



CSS

Information Technology



CSS — Cascading Styles Sheet

Каскадные таблицы стилей (CSS)

используется для задания внешнего вида элементов **HTML** или других текстовых форматов. Наиболее удобный способ записи стилей заключается в использовании отдельного файла с расширением «.css», но можно их включать непосредственно в **HTML** код.

Хокон Виум Ли

Учёный, специалист в области информатики, предложивший каскадные таблицы стилей (CSS). В англоязычных текстах его часто называют *howcome*, а норвежское имя считается трудночитаемым.

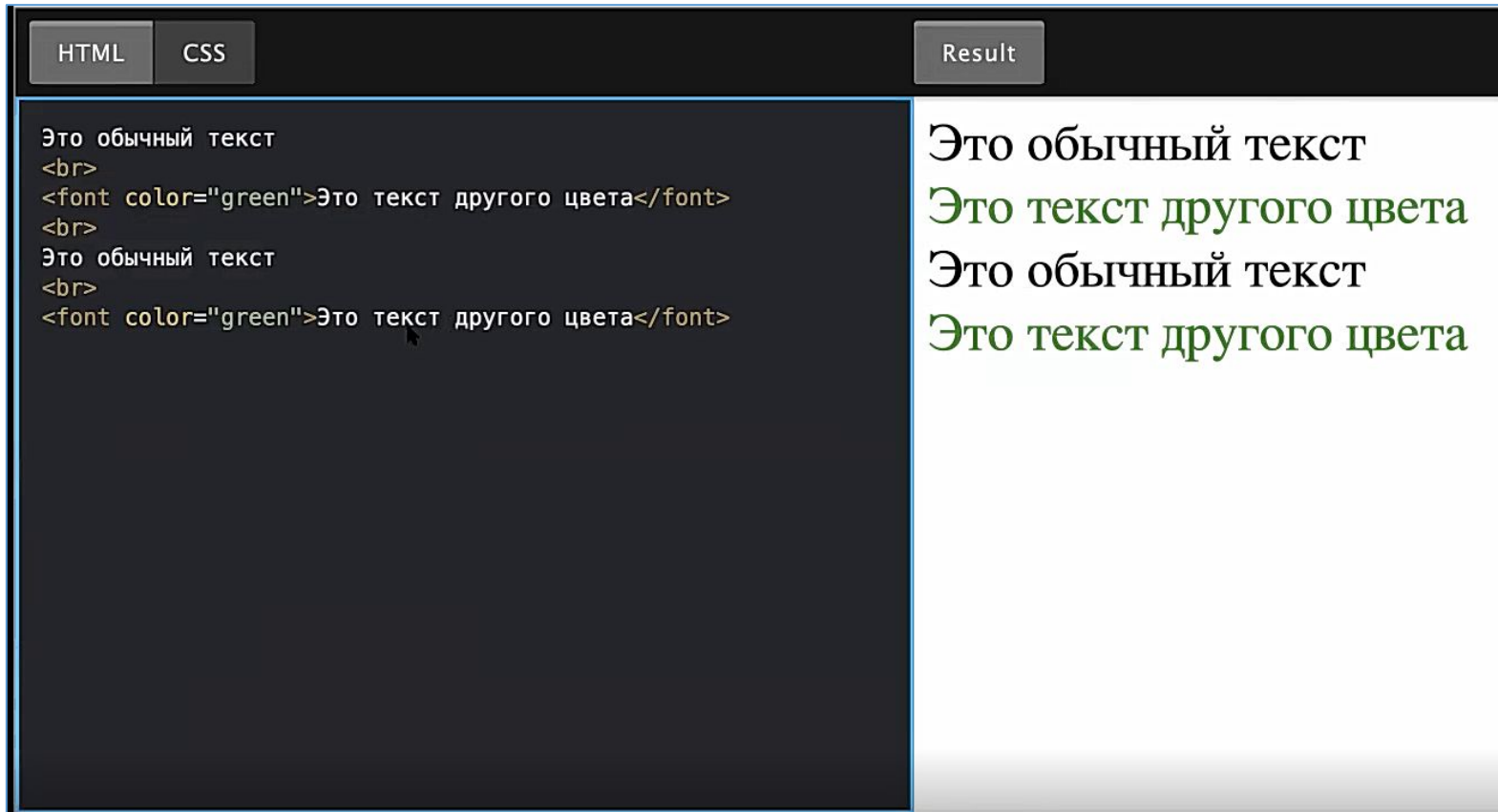
В 1994 году предложил концепцию каскадных таблиц стилей.

В 2006 году в Университете Осло защитил диссертацию по теме «Каскадные таблицы стилей».



<https://web.archive.org/web/20080906003804/http://people.opera.com/howcome/>

ПРИМЕР СТИЛЕЙ



The screenshot shows a web editor interface with three tabs: 'HTML', 'CSS', and 'Result'. The 'HTML' tab is active, displaying the following code:

```
Это обычный текст  
<br>  
<font color="green">Это текст другого цвета</font>  
<br>  
Это обычный текст  
<br>  
<font color="green">Это текст другого цвета</font>
```

The 'Result' tab shows the rendered output of the code:

Это обычный текст
Это текст другого цвета
Это обычный текст
Это текст другого цвета

Тег `` `` через атрибут `color` задает зеленый цвет.

ПРИМЕР СТИЛЕЙ

```
HTML
1 Это обычный текст
2 <br>
3 <font color="red">Это текст другого цвета</font>
4 <br>
5 Это обычный текст
6 <br>
7 <font color="red">Это текст другого цвета</font>

CSS
1 body {
2   font-size: 30px;
3 }
4
```

Это обычный текст
Это текст другого цвета
Это обычный текст
Это текст другого цвета

Тег ` ` через атрибут `color` задает красный цвет. Установка размера текста (`font-size`) с помощью пикселей (30 px) дает вам полный контроль над размером текста. Изменять цвет необходимо несколько раз.

ПРИМЕР СТИЛЕЙ

```
HTML
1 Это обычный текст
2 <br>
3 <span class="primary">Это текст другого цвета</span>
4 <br>
5 Это обычный текст
6 <br>
7 <span class="primary">Это текст другого цвета</span>
8
9 <style>
10 .primary {
11   color: red;
12 }
13 </style>

CSS
1 body {
2   font-size: 30px;
3 }
```

Это обычный текст

Это текст другого цвета

Это обычный текст

Это текст другого цвета

Тег `` предназначен для определения строчных элементов документа. HTML class позволяет присвоить класс элементу «primary» (его имя). Изменения цвета происходит в одном месте. CSS font-size устанавливает размер шрифта.

ПРИМЕР СТИЛЯ CSS

```
HTML
1 <!--
2 <head>
3   <link rel="stylesheet" href="styles.css">
4 </head>
5 →
6
7 Это обычный текст
8 <br>
9 <span class="primary">Это текст другого цвета</span>
10 <br>
11 Это обычный текст
12 <br>
13 <span class="primary">Это текст другого цвета</span>

CSS
1 .primary {
2   color: red;
3 }
4
5
6
7
8
9
10
11
12
13
14
15
```

Это обычный текст
Это текст другого цвета
Это обычный текст
Это текст другого цвета

CSS color: red позволяет устанавливать красный цвет для всех страниц где есть класс элемента «primary».



