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

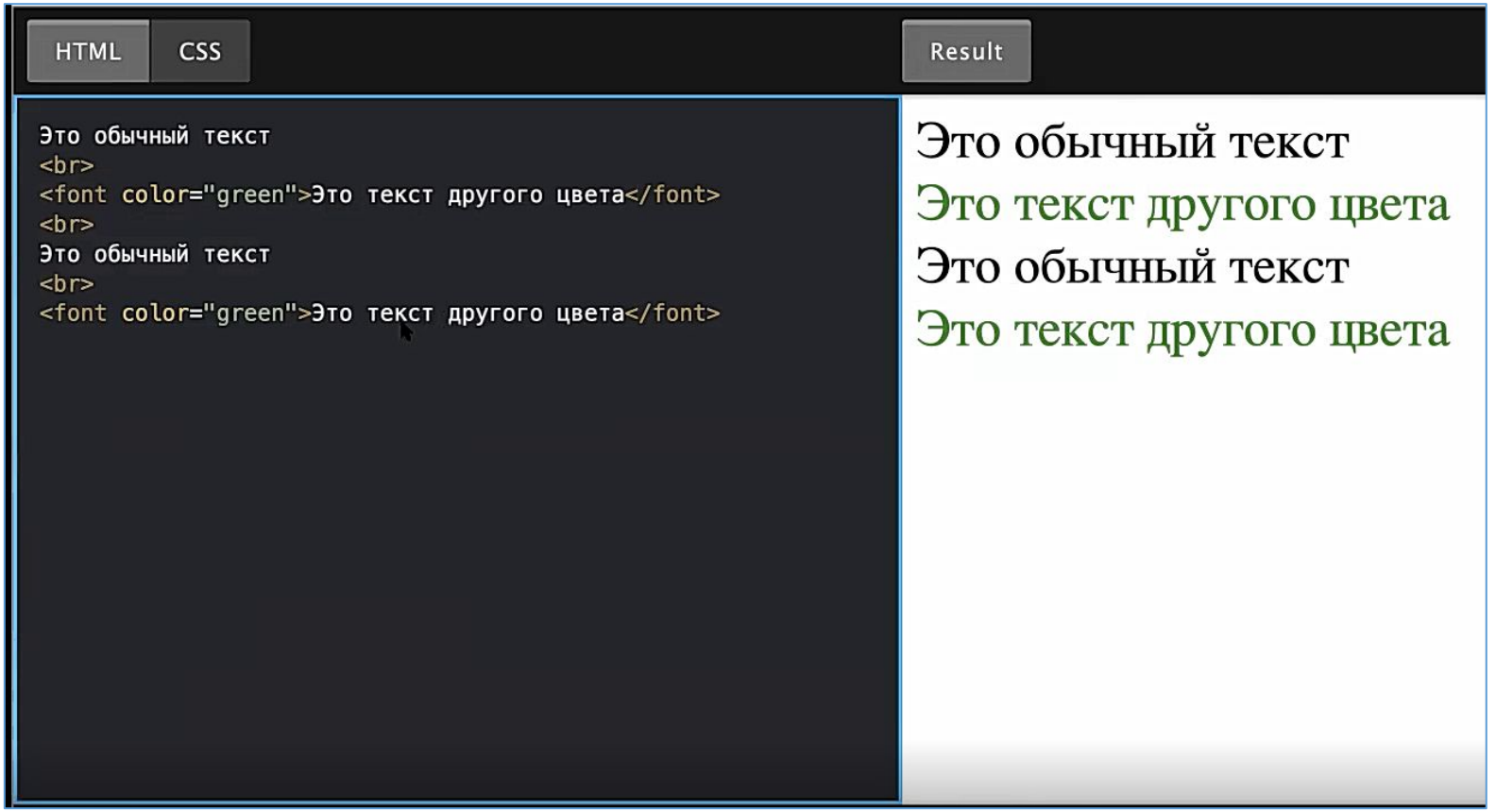
Учёный, специалист в области информатики, предложивший каскадные таблицы стилей (CSS). В англоязычных текстах его часто называют *howcome*, а норвежское имя считается трудночитаемым.

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




**Тег <font> </font> через атрибут color задает зеленый цвет.**

# ПРИМЕР СТИЛЕЙ

HTML

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

CSS

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

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

**Тег <font> </font> 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

```
Это обычный текст
<br>
<span class="primary">Это текст другого цвета</span>
<br>
Это обычный текст
<br>
<span class="primary">Это текст другого цвета</span>

<style>
  .primary {
    color: red;
  }
</style>
```

