

Особенности CSS в веб-компонентах

Гусев Артем
JavaScript-developer «БАРС Групп»



Как я пришел к этому?



Начну с того, что о некоторых особенностях стилизации я узнал в ходе работы.



После чего мне стало интересно, есть ли еще что-то подобное, о чем я не знал и что действительно поможет мне.



Поэтому я решил погрузиться в эту тему с головой и начал искать различную информацию.

О каких элементах пойдет речь?

`:host`

`:host()`

`::slotted()`

`:defined`

`:host-context()`

`::part()`

`::theme()`

Shadow DOM



Shadow DOM — часть документа, реализующая инкапсуляцию в DOM дереве.

Благодаря ему в компоненте есть собственное «тенивое» DOM-дерево, к которому нельзя просто так обратиться из главного документа, у него могут быть изолированные CSS-правила и т.д.

```
<todo-view>
  #shadow-root (open)
    <style>_</style>
    <div>
      <paper-input-container dir="null">_</paper-input-container>
    </div>
    <div>
      <todo-data></todo-data>
      <h2>Uncompleted to do tasks</h2>
      <todo-list filter-by="uncompleted">_</todo-list>
      <h2>Completed to do tasks</h2>
      <todo-list filter-by="completed">_</todo-list>
    </div>
  </todo-view>
```



Задача №1

Создать компонент nf-panel,
который содержал бы в себе 3 раздела:

01

Шапку

02

Контентную часть

03

Футер



Также добавить возможность стилизации компонента
внутри его собственного Shadow DOM

Псевдокласс :host

Псевдокласс :host

позволяет указать на сам
компонент внутри
собственного Shadow DOM



Пример работы :host

index.html

```
<nf-panel>  
  <div class="icon" slot="tools"></div>  
  <p slot="content">MoscowCSS Meetup №17</p>  
</nf-panel>
```

<nf-panel>

```
<style>  
  :host{  
    min-width: 50vw;  
    height: fit-content;  
    display: flex;  
    flex-direction: column;  
    border: 1px solid var(--border-color);  
    border-radius: 4px;  
  }  
</style>
```

MoscowCSS



MoscowCSS Meetup №17

Доклад: "Особенности CSS в веб-компонентах"



Поддержка браузерами

CSS selector: :host

Usage % of all users ?
Global 86.74%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	C	
		2-60													
		61-62	4-53	3.1-9.1	10-40	3.2-9.3							4-5.4		
6-10	12-18	63-71	54-78	10-12.1	41-63	10-13.1		2.1-4.4.4	12-12.1				6.2-9.2		
11	79	72	79	13	64	13.2	all	76	46	79	68	12.12	10.1		
		73-74	80-82	TP		13.3									

Задача №2

Создать компонент offer-block

который обладал бы
возможностью изменения
цвета кнопки, исходя
из указанного значения
в атрибуте



Псевдокласс :host()

Функция псевдокласса :host()

позволяет проверить компонент, который
удовлетворяет селектору, переданному
в качестве параметра

("*")

+

.

x

{

.

/

Пример работы :host()

index.html

```
<offer-block color="green"></offer-block>  
<offer-block color="red"></offer-block>
```

<offer-block>

```
<style>  
:host([color='red']) button{  
  background: var(--red-color);  
}  
:host([color='green']) button{  
  background: var(--green-color);  
}  
</style>
```

WOULD YOU LIKE TO BUY A BOOK?

Now there is an action on books. Discounts up to 50%

BUY

WOULD YOU LIKE TO BUY A BOOK?

Now there is an action on books. Discounts up to 50%

BUY



Поддержка браузерами

CSS selector: :host()


Usage % of all users ?
Global 86.74%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	C	
		2-60													
		¹ 61-62	4-53	3.1-9.1	10-40	3.2-9.3							4-5.4		
6-10	12-18	63-71	54-78	² 10-12.1	41-63	² 10-13.1		2.1-4.4.4	12-12.1				6.2-9.2		
11	79	72	79	² 13	64	² 13.2	all	76	46	79	68	12.12	10.1		
		73-74	80-82	² TP		² 13.3									

Задача №3

Покрасить иконку календаря
компонента **nf-datetime**
в красный цвет, если
значение **невалидно**

Дата регистрации

```
...  
<nf-datetime>  
  <nf-icon-button icon="nf-icons-16:calendar" slot="suffix"></nf-icon-button>  
  <nf-input-mask></nf-input-mask>  
</nf-datetime>  
....
```

<slot>

Title

h1#title-inside-web-component 1264 x 37

ipisicing elit. Dicta est nesciunt provident quas, saepe voluptatibus!