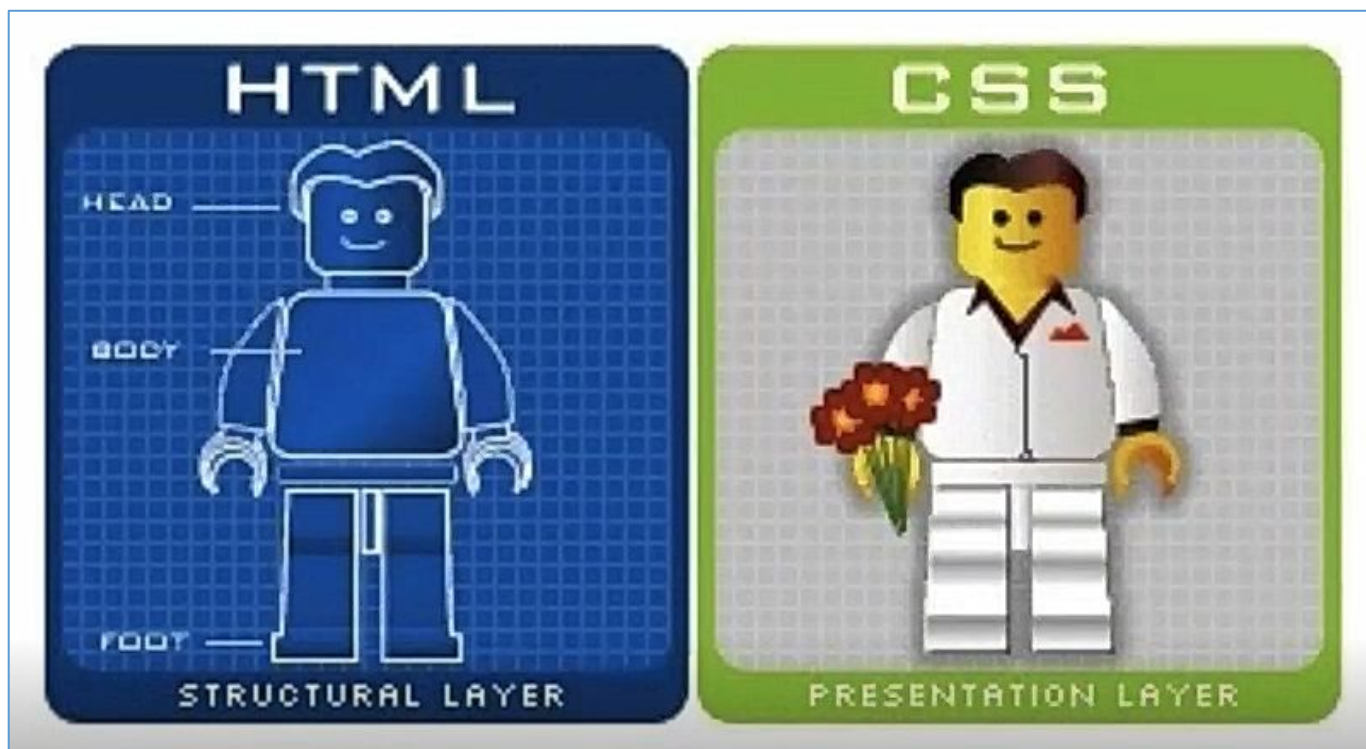
СТИЛЬ

```
1  ∨  h1{↵
2     |····color:·■blue;↵
3     }↵
```

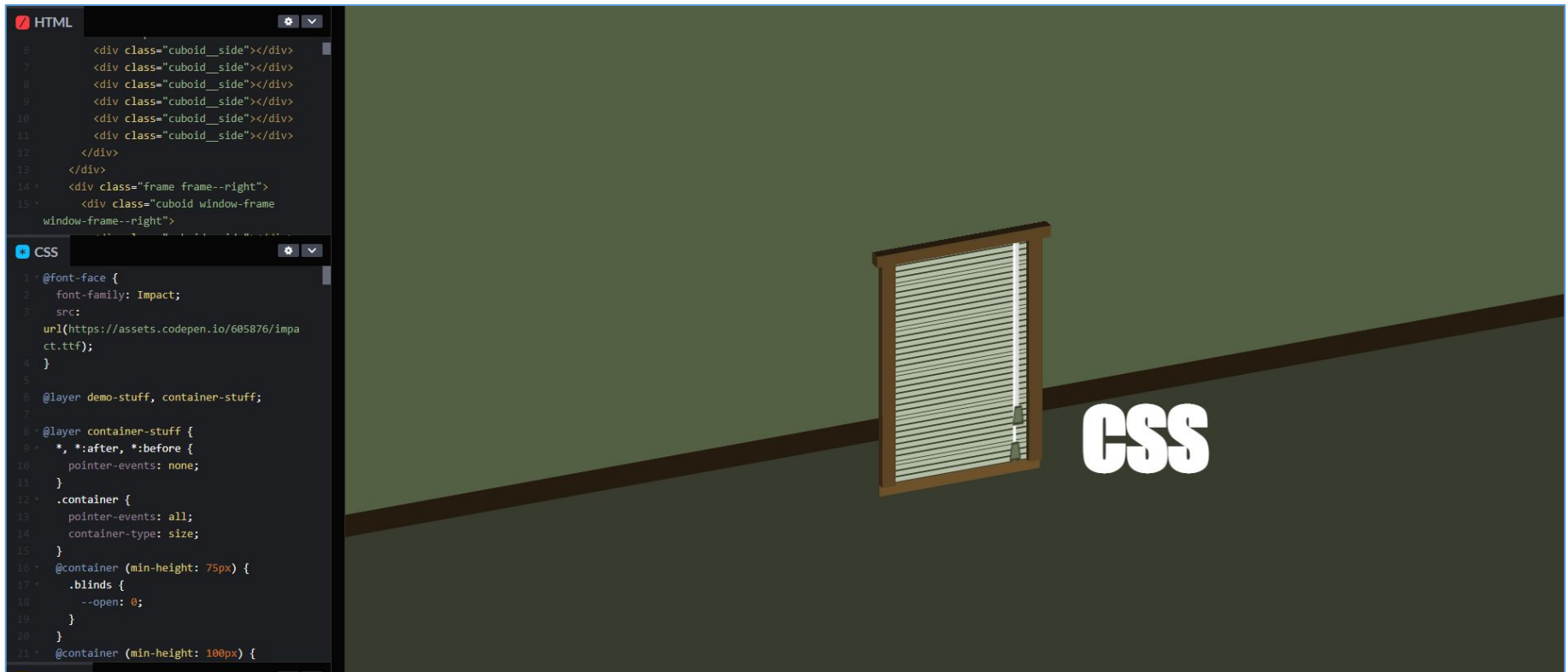
В этой записи **h1** описывает **HTML** тег, для которого задается стиль, а в фигурных скобках приводится описание самого стиля.

Описание стиля состоит из нескольких записей, которые заканчиваются точкой с запятой. Одна запись описывает некоторое свойства стиля, в примере задается свойство цвета текста (blue - синий).

РАЗДЕЛЕНИЕ СТРУКТУРЫ И ВНЕШНЕГО ВИДА



ПРИМЕРЫ CSS



The image displays a code editor interface with two panels: HTML and CSS. The HTML panel shows the structure of a window with blinds, using classes like 'cuboid_side', 'frame frame--right', and 'cuboid window-frame window-frame--right'. The CSS panel shows the styling, including a custom font 'Impact' and various container and blinds styles. To the right of the code editor is a 3D rendering of a window with horizontal blinds, set against a dark green background with a diagonal line. The word 'CSS' is written in large white letters next to the window.

```
HTML
6 <div class="cuboid_side"></div>
7 <div class="cuboid_side"></div>
8 <div class="cuboid_side"></div>
9 <div class="cuboid_side"></div>
10 <div class="cuboid_side"></div>
11 <div class="cuboid_side"></div>
12 </div>
13 </div>
14 <div class="frame frame--right">
15 <div class="cuboid window-frame
  window-frame--right">

CSS
1 @font-face {
2   font-family: Impact;
3   src:
4     url(https://assets.codepen.io/605876/impact.woff2);
5 }
6 @layer demo-stuff, container-stuff;
7
8 @layer container-stuff {
9   *, *:after, *:before {
10    pointer-events: none;
11  }
12  .container {
13    pointer-events: all;
14    container-type: size;
15  }
16  @container (min-height: 75px) {
17    .blinds {
18      --open: 0;
19    }
20  }
21  @container (min-height: 100px) {
```

ПРИМЕРЫ CSS

HTML

```
1 <h1>Perpetual bounce CSS.</h1>
2 <input type="radio" name="speedControl"
3 id="fast">
4 <input type="radio" name="speedControl"
5 id="default" checked>
6 <input type="radio" name="speedControl"
7 id="slow">
```

CSS

```
1 html,body{
2 height: 100%;
3 margin:0;
4 font-family: sans-serif;
5 background-image:linear-gradient(135deg,
6 white, lightgray );
7 overflow: hidden;
8 }
9 h1{
10 position: absolute;
11 font-weight: normal;
12 margin:0;
```

JS

Perpetual bounce CSS.