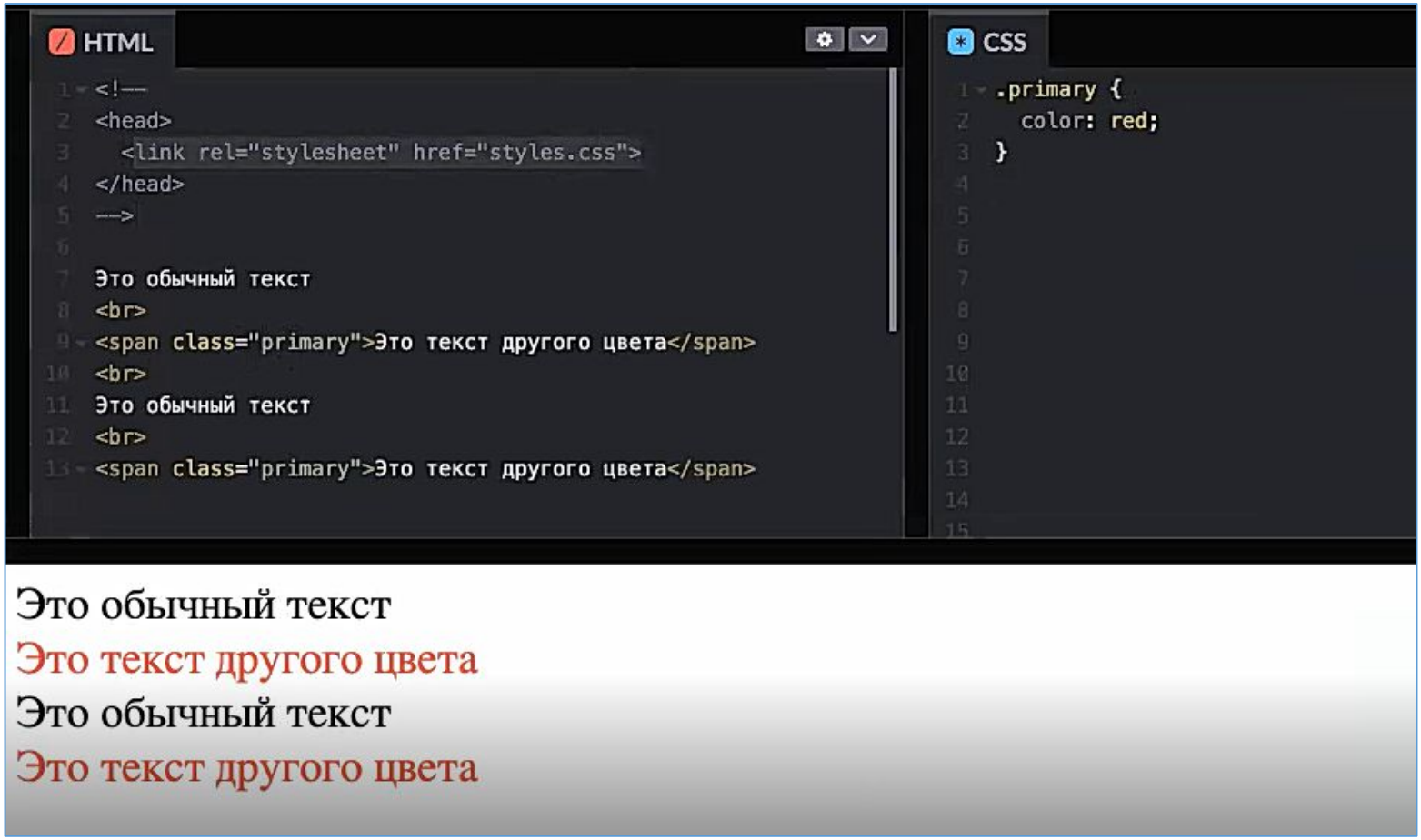
CSS

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

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




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

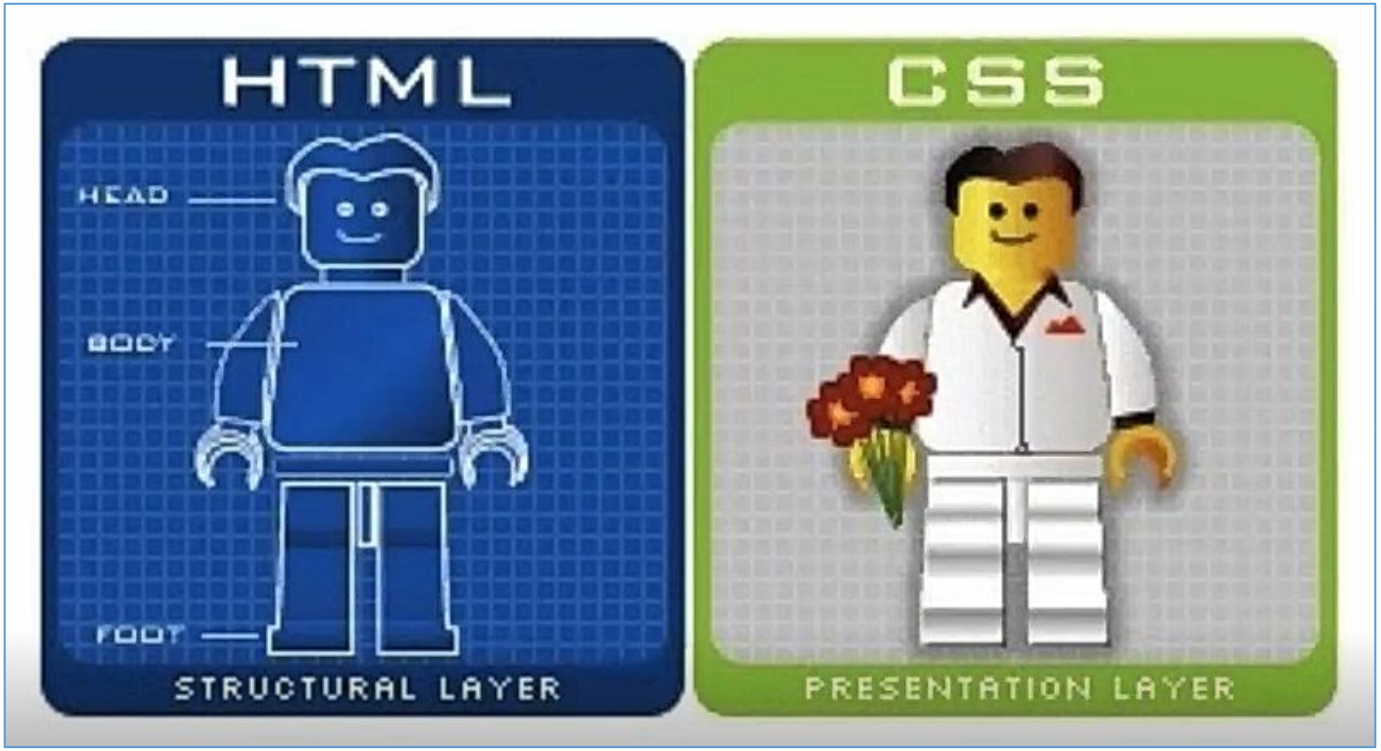
# стиль

```
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

HTML

```
<main>
  <header>
    <div class="logo">
      <img src="https://assets.codepen.io/652/floral_design_logo.png" alt="Penelope's Floral
Design" />
      <h1 class="visually-hidden">Penelope's Floral Design</h1>
    </div>
  </header>

  <div class="flowers">
    <div class="flower">
```

CSS (SCSS)

