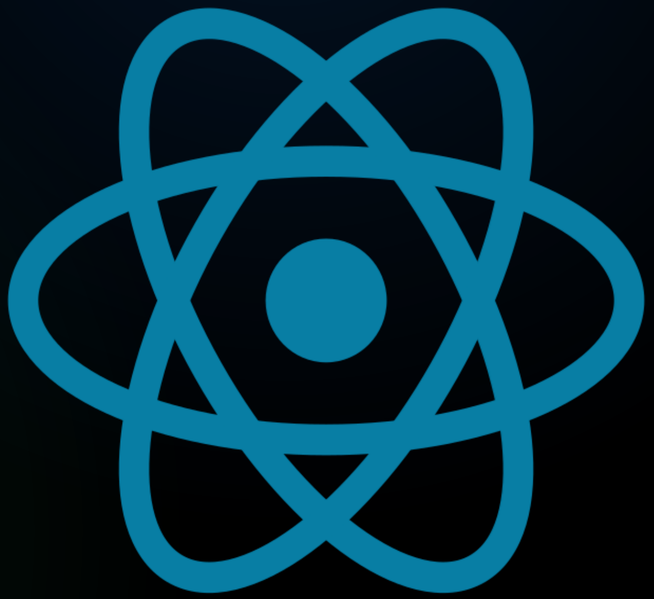
SSR

SSR



SSR

CLI



stencil (Workspace) - ChamomileInput.tsx

```
 8  @Component({
 9    tag: 'my-input',
10    styleUrl: 'my-input.css',
11    shadow: true,
12  })
```

stencil (Workspace) - ChamomileInput.tsx

```
14  export class MyInput {
15    @Prop() validationType: string;
16    @Prop({ mutable: true }) type: string = 'text';
17
18    @State() errorMessage: string = '';
19
20    @Event() onInput: EventEmitter<string>;
21    ...
22  }
```

stencil (Workspace) - ChamomileInput.tsx

```
14  export class MyInput {
15    @Prop() validationType: string;
16    @Prop({ mutable: true }) type: string = 'text';
17
18    @State() errorMessage: string = '';
19
20    @Event() onInput: EventEmitter<string>;
21    ...
22  }
```

stencil (Workspace) - ChamomileInput.tsx

```
14  export class MyInput {
15    @Prop() validationType: string;
16    @Prop({ mutable: true }) type: string = 'text';
17
18    @State() errorMessage: string = '';
19
20    @Event() onInput: EventEmitter<string>;
21    ...
22  }
```

stencil (Workspace) - ChamomileInput.tsx

```
27  render() {
28    return (
29      <label>
30        <input
31          type={this.type}
32          onInput={(e: any) => {
33            this.validate(e.target.value);
34            this.onInput.emit(e.target.value);
35          }}
36        />
37      </label>
38    )
39  }
```

stencil (Workspace) - stencil.config.ts

```
2  import { vueOutputTarget } from '@stencil/vue-output-target';
3  import { reactOutputTarget } from '@stencil/react-output-target';
```

stencil (Workspace) - stencil.config.ts

```
 5  export const config: Config = {
 6    outputTargets: [
 7      ...
 8       vueOutputTarget({
 9        componentCorePackage: 'chamomile-vue',
10        proxiesFile: '../vue/src/components-stencil.ts',
11      }),
12      reactOutputTarget({
13        componentCorePackage: 'chamomile-react',
14        proxiesFile: '../react/src/components-stencil.ts',
15      }),
16    ],
17    ...
18  }
```