Default

ПРИМЕРЫ CSS

```
HTML
1 <span class="jump"> 🎃 </span>
2 <span class="jump alt"> 🎃 </span>
3 <span class="jump"> 🎃 </span>
4 <span class="jump alt"> 🎃 </span>
5 <span class="jump"> 🎃 </span>
6 <br>
7 <span class="jump alt"> 🎃 </span>
8 <span class="jump"> 🎃 </span>
9 <span class="jump alt"> 🎃 </span>
10 <span class="jump"> 🎃 </span>
11 <span class="jump alt"> 🎃 </span>

CSS
1 body{width:100%;text-align:center;margin-
2 top:1em;}
3 *{font-size:2em;}
4 /*
5 https://gist.github.com/davidsharp/a2cc3c
6 8988dbfb0e5d291d1d084396cf */
7 .jump {
8   display: inline-block;
9   animation-duration: 1.5s;
10  animation-name: jump;
11  animation-iteration-count: infinite;
12 }
13 .jump.alt {
14   animation-delay: .75s;
15 }
16 @keyframes jump {
17   0%, 100% {
18     transform: scale(1.1,1)
19     translateY(0);
20   }
21 }
```

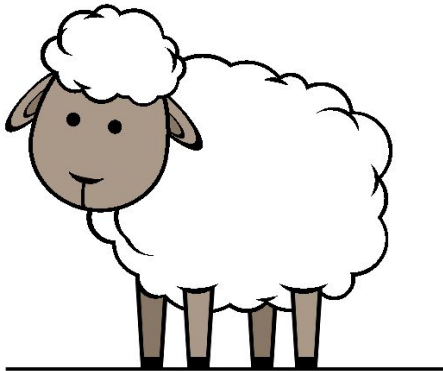


ПРИМЕРЫ CSS

```
HTML
1 <article role="img" aria-label="cartoon of
  a sheep"></article>

CSS
1 article {
2   --skin: #a98;
3   --skin-dark: #876;
4   width: 60vmin;
5   aspect-ratio: 1.2;
6   position: absolute;
7   top: 50%;
8   left: 50%;
9   transform: translate(-50%, -50%);
10  background:
11    /* details 1 */
12    radial-gradient(130% 140% at 74.5%
13    48%, #fff 7.5%, #0000 0),
14    radial-gradient(110% 100% at 77.5%
15    48%, #000 7%, #0000 0),
16    radial-gradient(circle at 74% 37%,
17    #fff 7.5%, #0000 0),
18    radial-gradient(90% 100% at 74.8%
19    34.5%, #000 6.5%, #0000 0),
20    radial-gradient(circle at 32% 58%,
21    #fff 7.5%, #0000 0),
22    radial-gradient(circle at 34% 61%,
23    #fff 7.5%, #0000 0),
24  }

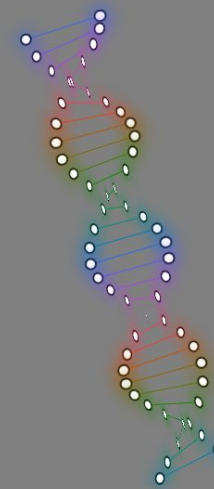
JS
```



ПРИМЕРЫ CSS

```
HTML
1 <div class="items">
2   <div class="item"></div>
3   <div class="item"></div>
4   <div class="item"></div>
5   <div class="item"></div>
6   <div class="item"></div>
7   <div class="item"></div>
8   <div class="item"></div>
9   <div class="item"></div>
10  <div class="item"></div>
11 </div>
12

CSS (SCSS)
1 ,
2
3
4
5 html {
6   display: flex;
7   justify-content: center;
8   align-items: center;
9   min-width: 100vw;
10  min-height: 100vh;
11 }
12
```



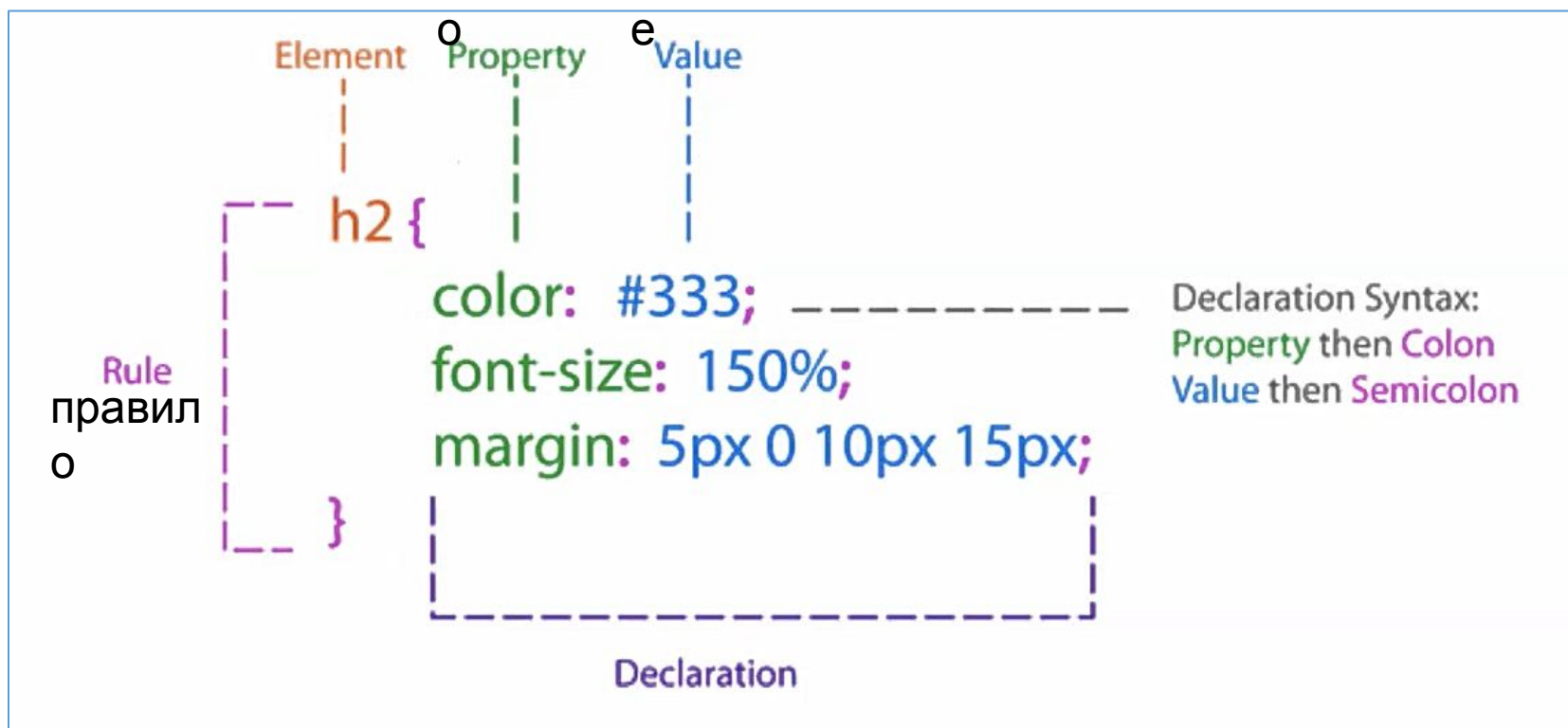


ПРЕИМУЩЕСТВА CSS

- Простота последующего изменения дизайна, изменяя css файл.
- Несколько дизайнов страницы с одним и тем же HTML-кодом (светлая&темная тема страницы).
- Уменьшение времени загрузки страниц сайта.
 - Уменьшение загрузки сайта, так как кода становится меньше.
 - **Кэширование** CSS-файлов в браузере.
- Дополнительные возможности стилизации страниц которых не было в HTML.