```
main {
  margin-block: 1rem;
  * {
    margin: 0;
    padding: 0;
  }
  > * + * {
    margin-block-start: 1.5rem;
  }
}
```

JS

# ПРИМЕРЫ CSS

HTML

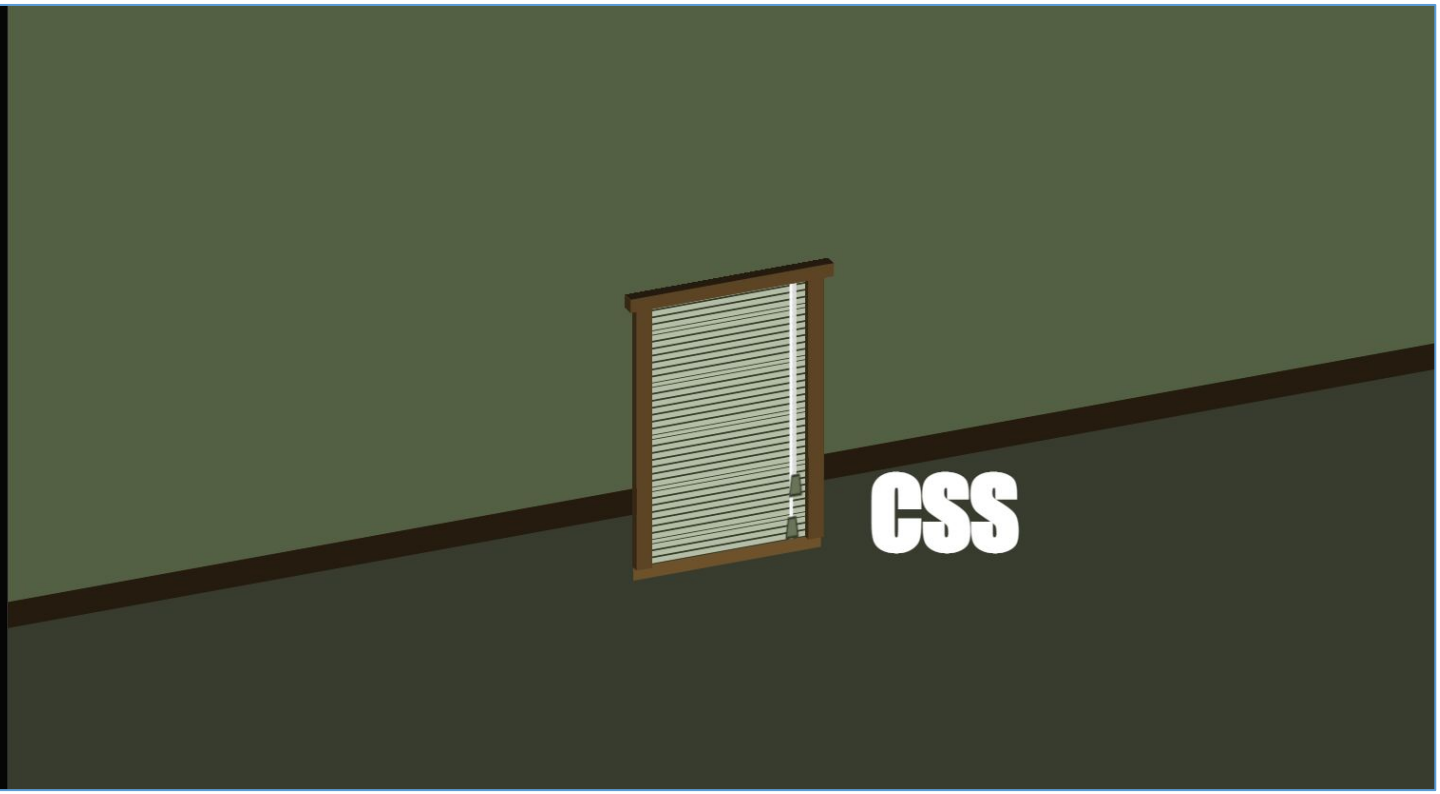
```
        <div class="cuboid__side"></div>
        <div class="cuboid__side"></div>
        <div class="cuboid__side"></div>
        <div class="cuboid__side"></div>
        <div class="cuboid__side"></div>
        <div class="cuboid__side"></div>
      </div>
    </div>
    <div class="frame frame--right">
      <div class="cuboid window-frame
window-frame--right">
```

CSS

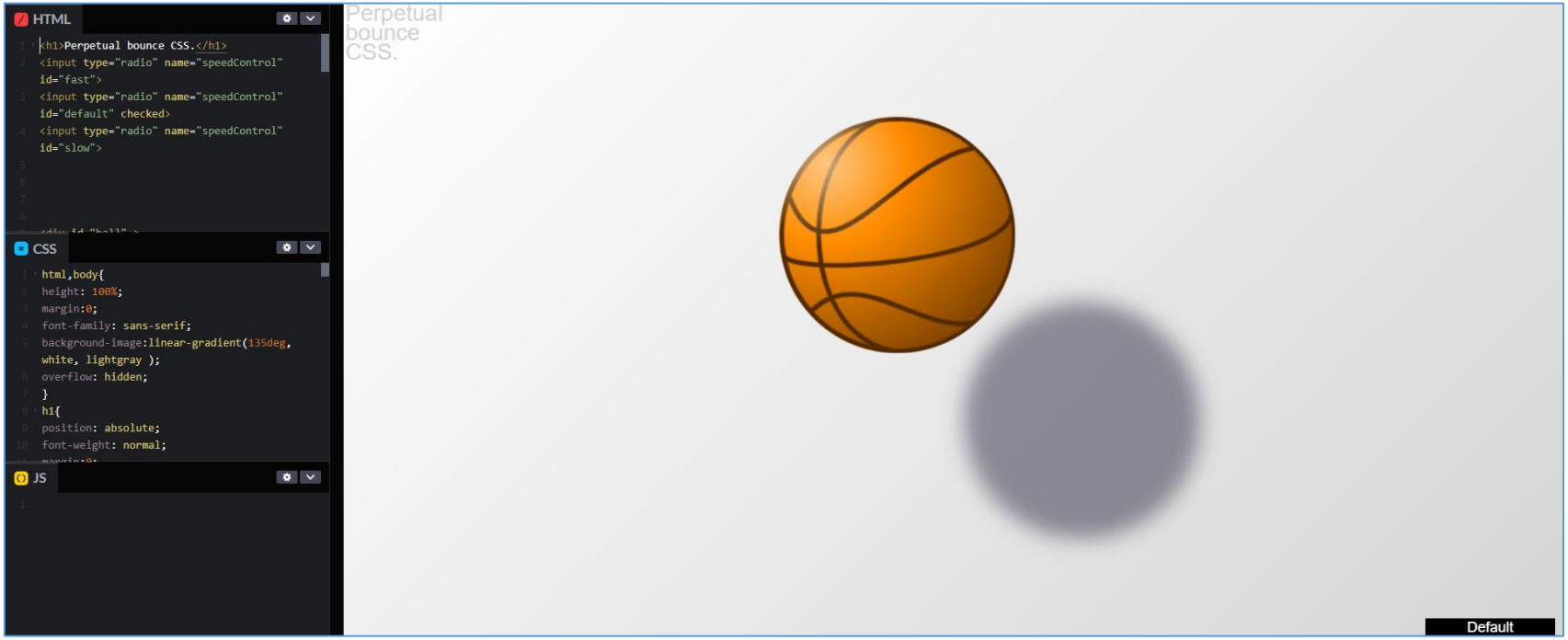
```
@font-face {
  font-family: Impact;
  src:
url(https://assets.codepen.io/605876/impa
ct.ttf);
}