The screenshot displays the browser's developer tools with the 'Elements' panel open. The DOM tree shows a custom element containing a shadow root. Inside the shadow root, there is a slot containing an h1 element with the text 'reveal' and a p element with the text 'reveal'. The h1 element is selected, and its styles are shown in the 'Styles' panel. The styles include default user agent styles for h1, such as display: block, font-size: 2em, and font-weight: bold. The text 'user agent styles' is visible at the end of the style list.

```
<!doctype html>
<html lang="en">
  <head>...</head>
  <body>
    <custom-element>
      <#shadow-root (open)>
        <slot>
          <h1> reveal
          <p> reveal
        </slot>
        <h1 id="title-inside-web-component">Title</h1>
        <p id="descr-inside-web-component">...</p>
      </custom-element>
```

Styles

element.style { }

h1 {

- display: block;
- font-size: 2em;
- margin-block-start: 0.67em;
- margin-block-end: 0.67em;
- margin-inline-start: 0px;
- margin-inline-end: 0px;
- font-weight: bold;

user agent styles

Псевдоэлемент `::slotted()`

Чтобы стилизовать содержимое в слотах,
используйте синтаксис

`::slotted()`.

Примечание:

Для работы в пределах ограничений с ShadyCSS, и обеспечения согласованного поведения в различных браузерах, добавьте селектор в левой части от записи `::slotted classname`



Пример работы ::slotted()

index.html

```
<nf-datetime invalid>  
  <nf-icon-button slot="suffix">  
  </nf-iconbutton>  
</nf-datetime>
```



<nf-datetime>

```
<style>  
  :host([invalid]) ::slotted(nf-icon-button){  
    fill: var(--invalid-color);  
  }  
</style>  
<input required disabled placeholder="___-___-___">  
  <slot name="suffix"></slot>  
</input>
```



Поддержка браузерами

CSS selector: ::slotted

Usage % of all users ?
Global 86.84%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	C	
		2-58													
		59-62	4-49	3.1-9.1	10-36	3.2-9.3							4		
6-10	12-18	63-71	50-78	10-12.1	37-63	10-13.1		2.1-4.4.4	12-12.1				5-9.2		
11	79	72	79	13	64	13.2	all	76	46	79	68	12.12	10.1		
		73-74	80-82	TP		13.3									

Задача №4

Добавить возможность
отображения прелоадера
во время загрузки
composition-component



Псевдокласс :defined

Псевдокласс :defined

позволяет проверить элемент,
который был определён,
включая любой стандартный
встроенный в браузер элемент,
а также пользовательские
элементы

("*")

.

+

.

x

{

.

/

Пример работы :defined

index.html

```
<style>
...
  composition-component:not(:defined){
    content: url("img/icon/loader.svg");
  }
</style>
<composition-component title="The School of Athens" description="Raphael (1509–1511)"
image="img/art/art2.jpg"></composition-component>
```

<composition-component>

```
setTimeout(() => customElements.define('composition-component',
CompositionComponent), 2000);
```



Work with :defined pseudo-class



The School of Athens

Raphael (1509–1511)

Поддержка браузерами

CSS selector: :defined

Usage % of all users ?
Global 86.74%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?									
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	C
		2-62	4-53	3.1-9.1	10-40	3.2-9.3							4-5,4	
6-10	12-18	63-71	54-78	10-12.1	41-63	10-13.1		2.1-4.4.4	12-12.1				6.2-9.2	
11	79	72	79	13	64	13.2	all	76	46	79	68	12.12	10.1	
		73-74	80-82	TP		13.3								

Задача №5

Создать компонент context-button

который бы обладал возможностью
изменения цвета background'a
и border'a исходя из того,
где он находится



Псевдокласс :host-context()



("*")

Псевдокласс :host-context()

позволяет узнать контекст,
в котором использован
веб-компонент

+

.

x

{

.