КРУТО! ДА ЭТО Ж КРУТО

stencil (Workspace) - main.ts

```
4  import { ComponentLibrary } from 'chamomile-vue';
5
6  createApp(App)
7    .use(ComponentLibrary)
8    .mount('#app');
```

stencil (Workspace) - App.vue

```
4  <template>
5    <my-input
6      placeholder="Введите имя"
7      type="text"
8      default-value="Vue тащит"
9      @value-changed="console.log($event)"
10   />
11 </template>
```

stencil (Workspace) - stencil.config.ts

```
5  export const config: Config = {
6    outputTargets: [
7      {
8        type: 'docs-json',
9        file: '../docs.json'
10     },
11     {
12       type: 'docs-readme',
13     }
14     ...
15   ],
16   ...
17 }
```

stencil (Workspace) - ChamomileInput.tsx

```
 4  /**
 5   * Компонент Input с валидацией
 6   * Работает, как часы)
 7   */
 8  @Component({
 9    tag: 'my-input',
10    styleUrl: 'my-input.css',
11    shadow: true,
12  })
```

stencil (Workspace) - ChamomileInput.tsx

```
14  export class MyInput {
15    /**
16    * Пропс попадёт в документацию
17    */
18    @Prop() validationType: string;
19
20    /**
21     * Стейт не попадёт в документацию
22    */
23    @State() errorMessage: string = '';
24    ...
25  }
```

@pxtrn/storybook-addon-docs-stencil

Converts stencil.js JsonDoc to storybook ArgTypes

```
npm install @pxtrn/storybook-addon-docs-stencil
```

Last updated about 2 years ago

README View on GitHub

storybook-addon-docs-stencil

Converts stencil.js doc json derived from stencils output target `docs-json` to storybook `ArgTypes`.

With this addon activated

- Storybook will render basic controls for properties Controls.
- Storybook will auto generate documentation for Props, Events, Methods, Slots, Shadow Parts and Custom Properties.
- Storybook doc page will contain stencils component documentation (readme.md or inline)

Duration in milliseconds ± 95% confidence interval (Slowdown = Duration / Fastest)

Name	vue-v3.4.3	angular-cf-v17.0.2	react-hooks-v18.2.0	stencil-v4.4.1
create rows	47.5	47.2	48.8	48.9
replace all rows	51.9	56.8	57.0	58.3
partial update	20.6	18.5	22.1	32.7
select row	4.7	4.4	5.6	18.3
swap rows	21.9	21.6	174.3	35.0
remove row	20.5	17.9	18.2	24.6
create many rows	462.8	469.1	622.5	487.1
append rows to large table	51.5	51.4	54.3	59.1
clear rows	16.7	27.5	27.2	16.8

Memory allocation in MBs ± 95% confidence interval

Name	vue-v3.4.3	angular-cf-v17.0.2	react-hooks-v18.2.0	stencil-v4.4.1
ready memory	0.7	1.4	1.0	0.5
run memory	3.8	4.7	4.4	3.3
update every 10th row for 1k rows (5 cycles)	3.8	4.8	5.0	3.4
creating/clearing 1k rows (5 cycles)	1.1	2.1	1.8	0.8
run memory 10k	28.3	29.7	32.2	27.9

Transferred size (in kBs) and first paint

Name	vue-v3.4.3	angular-cf-v17.0.2	react-hooks-v18.2.0	stencil-v4.4.1
uncompressed size	60.2	137.7	142.3	11.9
compressed size	21.4	43.0	40.1	5.0
first paint	105.1	201.5	212.9	75.0

Субъективщина

Проект

- Графики, таблицы;

Проект

- Графики, таблицы;
- StencilJS;

Проект

- Графики, таблицы;
- StencilJS;
- Простые вещи - да;

Проект

- Графики, таблицы;
- StencilJS;
- Простые вещи - да;
- Замкнутые в себе - да;

Проект

- Графики, таблицы;
- StencilJS;
- Простые вещи - да;
- Замкнутые в себе - да;
- Работает.

СОМНИТЕЛЬНО, НО ОК

Выводы

1. Технология жива и развивается.

Выводы

1. Технология жива и развивается.
2. Учесть плюсы и минусы.

Выводы

1. Технология жива и развивается.
2. Учесть плюсы и минусы.
3. Ждём…

Спасибо

Материалы

ЛС телеграм