@layer demo-stuff, container-stuff;

@layer container-stuff {
  *, *:after, *:before {
    pointer-events: none;
  }
  .container {
    pointer-events: all;
    container-type: size;
  }
  @container (min-height: 75px) {
    .blinds {
      --open: 0;
    }
  }
  @container (min-height: 100px) {
```



# ПРИМЕРЫ CSS


HTML
<h1>Perpetual bounce CSS.</h1>
<input type="radio" name="speedControl" id="fast">
<input type="radio" name="speedControl" id="default" checked>
<input type="radio" name="speedControl" id="slow">
CSS
html,body{
height: 100%;
margin:0;
font-family: sans-serif;
background-image:linear-gradient(135deg, white, lightgray );
overflow: hidden;
}
h1{
position: absolute;
font-weight: normal;
JS
Perpetual bounce CSS.
Default

# ПРИМЕРЫ CSS

HTML

```
<span class="jump">🎃</span>
<span class="jump alt">🎃</span>
<span class="jump">🎃</span>
<span class="jump alt">🎃</span>
<span class="jump">🎃</span>
<br>
<span class="jump alt">🎃</span>
<span class="jump">🎃</span>
<span class="jump alt">🎃</span>
<span class="jump">🎃</span>
<span class="jump alt">🎃</span>
```

CSS

```
body{width:100%;text-align:center;margin-top:1em;}
*{font-size:2em;}

/*
https://gist.github.com/davidsharp/a2cc3c8988dbfb8e5d291d1d084396cf */
.jump {
  display: inline-block;
  animation-duration: 1.5s;
  animation-name: jump;
  animation-iteration-count: infinite;
}
.jump.alt {
  animation-delay: .75s;
}
@keyframes jump {
  0%, 100% {
    transform: scale(1.1,1)
translateY(0);
  }
```

# ПРИМЕРЫ CSS

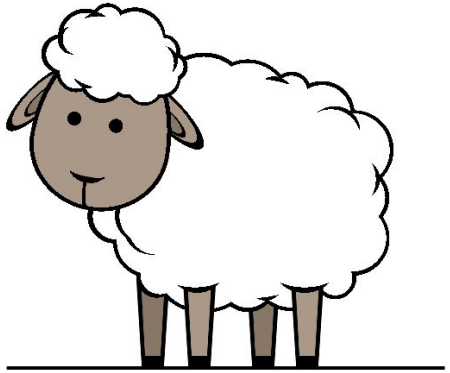
HTML

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

CSS

```
article {
  --skin: #a98;
  --skin-dark: #876;
  width: 60vmin;
  aspect-ratio: 1.2;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  background:
    /* details 1 */
    radial-gradient(130% 140% at 74.5%
48%, #fff 7.5%, #0000 0),
    radial-gradient(110% 100% at 77.5%
48%, #000 7%, #0000 0),
    radial-gradient(circle at 74% 37%,
#fff 7.5%, #0000 0),
    radial-gradient(90% 100% at 74.8%
34.5%, #000 6.5%, #0000 0),
    radial-gradient(circle at 32% 58%,
#fff 7.5%, #0000 0),
    radial-gradient(circle at 34% 61%,
#fff 7.5%, #0000 0),
```

JS



# ПРИМЕРЫ CSS

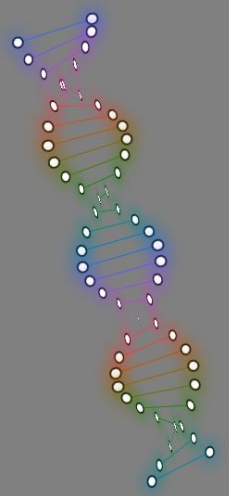
HTML

```
<div class="items">
  <div class="item"></div>
  <div class="item"></div>
  <div class="item"></div>
```

CSS (SCSS)