/

Пример работы :host-context()

index.html

```
<header>  
  <context-button>  
  </context-button>  
</header>
```

<context-button>

```
<style>  
  :host-context(header){  
    border: 1px solid var(--light-gray-color);  
    background: var(--white-color);  
  }  
</style>
```



Поддержка браузерами

CSS selector: :host-context()

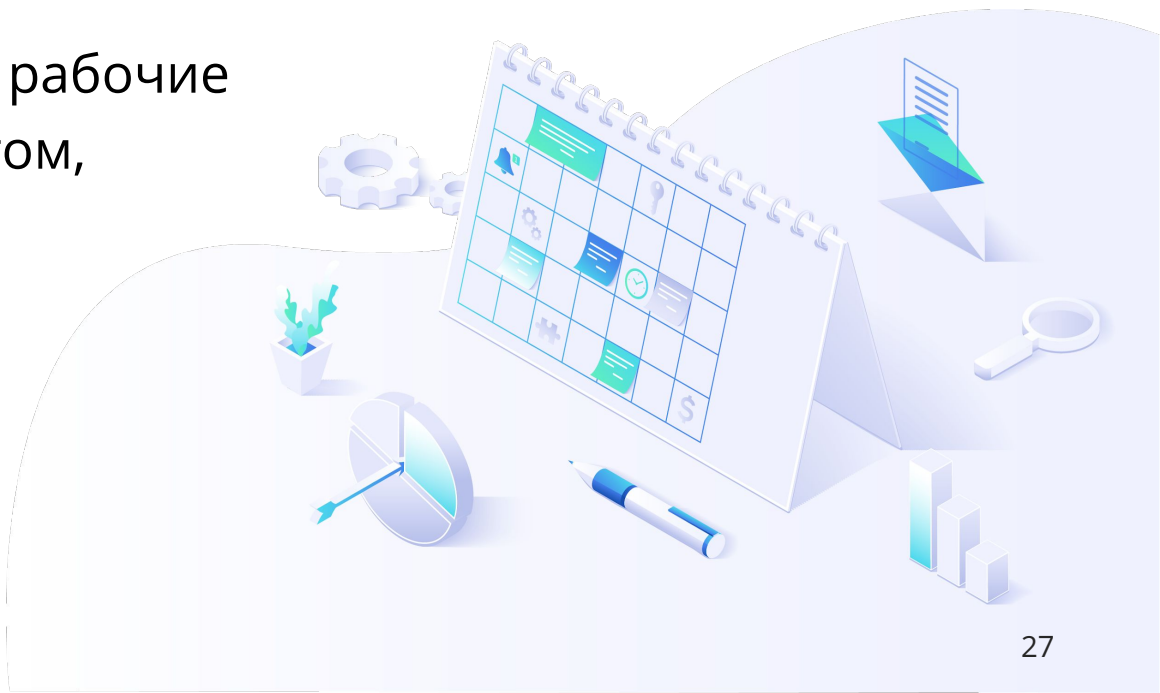
Usage % of all users ?
Global 67.11%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?									
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	C
			4-53		10-40								4-5.4	
6-10	12-18	2-71	54-78	3.1-12.1	41-63	3.2-13.1		2.1-4.4.4	12-12.1				6.2-9.2	
11	79	72	79	13	64	13.2	all	76	46	79	68	12.12	10.1	
		73-74	80-82	TP		13.3								

Задача №6

Создать компонент week-list

который бы отображал рабочие
дни недели одним цветом,
а выходные - другим



Псевдоэлемент ::part()



("*")

Псевдоэлемент ::part()

позволяет выбрать внутренние
элементы указанного компонента,
у которых наименование атрибута
part удовлетворяет значению
селектора

+

.

x

{

.