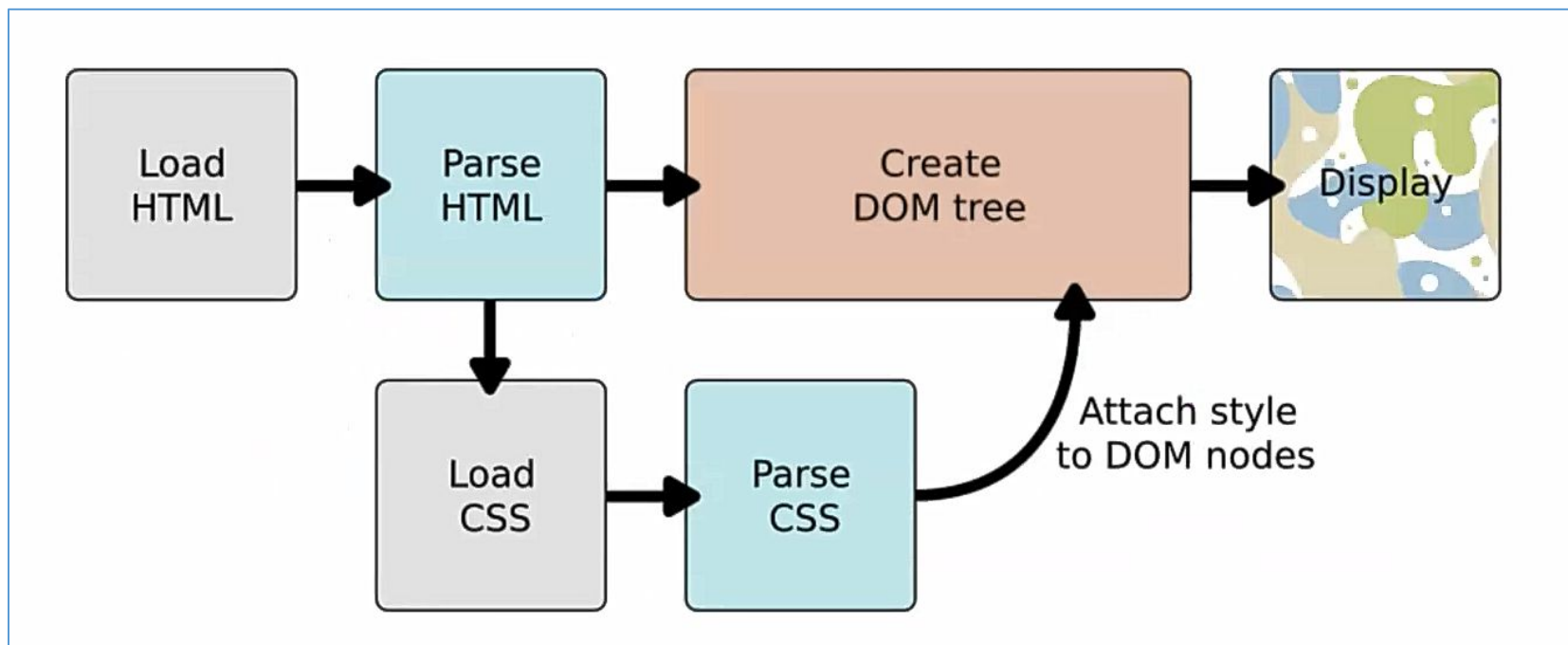
СИНТАКСИС CSS-ПРАВИЛА

селектор свойств значения



описани
е

СХЕМА РАБОТЫ БРАУЗЕРА



CSS COM дерево накладывается на DOM дерево=результатирующее дерево.



ТИПЫ СЕЛЕКТОРОВ

- Базовые селекторы.
- Комбинаторы.
- Псевдоклассы.
- Псевдоэлементы.

БАЗОВЫЕ СЕЛЕКТОРЫ

The screenshot shows a web browser's developer tools interface. On the left, the 'CSS' tab is active, displaying four CSS rules:

```
/* 1. По тегу: */
a {
  color: green;
}

/* 2. По классу */
.text {
  border-bottom: 3px dotted #00f;
}
.text.secondary {
  font-size: 20px;
}

/* 3. По идентификатору */
#footer {
  background-color: #000;
  color: #fff;
}

/* 4. По атрибуту: */
[target="_blank"] {
  text-decoration: none;
}
```

On the right, the 'Result' tab shows the visual output of these rules:

- Базовые селекторы**
- По тегу**: The text 'По тегу' is displayed in green.
- По классу 1**: The text 'По классу 1' is displayed with a blue dotted underline.
- По нескольким классам**: The text 'По нескольким классам' is displayed with a blue dotted underline.
- По идентификатору**: The text 'По идентификатору' is displayed in white on a black background.
- По атрибуту**: The text 'По атрибуту' is displayed in green.
- Универсальный**: The text 'Универсальный' is displayed in black.

At the bottom of the developer tools, there are buttons for 'Resources', '1x', '0.5x', and '0.25x'.

Базовых селекторов существует 5 видов:

1. Color green применяется ко всем тегам `<a> `.
2. Стиль по классу: для всех элементов у которых есть `.text` сделаем границу `3px dotted` (точечную) `#00f` (синего цвета).

БАЗОВЫЕ СЕЛЕКТОРЫ

```
<h1>Базовые селекторы</h1>
<a href="">По тегу</a>
<p class="text">По классу 1</p>
<p class="text secondary">По нескольким классам</p>
<footer id="footer">По идентификатору</footer>
<a href="#" target="_blank">По атрибуту</a>
<br>
<span>Универсальный</span>
```

Базовые селекторы

По тегу

По классу 1

По нескольким классам

По идентификатору

По атрибуту

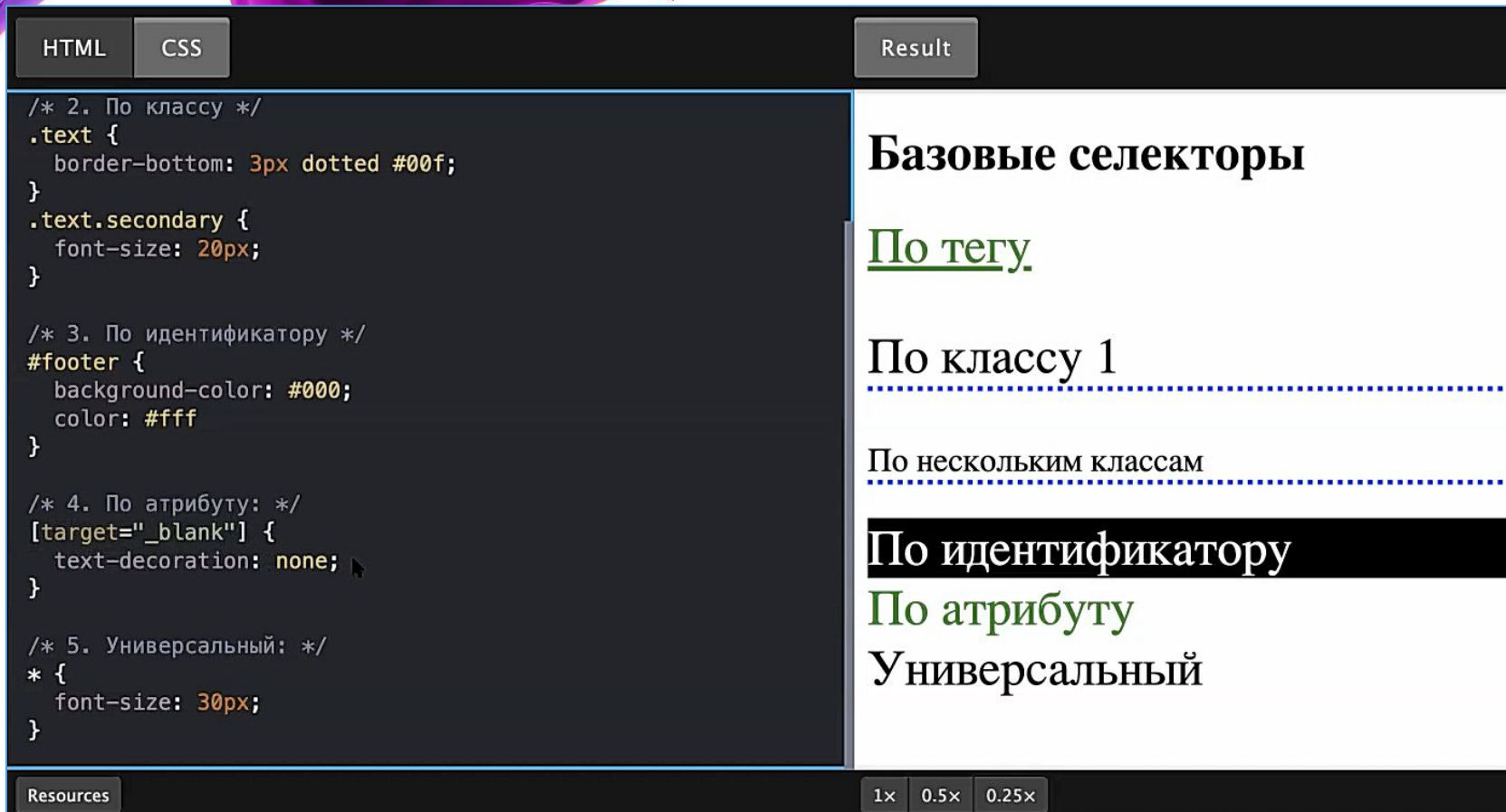
Универсальный

class – это атрибут, а “text” – значение.

Если несколько class для одного элемента может быть несколько, разделяются они пробелом “text secondary”.

ID – идентификатор. “footer” – имя идентификатора может быть только одно. “footer” имя, которое мы придумали.

БАЗОВЫЕ СЕЛЕКТОРЫ



The screenshot shows a CSS editor interface with a dark theme. On the left, there are tabs for 'HTML' and 'CSS', and a 'Result' tab on the right. The CSS code is as follows:

```
/* 2. По классу */  
.text {  
  border-bottom: 3px dotted #00f;  
}  
.text.secondary {  
  font-size: 20px;  
}  
  
/* 3. По идентификатору */  
#footer {  
  background-color: #000;  
  color: #fff  
}  
  
/* 4. По атрибуту: */  
[target="_blank"] {  
  text-decoration: none;  
}  
  
/* 5. Универсальный: */  
* {  
  font-size: 30px;  
}
```

The 'Result' tab displays the following visual output:

- Базовые селекторы**
- По тегу.
- По классу 1
- По нескольким классам
- По идентификатору**
- По атрибуту
- Универсальный

At the bottom of the editor, there are 'Resources' and zoom controls (1x, 0.5x, 0.25x).