```
}

html {
  display: flex;
  justify-content: center;
  align-items: center;
  min-width: 100vw;
  min-height: 100vh;
}
```

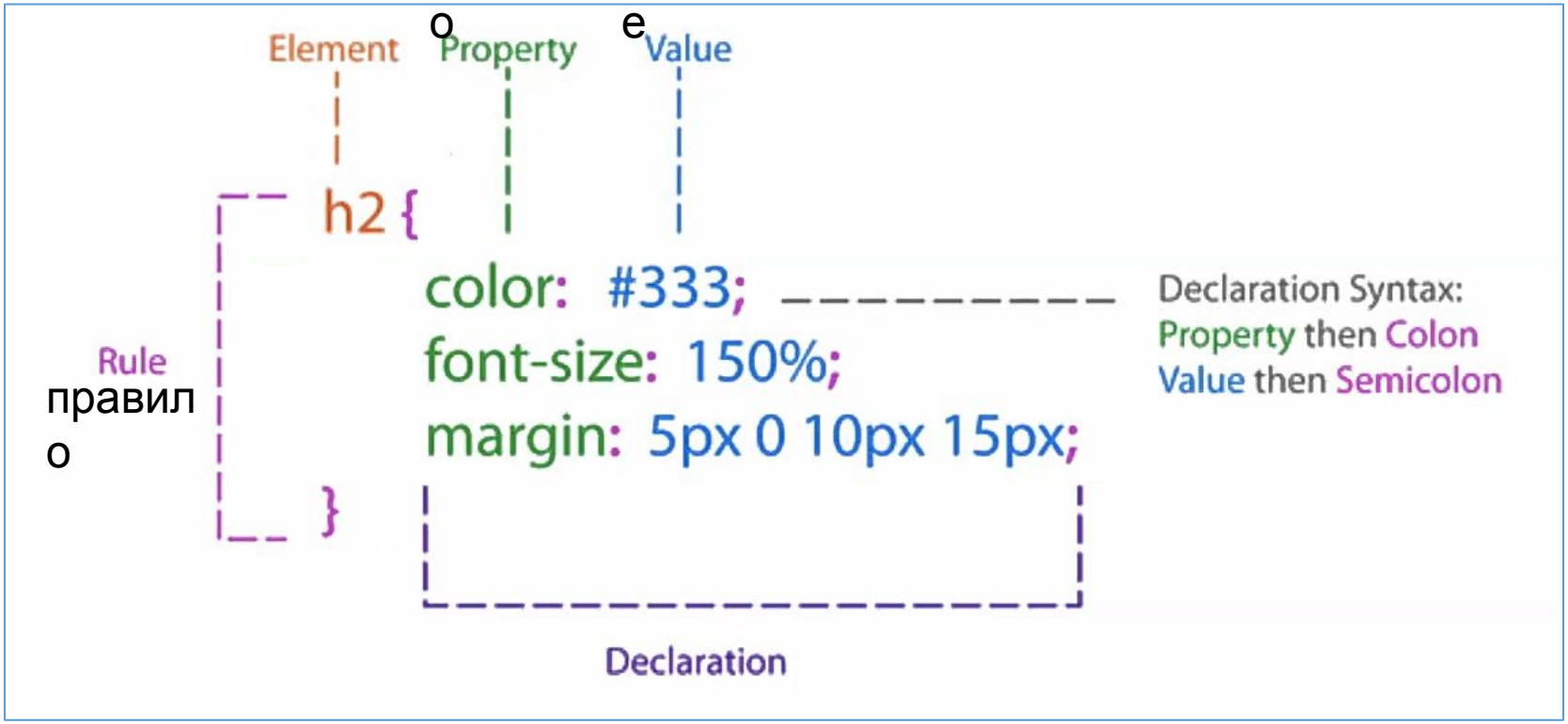


# ПРЕИМУЩЕСТВА CSS

- 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

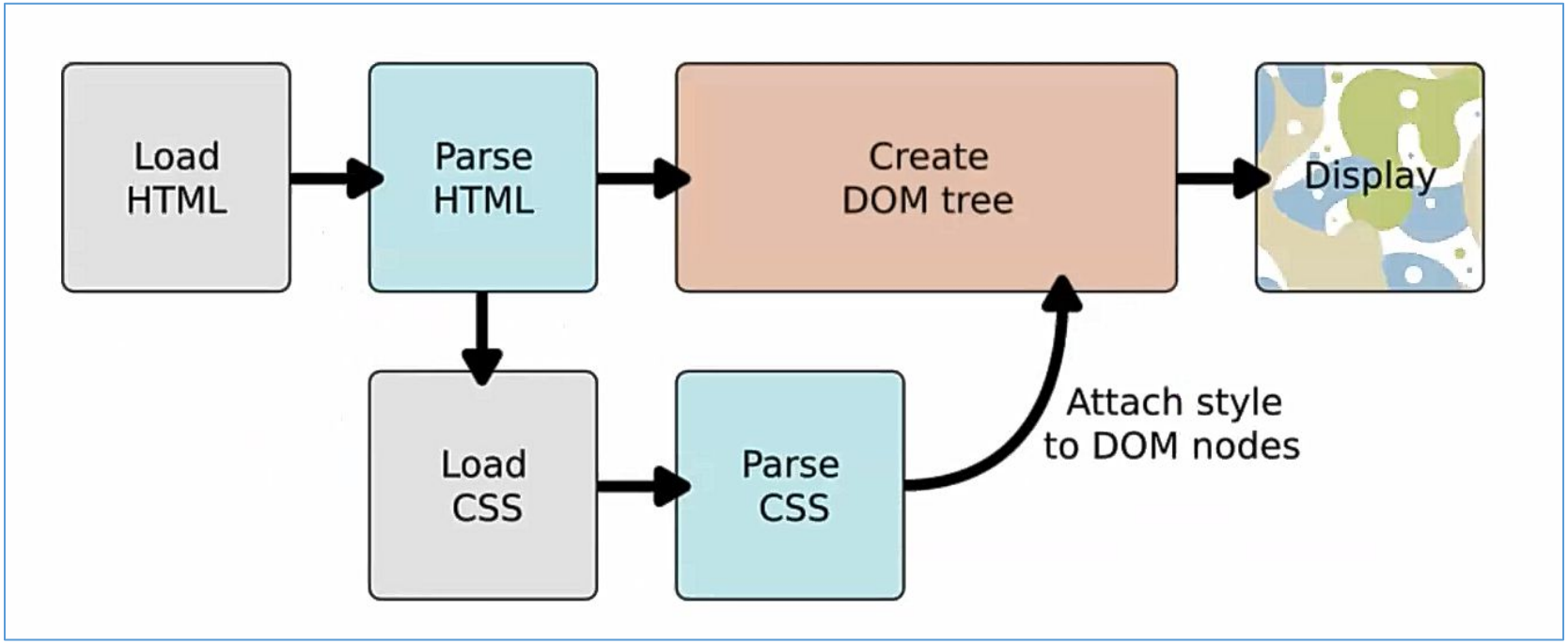
селектор свойство значение




правило

описание