/

Пример работы ::part()

index.html

```
<style>
  week-list::part(holiday){
    color: var(--red-color);
  }
</style>
<week-list></week-list>
```

<week-list>

```
<ul>
  <li part="holiday">Sunday</li>
  <li part="work">Monday</li>
  <li part="work">Tuesday</li>
  <li part="work">Wednesday</li>
  <li part="work">Thursday</li>
  <li part="work">Friday</li>
  <li part="holiday">Saturday</li>
</ul>
```

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday



Поддержка браузерами

CSS selector: `::part`

Usage % of all users ?
Global 62.24%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet		
		2-68	4-72		10-58										
6-10	12-18	69-71	73-78	3.1-12.1	60-63	3.2-13.1		2.1-4.4.4	12-12.1				4-9.2		
11	79	72	79	13	64	13.2	all	76	46	79	68	12.12	10.1		
		73-74	80-82	TP		13.3									



01

::part не позволяет использовать после себя структурные псевдоклассы

02

Нельзя использовать сложные селекторы внутри функции **::part**, поэтому вы не можете использовать его для стилизации дочернего узла

03

Нельзя использовать классы и селекторы атрибутов после **::part**

Задача №7

Создать компонент

который подстраивал бы
все вложенные в него
дочерние элементы под
одну тему. Независимо
от того, каким уровнем
вложенности они
обладают



Псевдоэлемент ::theme()

Псевдоэлемент ::theme

позволяет находить
указанный элемент,
независимо от того, насколько
глубоко он вложен в теневых
деревьях



("*")



Here is <a-form>. It's only styled using ::theme

Theme: ☒ Blue ☐ Pink ☐ Green

Basic info

First name

Last name

This switch exposes no parts, but uses custom properties. Use **left** or **right** arrows to switch with the keyboard.

Yes

No

Credit card information

CC number

Expiration date

dd

mm

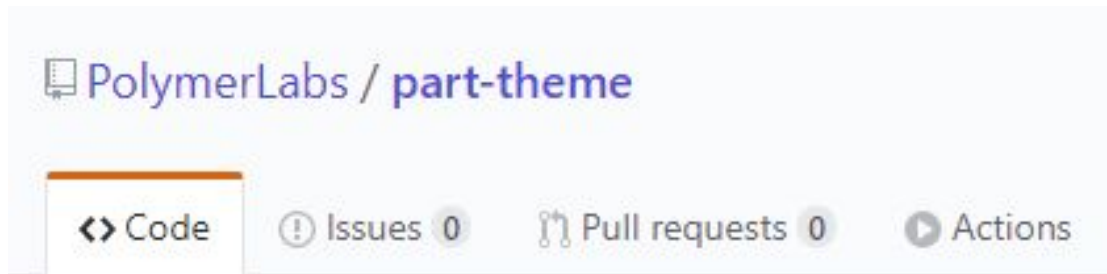
yy

```

1 import {Froyo} from './froyo.js';
2 import {PrettyInput} from './pretty-input.js';
3 import {BasicInfo} from './basic-info.js';
4 import {CCInfo} from './cc-info.js';
5
6 export class AForm extends Froyo(HTMLElement) {
7   get template() {
8     return `
9       <style>
10         :host {
11           display: block;
12         }
13
14         basic-info, cc-info {
15           border: 1px solid var(--text-color);
16         }
17         :host::theme(input)::placeholder {
18           color: var(--placeholder-color);
19         }
20         :host::theme(wrapper) {
21           border-bottom: 1px solid var(--underline-color);
22         }
23         :host::theme(input) {
24           color: var(--text-color);
25         }
26         :host::theme(label) {
27           color: var(--text-color);
28           font-weight: bold;
29         }
30         :host::theme(button):hover,
31         :host::theme(button):focus {
32           color: white;
33           background: var(--text-color);
34         }
35       </style>
36       <basic-info class="basic"></basic-info>
37       <cc-info class="cc" ></cc-info>
38     `;
39   };
40 }
41
42 customElements.define('a-form', AForm);

```

Модуль, позволяющий использовать `::theme`



Experimental proof of concept shim for `::part::theme` css.

Почему не стоит использовать данный модуль в продакшене?

`::theme()` - дорогой в плане производительности

Нет гарантии того, что этот модуль будет работать во всех случаях, описанных в старом черновике



Противоречит одной из концепций Shadow DOM



Что из всего этого не стоит использовать на проде?

`::part()`

`::theme()`

`:host-context()`

Полезные ссылки

Статья [::part and ::theme, an ::explainer](#)

Коллекция работ на [CodePen](#)

Статья [Encapsulating Style and Structure with Shadow DOM](#)

Репозиторий [PolymerLabs/part-theme](#)

Статья [CSS Shadow Parts are coming!](#)

Спасибо за внимание!



CreativeRusBear



@art_gusev