2. Для одного элемента можно задавать несколько class. Селектор `.text.secondary` содержит `font-size`(размер шрифта) `20px`.
3. Кроме класса можно задать любому элементу `id` (идентификатор). ID у элемента может быть только один. Для `#footer` стили `background-color` (цвет фона элемента) `#000` (black), `#fff` (white).

БАЗОВЫЕ СЕЛЕКТОРЫ

The screenshot shows a web browser's developer tools interface. On the left, the 'CSS' tab is active, displaying the following CSS code:

```
/* 2. По классу */
.text {
  border-bottom: 3px dotted #00f;
}
.text.secondary {
  font-size: 20px;
}

/* 3. По идентификатору */
#footer {
  background-color: #000;
  color: #fff
}

/* 4. По атрибуту: */
[target="_blank"] {
  text-decoration: none;
}

/* 5. Универсальный: */
* {
  font-size: 30px;
}
```

On the right, the 'Result' tab shows the visual output of the CSS. The text 'Базовые селекторы' is displayed in a large, bold, black font. Below it, the text 'По тегу' is underlined in green. The text 'По классу 1' is followed by a blue dotted line. The text 'По нескольким классам' is followed by a blue dotted line. The text 'По идентификатору' is displayed in white on a black background. The text 'По атрибуту' is displayed in green. The text 'Универсальный' is displayed in a large, bold, black font. At the bottom of the developer tools, there are tabs for 'Resources' and zoom levels '1x', '0.5x', and '0.25x'.

4. Можно задавать стили в зависимости от атрибутов элементов. Атрибуты записываются в квадратных скобках. «text-decoration: none», т.е убрать подчеркивание у которых есть атрибут target="_blank".
5. Универсальный селектор. * применится ко всем тегам.

БАЗОВЫЕ СЕЛЕКТОРЫ

The screenshot shows a web browser interface with two main panels: 'HTML' and 'Result'. The 'HTML' panel contains the following code:

```
<h1>Базовые селекторы</h1>
<a href="">По тегу</a>
<p class="text">По классу 1</p>
<p class="text secondary">По нескольким классам</p>
<footer id="footer">По идентификатору</footer>
<a href="#" target="_blank">По атрибуту</a>
<br>
<span>Универсальный</span>
```

The 'Result' panel displays the rendered output:

- Базовые селекторы** (Title)
- По тегу (Text with green underline)
- По классу 1 (Text with blue dotted underline)
- По нескольким классам (Text with blue dotted underline)
- По идентификатору** (Text with black background)
- По атрибуту (Text with green color)
- Универсальный (Text with normal styling)

At the bottom of the browser window, there are tabs for 'Resources' and zoom controls set to '1x'.

target="_blank", т. е ссылка должна открываться в новой вкладке.

КОМБИНАТОРЫ

The screenshot shows a web browser interface with a dark theme. On the left, there is a code editor with tabs for 'HTML' and 'CSS'. The CSS code includes: `body {font-size: 30px;}`, `/* 1. дочерний элемент A B */ .content p { color: red; }`, `/* 2. прямой потомок A > B */ .content > p { border: 1px solid green; }`, `/* 3. соседний элемент A + B */ .content + p { color: blue; }`, and `/* 4. следующие элементы A ~ B */ li ~ li { list-style-type: circle; }`. On the right, the 'Result' tab shows a white page with the title 'Комбинаторы'. Below the title, there are four sections: 'Дочерний селектор' (red text), 'Не прямой потомок' (red text), 'Соседний элемент' (blue text), and a list with 'Первый элемент' (bullet point) and 'Второй элемент' (radio button). The browser's address bar shows 'Resources' and zoom levels '1x', '0.5x', and '0.25x'.

1. Дочерний элемент часто используется. Применить цвет красный (color: red) ко всем элементам «p», которые находятся внутри элементов с классом «.content».
2. Применять к непосредственным детям «.content > p».
3. Применять к соседним элементам «.content + p».
4. Применять к следующим элементам «li ~ li».

КОМБИНАТОРЫ

The screenshot shows a web browser interface with two panes. The left pane is labeled 'HTML' and contains the following code:

```
<h1>Комбинаторы</h1>
<div class="content">
  <p>Дочерний селектор</p>
  <div>
    <p>Не прямой потомок</p>
  </div>
</div>
<p>Соседний элемент</p>
<ul>
  <li>Первый элемент</li>
  <li>Второй элемент</li>
  <li>Третий элемент</li>
</ul>
```

The right pane is labeled 'Result' and shows the rendered output. The title 'Комбинаторы' is in large black font. Below it, three sections are shown:

- Дочерний селектор** (Child selector): The text 'Дочерний селектор' is highlighted in red.
- Не прямой потомок** (Not a direct descendant): The text 'Не прямой потомок' is highlighted in red.
- Соседний элемент** (Adjacent element): The text 'Соседний элемент' is highlighted in blue.

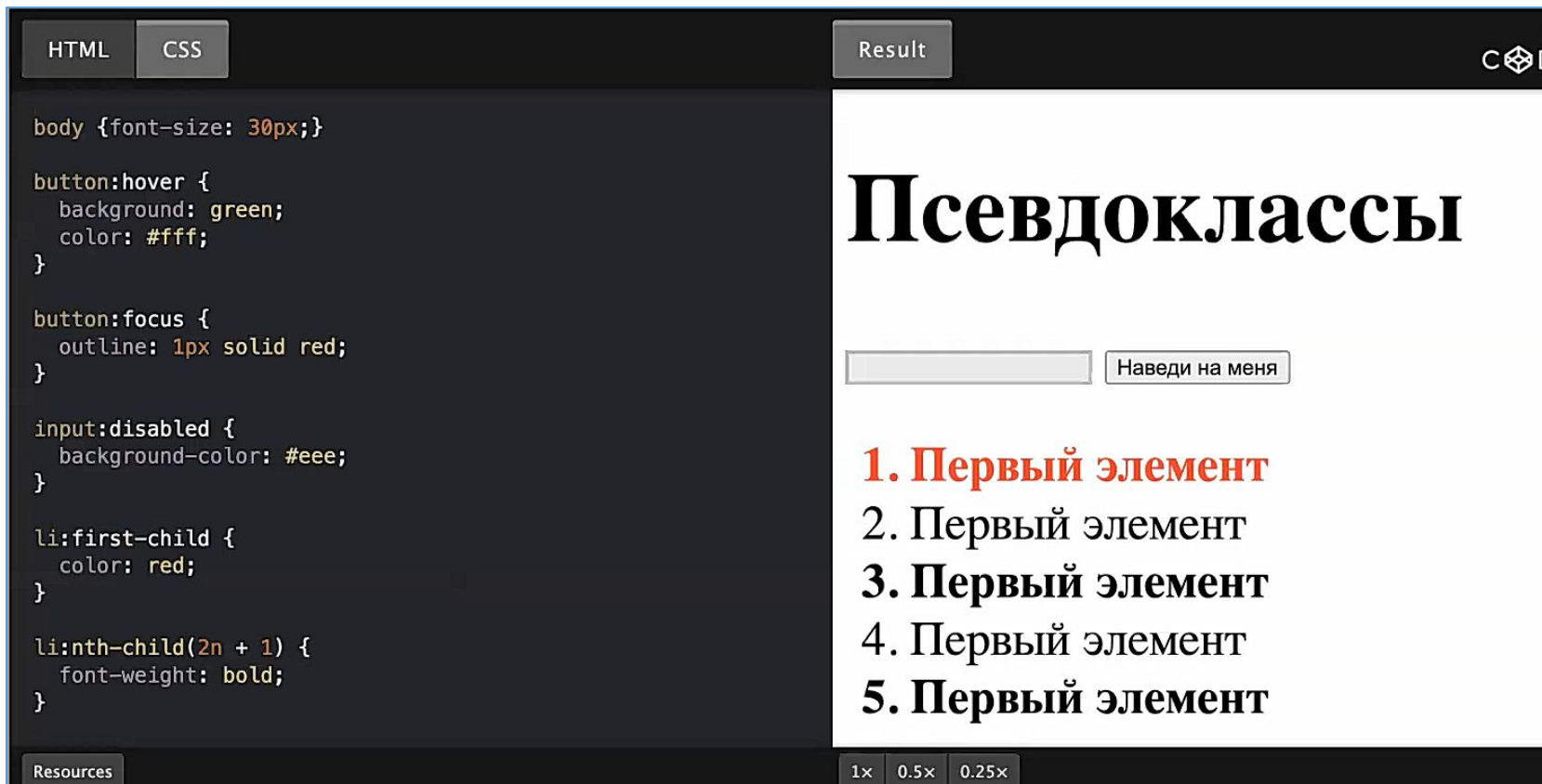
Below these sections, a list is shown:

- Первый элемент
- Второй элемент

At the bottom of the browser interface, there are tabs for 'Resources' and zoom controls (1x, 0.5x, 0.25x).