# СХЕМА РАБОТЫ БРАУЗЕРА

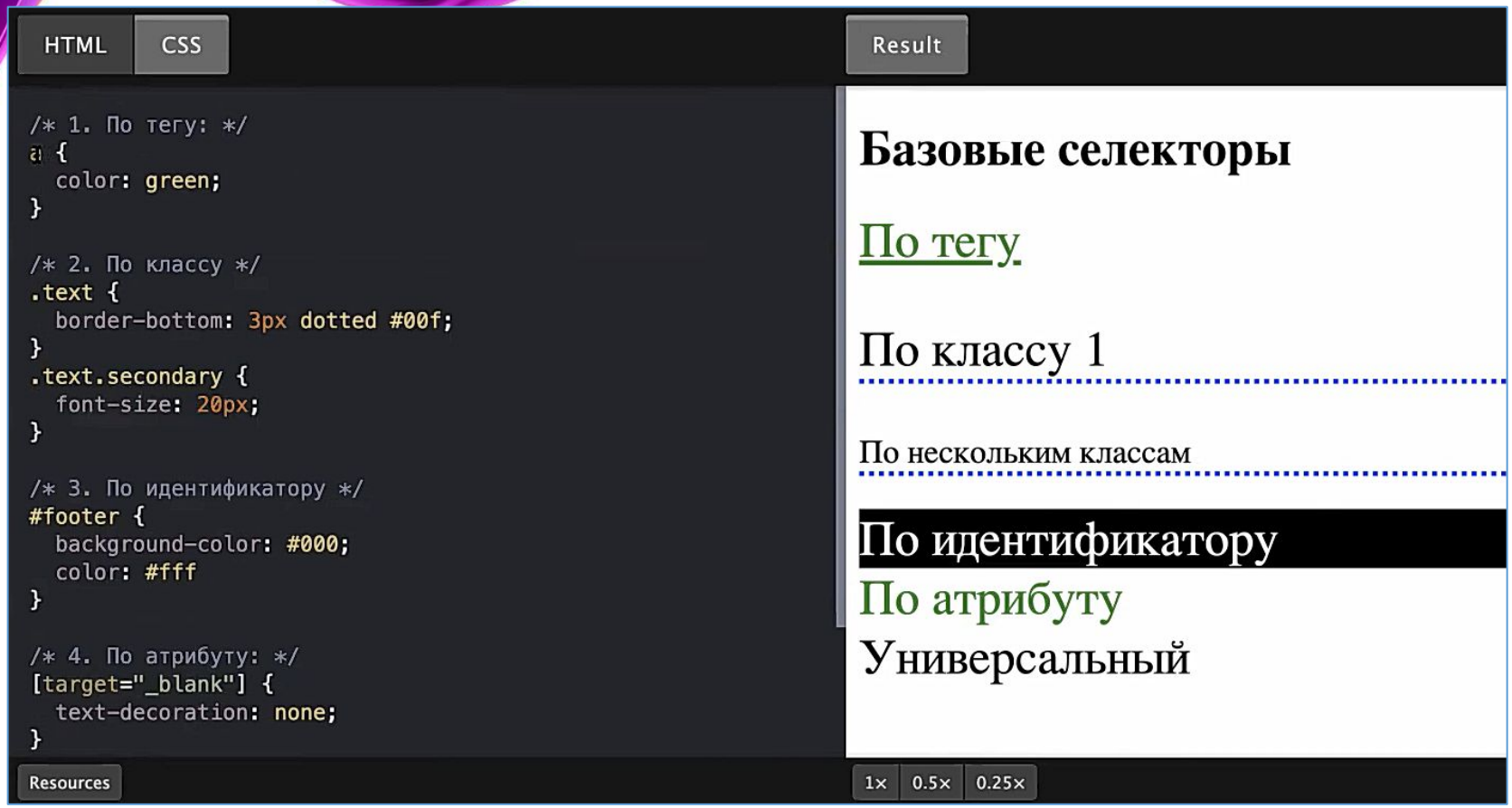



CSS COM дерево накладывается на DOM дерево=результирующее дерево.

# ТИПЫ СЕЛЕКТОРОВ

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

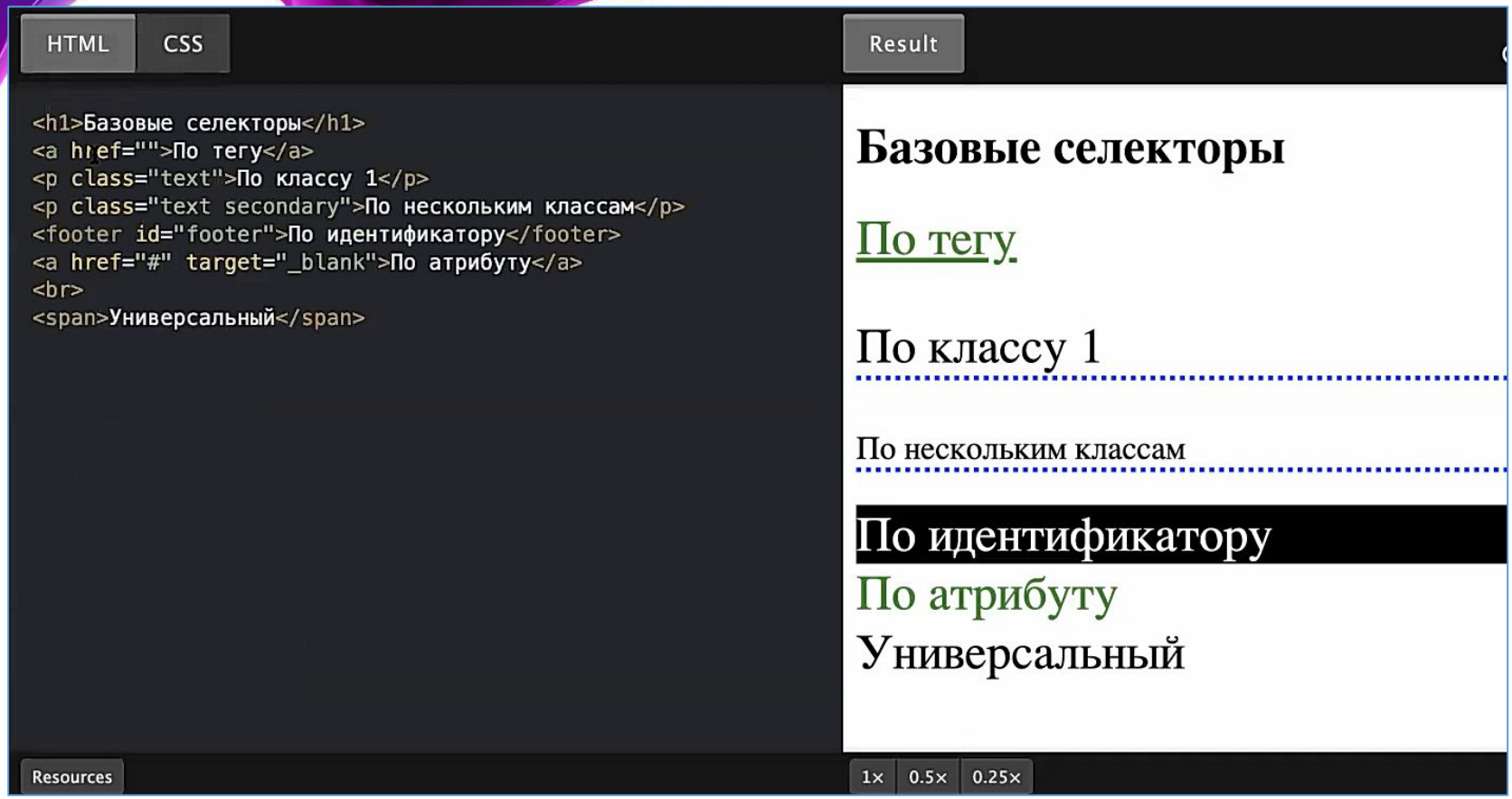
# БАЗОВЫЕ СЕЛЕКТОРЫ




Базовых селекторов существует 5 видов:

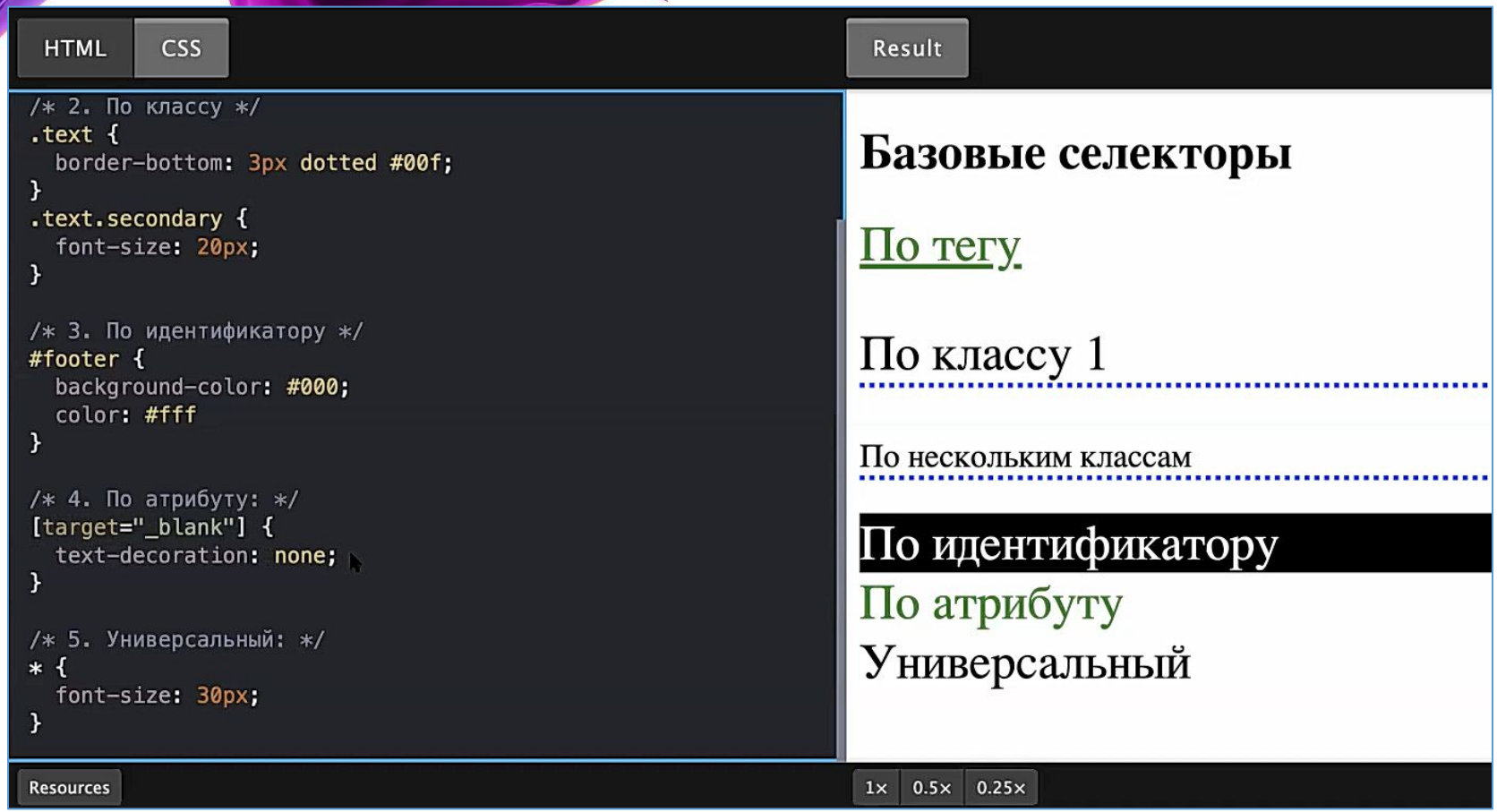
1. Color green применяется ко всем тегам <a> </a>.