1. Как видно в примере “content” содержит 3 раза тег `<p> </p>`. Два раза тег `<p> </p>` находятся внутри “content”. Один тег `<p> </p>` не находится внутри “content”, поэтому цвет текста не стал красным.

ПСЕВДОКЛАССЫ



The screenshot shows a web browser interface with a dark theme. On the left, the 'HTML' and 'CSS' tabs are visible, with the CSS tab selected. The code editor contains the following CSS rules:

```
body {font-size: 30px;}

button:hover {
  background: green;
  color: #fff;
}

button:focus {
  outline: 1px solid red;
}

input:disabled {
  background-color: #eee;
}

li:first-child {
  color: red;
}

li:nth-child(2n + 1) {
  font-weight: bold;
}
```

On the right, the 'Result' tab shows the rendered page. The title 'Псевдоклассы' is displayed in a large, bold, black serif font. Below the title, there is a button with the text 'Наведи на меня'. Underneath the button, a list of five items is shown, each starting with 'Первый элемент'. The first item is in red, and the odd-numbered items (1, 3, 5) are in bold.

Resources

1x 0.5x 0.25x



ПСЕВДОКЛАССЫ

Псевдокласс — это селектор, который выбирает элементы, находящиеся в специфическом состоянии, например, они являются первым элементом своего типа, или на них наведён указатель мыши. Они обычно действуют так, как если бы вы применили класс к какой-то части вашего документа, что часто помогает сократить избыточные классы в разметке и даёт более гибкий, удобный в поддержке код.

Псевдоклассы — это ключевые слова, которые начинаются с двоеточия.

:pseudo-class-name

ПРИМЕР CSS

Если бы мы хотели сделать шрифт первого абзаца статьи более крупным и жирным, мы могли бы добавить класс к этому абзацу, а затем добавить CSS к этому классу, как показано в примере справа:

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Interactive editor

```
.first {  
  font-size: 120%;  
  font-weight: bold;  
}
```

```
<article>  
  <p class="first">Veggies es bonus vobis, proinde vos postulo essum magis  
  kohlrabi welsh onion daikon amaranth tatsoi tomatillo  
    melon azuki bean garlic.</p>  
  
  <p>Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette  
  tatsoi pea sprouts fava bean collard  
    greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea  
  peanut soko zucchini.</p>  
</article>
```

Reset

ПРОСТОЙ ПРИМЕР ПСЕВДОКЛА ССА

Однако поддержка может оказаться утомительной — что если новый абзац будет добавлен в верхнюю часть документа? Тогда нам нужно будет передвинуть класс к новому абзацу. Вместо добавления класса мы могли бы использовать селектор псевдокласса `:first-child` — он всегда будет нацелен на первый дочерний элемент в статье, и нам больше не нужно будет редактировать HTML.

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Interactive editor

```
article p:first-child {  
  font-size: 120%;  
  font-weight: bold;  
}
```

```
<article>  
  <p>Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh  
  onion daikon amaranth tatsoi tomatillo  
  melon azuki bean garlic.</p>  
  
  <p>Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette  
  tatsoi pea sprouts fava bean collard  
  greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea  
  peanut soko zucchini.</p>  
</article>
```

Reset

ПСЕВДОКЛАССЫ

Все псевдоклассы ведут себя подобным образом. Они нацелены на какой-то фрагмент вашего документа, находящийся в определённом состоянии, и ведут себя так, как если бы вы добавили класс в свой HTML.

- `:last-child` — находит любой элемент, являющийся последним в его родителе.
- `:only-child` — находит любой элемент, являющийся единственным потомком родителя. Это тоже, что и `:first-child:last-child` или `:nth-child(1):nth-last-child(1)`, но с меньшей специфичностью.
- `:invalid` — находит любые `<input>` или `<form>` элементы, контент которых не проходит валидацию (en-US), в соответствии с типом поля. Он позволяет вам легко менять внешний вид полей, что позволяет пользователю видеть и исправлять ошибки.

ПСЕВДОКЛАСС :last-child

HTML

```
<ul>
  <li>Этот элемент не лаймовый.</li>
  <li>И этот тоже.</li>
  <li>А этот да! :)</li>
</ul>
```



CSS

```
li:last-child {
  background-color: lime;
}
```





ПСЕВДОКЛАСС :last-child

Результат

T.

- Этот элемент не лаймовый.
- И этот тоже.
- А этот да! :)

ПСЕВДОКЛАСС :invalid

HTML

```
<form>  
  <label>Введите URL:</label>  
  <input type="url" />  
  <br />  
  <br />  
  <label>Введите эл. почту:</label>  
  <input type="email" required/>  
</form>
```



Этот пример представляет собой простую форму, цвета элементов которой зелёные, когда данные корректные, и красные, когда нет.



ПСЕВДОКЛАСС CSS :invalid

Этот пример представляет собой простую форму, цвета элементов которой зелёные, когда данные корректные, и красные, когда нет.

CSS

```
input:invalid {  
  background-color: #ffdddd;  
}  
  
form:invalid {  
  border: 5px solid #ffdddd;  
}  
  
input:valid {  
  background-color: #ddffdd;  
}  
  
form:valid {  
  border: 5px solid #ddffdd;  
}  
  
input:required {  
  border-color: #800000;  
  border-width: 3px;  
}
```

ПСЕВДОКЛАСС :only-child

Синтаксис.

```
parent child:only-child
{
    property: value;
}
```

```
span:only-child {
    color: red;
}
```

```
<div>
  <span>Этот span единственный ребёнок своего папы:</span>
</div>

<div>
  <span>Этот span один из потомков родителя</span>
  <span>Этот span один из детей отца</span>
</div>
```

Результат

Этот span единственный ребёнок своего папы:(
Этот span один из потомков родителя Этот span один из детей отца

ПСЕВДОКЛАСС :only-child

```
li li {  
  list-style-type: disc;  
}  
li:only-child {  
  color: #6699ff;  
  font-style: italic;  
  list-style-type: square;  
}
```

Пример со
списком.

```
<ol>  
  <li>Первый  
    <ul>  
      <li>Это единственный ребёнок  
    </ul>  
  </li>  
  <li>Второй  
    <ul>  
      <li>Этот список с двумя элементами  
      <li>Этот список с двумя элементами  
    </ul>  
  </li>  
  <li>Третий  
    <ul>  
      <li>Этот список с тремя элементами  
      <li>Этот список с тремя элементами  
      <li>Этот список с тремя элементами  
    </ul>  
  </li>  
</ol>
```



ПСЕВДОКЛАСС :only-child

Результат

1. Первый
 - *Это единственный ребёнок*
2. Второй
 - Этот список с двумя элементами
 - Этот список с двумя элементами
3. Третий
 - Этот список с тремя элементами
 - Этот список с тремя элементами
 - Этот список с тремя элементами



ДИНАМИЧЕСКИЕ ПСЕВДОКЛАССЫ

Некоторые **псевдоклассы** применяются только тогда, когда пользователь некоторым образом взаимодействует с документом. Эти **псевдоклассы** действий пользователя, иногда называемые **динамическими псевдоклассами**, действуют так, как если бы класс был добавлен к элементу в момент взаимодействия с ним

- `:hover` — он применяется только в том случае, если пользователь наводит указатель мыши на элемент, обычно на ссылку.
- `:focus` — применяется только в том случае, если пользователь фокусируется на элементе, используя управление с клавиатуры.

ДИНАМИЧЕСКИЕ ПСЕВДОКЛАССЫ


[Hover over me](#)

Interactive editor

```
a:link,  
a:visited {  
  color: rebeccapurple;  
  font-weight: bold;  
}  
  
a:hover {  
  color: hotpink;  
}
```

```
<p><a href="">Hover over me</a></p>
```

Reset



ПСЕВДОКЛАССЫ

1

:active;

2

:link;

3

:focus;

4

:hover;

5

:visites.

ПСЕВДОЭЛЕМЕНТЫ

The image shows a browser's developer tools interface. On the left, the 'CSS' tab is active, displaying the following CSS code:

```
body {font-size: 30px;}

p:first-letter {
  text-transform: uppercase;
  color: green;
}

p:first-line {
  color: red;
}

a[target="_blank"]:after {
  content: ' (откроется в новой вкладке)';
  font-size: 14px;
}
```

On the right, the 'Result' tab shows the rendered page. The main heading is 'Псевдоэлементы'. Below it, the first letter 'П' is green and uppercase. The first line of the paragraph is red. A link is followed by the text ' (откроется в новой вкладке)' in a smaller font size.

Resources: 1x 0.5x 0.25x



ПСЕВДОЭЛЕМЕНТЫ

Псевдоэлементы ведут себя сходным образом, однако они действуют так, как если бы вы добавили в разметку целый новый HTML-элемент, а не применили класс к существующим элементам. Псевдоэлементы начинаются с двойного двоеточия ::.

ПСЕВДОЭЛЕМЕНТЫ



1

:active;

2

:link;

3

:focus;

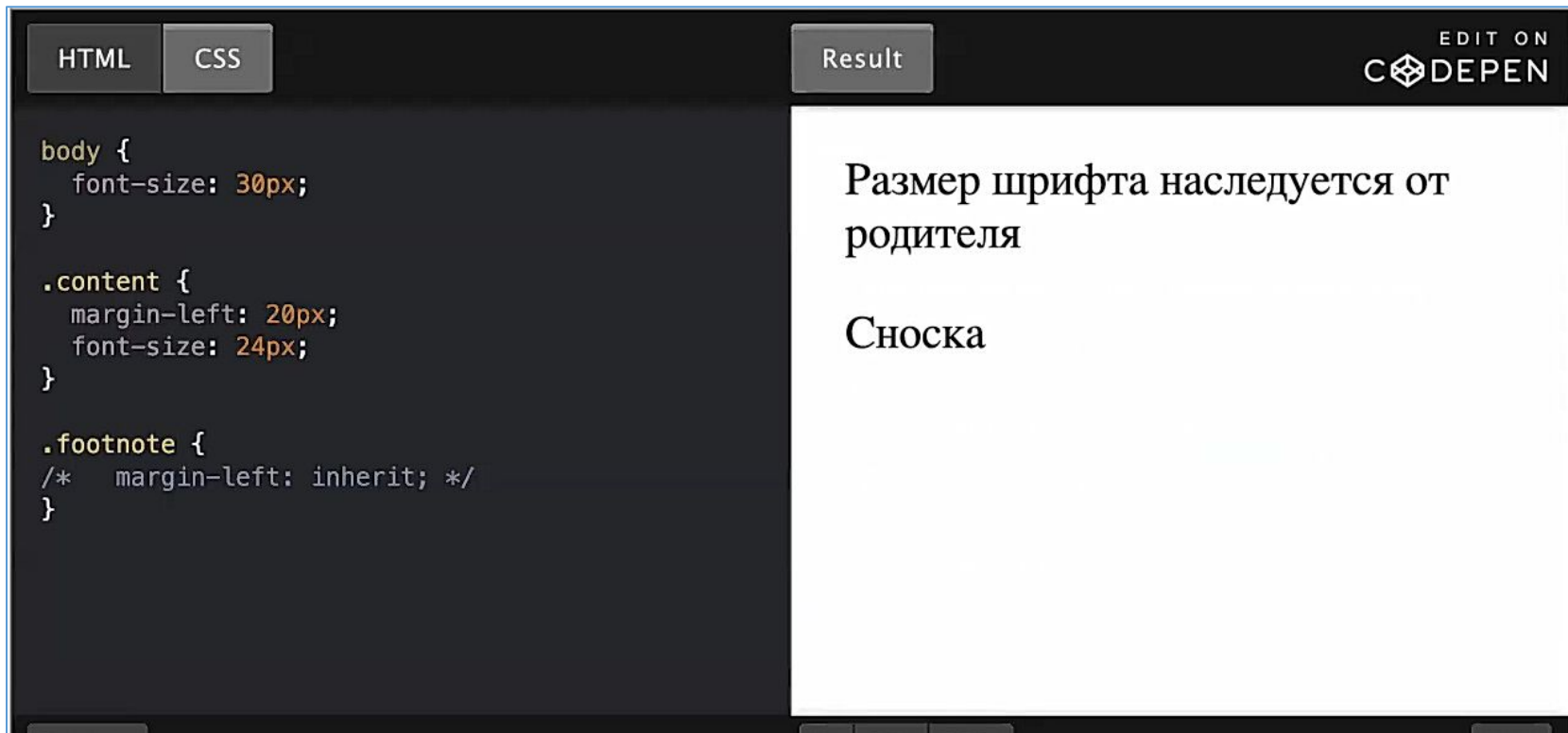
4

:hover;

5

:visites.

НАСЛЕДОВАНИЕ



HTML CSS Result EDIT ON CODEPEN

```
body {  
  font-size: 30px;  
}  
  
.content {  
  margin-left: 20px;  
  font-size: 24px;  
}  
  
.footnote {  
  /* margin-left: inherit; */  
}
```

Размер шрифта наследуется от родителя

Сноска

НАСЛЕДОВАНИЕ

The screenshot displays a web browser interface with a dark theme. At the top, there are navigation buttons for 'Settings', 'Sign Up', and 'Log In'. The main content area is divided into three panels:

- HTML:** Shows the following code:

```
1 <section class="content">
2 <p>Размер шрифта наследуется от родителя</p>
3
4 <footer class="footnote">Сноска</footer>
5 </section>
```
- CSS:** Shows the following code:

```
1 body {
2   font-size: 30px;
3 }
4
5 .content {
6   margin-left: 20px;
7   font-size: 24px;
8 }
9
10 .footnote {
11   /* margin-left: inherit; */
12 }
```
- Browser View:** Shows the rendered page with the text "Размер шрифта наследуется от родителя" and "Сноска".
- Developer Tools:** The 'Styles' panel is open, showing the style for the selected paragraph element. It lists the styles inherited from the parent `.content` class, including `margin-left: 20px;` and `font-size: 24px;`. The `font-size` property is highlighted in red, indicating it is inherited from the parent.

Во всех современных браузерах есть просмотр DOM дерева, инспектор HTML.

HTML: Элемент `<section>` у которого class «content». Внутри него есть теги `<p></p>` и `<footer> </footer>` с классом «footnote».

CSS: `margin-left` (отступ слева), `font-size` (размер шрифта).

Тег `<p></p>` унаследовал стили у своего родителя «content». Стили для отступов `margin-left` по-умолчанию не наследуется.

НАСЛЕДОВАНИЕ

The screenshot displays a web browser's developer tools interface. The HTML panel on the left shows the following code:

```
1 <section class="content">
2 <p>Размер шрифта наследуется от родителя</p>
3
4 <footer class="footnote">Сноска</footer>
5 </section>
```

The CSS panel in the middle shows the following rules:

```
1 body {
2   font-size: 30px;
3 }
4
5 .content {
6   margin-left: 20px;
7   font-size: 24px;
8 }
9
10 .footnote {
11 /* margin-left: inherit; */
12 }
```

The Styles panel on the right shows the computed style for the selected element (a footnote). The 'margin-left' property is set to 'inherit', and the 'Inherited from' section shows it is inherited from the '.content' selector. The 'font-size' property is also shown as inherited from the 'body' selector.

The browser's main content area shows the rendered page with the text "Размер шрифта наследуется от родителя" and "Сноска".

Унаследовать ненаследуемое свойство с помощью «inherit». Для .footnot унаследовать свойство «margin-left».



НАСЛЕДУЕМЫЕ СТИЛИ

```
color, font, letter-spacing, line-height, list-style,  
text-align, text-indent, text-transform, visibility,  
white-space и word-spacing
```

Это стили, которые относятся: к шрифту, к размеру, к цветам текста.




НЕНАСЛЕДУЕМЫЕ СТИЛИ

background, border, display, float и clear, height и width, margin, min-max-height и -width, outline, overflow, padding, position, text-decoration, vertical-align и z-index

Это стили, которые относятся: к размеру, отступам, фоновые картинки.

СПЕЦИФИЧНОСТЬ СЕЛЕКТОРОВ

```
HTML CSS Result   
  
<h1>Специфичность селекторов</h1>  
<div id="foo">  
  <span class="bar">Текст</span>  
</div>  
  
<style>  
  div .bar {color: green;}  
  #foo .bar {color: blue;}  
  div span {color: orange;}  
</style>  
  
Resources
```

К одному и тому же HTML элементу `` можно применить стили с помощью разных селекторов.

Зеленый цвет для всех элементов у которых есть класс «.bar» и они должны находится внутри div. Все 3 селектора подходят.