2. Стиль по классу: для всех элементов у которых есть .text сделаем границу 3px dotted (точечную) #00f (синего цвета).

# БАЗОВЫЕ СЕЛЕКТОРЫ



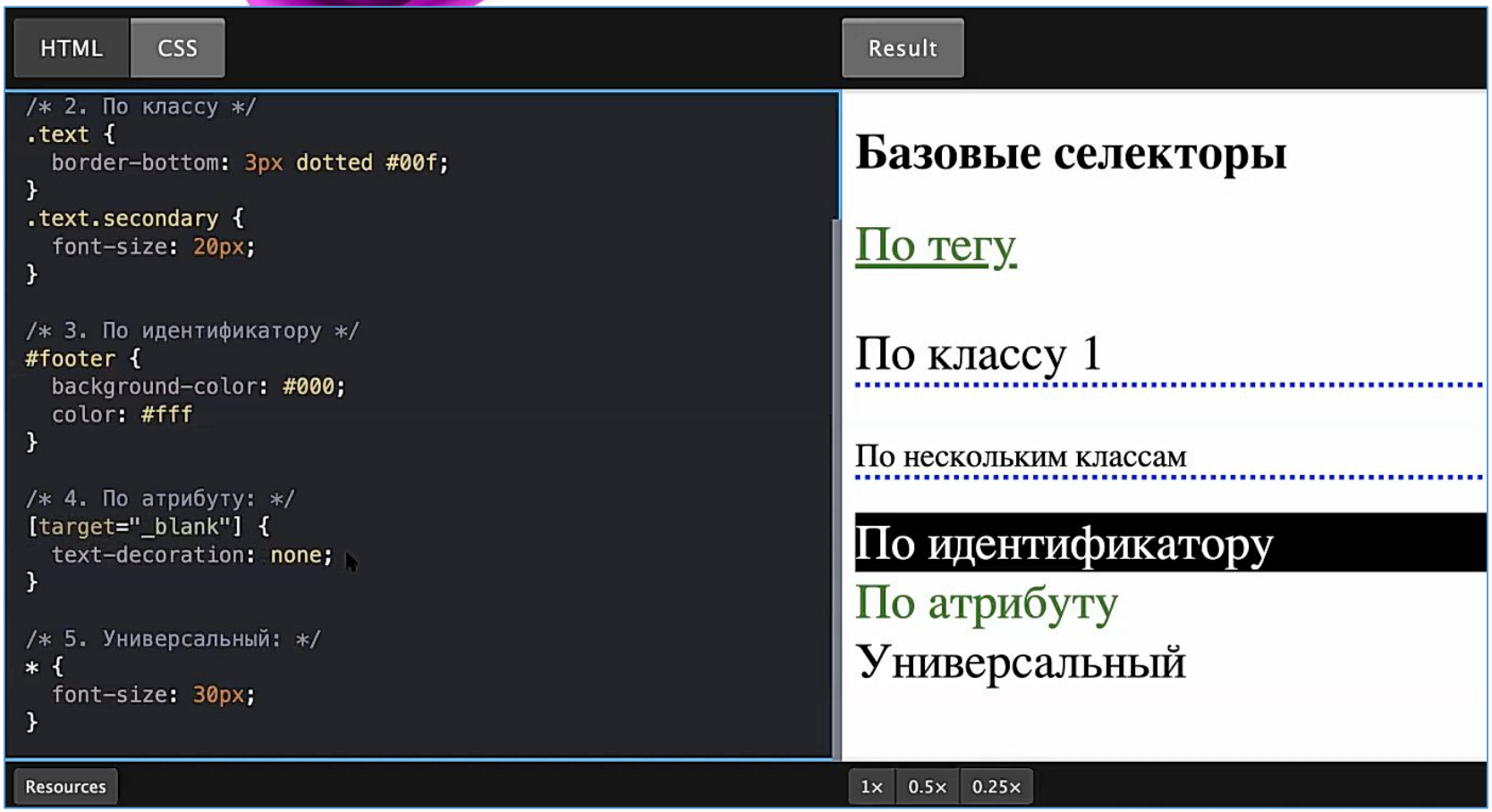

class – это атрибут, а “text” – значение.
Если несколько class для одного элемента может быть несколько, разделяются они пробелом “text secondary”.