СПЕЦИФИЧНОСТЬ СЕЛЕКТОРОВ

```
HTML CSS Result
<h1>Специфичность селекторов</h1>
<div id="foo">
  <span class="bar">Текст</span>
</div>

<style>
  div .bar {color: green;}
  #foo .bar {color: blue;}
  div span {color: orange;}
</style>
```

Специфичность селекторов

Текст

Resources 1x 0.5x 0.25x

Цвет текста синий, так как для определения того, какое свойство будет в итоге у нас используется понятие специфичность селекторов или вес селектора. Вес «#foo .bar» больше, чем других.

Если специфичность одинаковая, то применялось свойство, которое написано ниже.

РАСЧЕТ СПЕЦИФИЧНОСТЬ

Селектор	Специфичность
тег, псевдо-элемент(span, ::after)	0 0 0 1

0001 – каждая цифра от 0 до 256-тиричное число. Каждый тег или псевдоэлемент добавляет единицу в 1 разряд.

Селектор	Специфичность
тег, псевдо-элемент(span, ::after)	0 0 0 1
класс, псевдокласс, атрибут (.header, :hover)	0 0 1 0

0010 – Класс, псевдокласс, атрибут добавляет единицу во 2 разряд.

РАСЧЕТ СПЕЦИФИЧНОСТЬ

Селектор	Специфичность
тег, псевдо-элемент(span, ::after)	0 0 0 1
класс, псевдокласс, атрибут (.header, :hover)	0 0 1 0
идентификатор (#footer)	0 1 0 0

0100 – идентификатор добавляет единицу во 3 разряд.

Селектор	Специфичность
тег, псевдо-элемент(span, ::after)	0 0 0 1 ⁱ
класс, псевдокласс, атрибут (.header, :hover)	0 0 1 0
идентификатор (#footer)	0 1 0 0
инлайн-стили (style="color: red;")	1 0 0 0

1000 – инлайн-стили добавляет единицу в 4 разряд.

РАСЧЕТ СПЕЦИФИЧНОСТЬ

```
HTML Result
```

```
<h1>Специфичность селекторов</h1>
<div id="foo">
  <span class="bar">Текст</span>
</div>

<style>
  div .bar {color: green;} /* 0 0 1 1 */
  #foo .bar {color: blue;} /* 0 1 1 0 */
  div span {color: orange;} /* 0 0 0 2 */

  span {color: orange !important;}
</style>
```

«#foo .bar» больше, чем другие селекторы.

Свойство «important» позволяет повысить специфичность.

РАСЧЕТ СПЕЦИФИЧНОСТЬ

```
HTML
```

```
<h1>Специфичность селекторов</h1>
<div id="foo">
  <span class="bar">Текст</span>
</div>

<style>
  div .bar {color: green;} /* 0 0 1 1 */
  #foo .bar {color: blue;} /* 0 1 1 0 */
  div span {color: orange;}/* 0 0 0 2 */

  span {color: orange !important;}
</style>
```

```
Result
```

Специфичность селекторов

Текст

```
Resources
```

1x 0.5x 0.25x

Свойство «important» позволяет повысить специфичность. Поэтому текст стал оранжевым.



СЕЛЕКТОРЫ

1

*;

2

p;

3

.class;

4


#id;

5

[name="value"];

6

:focus.



СЛОЖНЫЕ СЕЛЕКТОРЫ

1

.h1.h2;

2

p:before;

3

ol ol;

4

ul > ul.

ПРОБЛЕМЫ

```
.text #name div.red.small { ... }
```

- сложно понимать
- сложно переиспользовать
- сложно переопределить
- усложняет поиск селекторов браузерами
- лишние селекторы

Селектор `.text #name div.red.small { ... }`

Если захотите переиспользовать какие-то стили, которые описаны тут { ... },

чтобы они использовались в другом месте вашего кода, то необходимо повторить структуру `.text #name div.red.small`.

Если переопределяете, то пишите более сложный селектор.



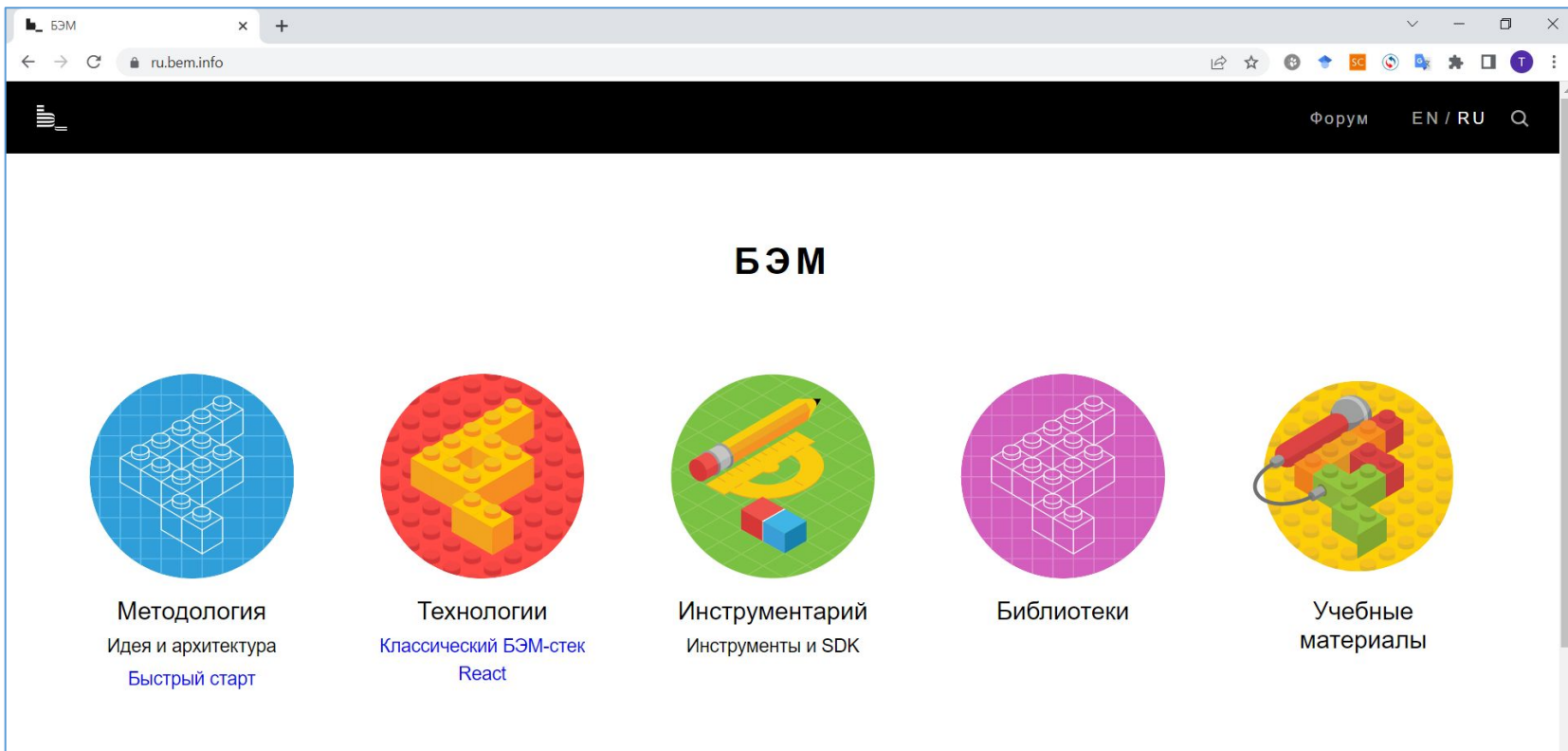
РЕШЕНИЕ

- Используйте селекторы классов
- Повышайте специфичность в крайнем случае
- БЭМ методология

БЭМ (БЛОК, ЭЛЕМЕНТ, МОДИФИКАТОР) – компонентный подход в веб-разработке. В его основе лежит принцип разделения интерфейса на независимые блоки.

Придумали его в Яндексе. БЭМ – широкая методология.

БЭМ



<https://ru.bem.info/>



ЗНАЧЕНИЯ АТТРИБУТОВ

1

Текстовые: "так" или
'так';

2

Размер
ы:

2.1

px;

2.2

em, rem;

2.3

vw, vh, vmin, vmax.



ЧТО ПОЧИТАТЬ?

- Hakon Wium Lie, Bert Bos «Cascading Style Sheets: Designing for the Web (3rd Edition)».
- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Ian Pouncey, Richard York «Beginning CSS: Cascading Style Sheets for Web Design».
- Ummy Gusti Salamah, S.ST,. MIT «Tutorial Cascading Style Sheets (CSS)».
- Steven E. Callihan «Cascading Style Sheets (CSS) by Example».