ID – идентификатор. “footer” – имя идентификатора может быть только одно. “footer” имя, которое мы придумали.

# БАЗОВЫЕ СЕЛЕКТОРЫ




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-collor (цвет фона элемента) #000 (black), #fff (white).

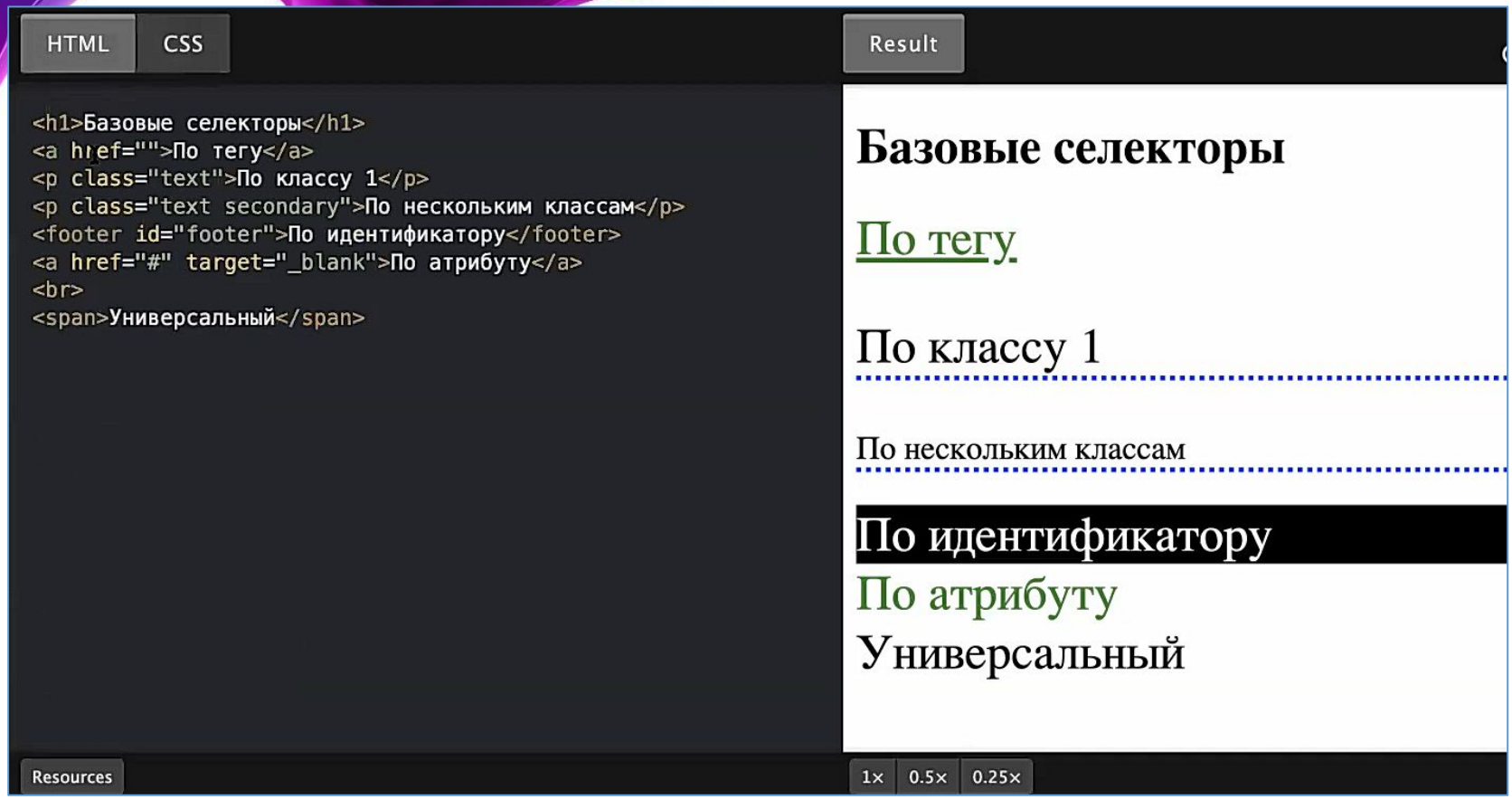
# БАЗОВЫЕ СЕЛЕКТОРЫ




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".

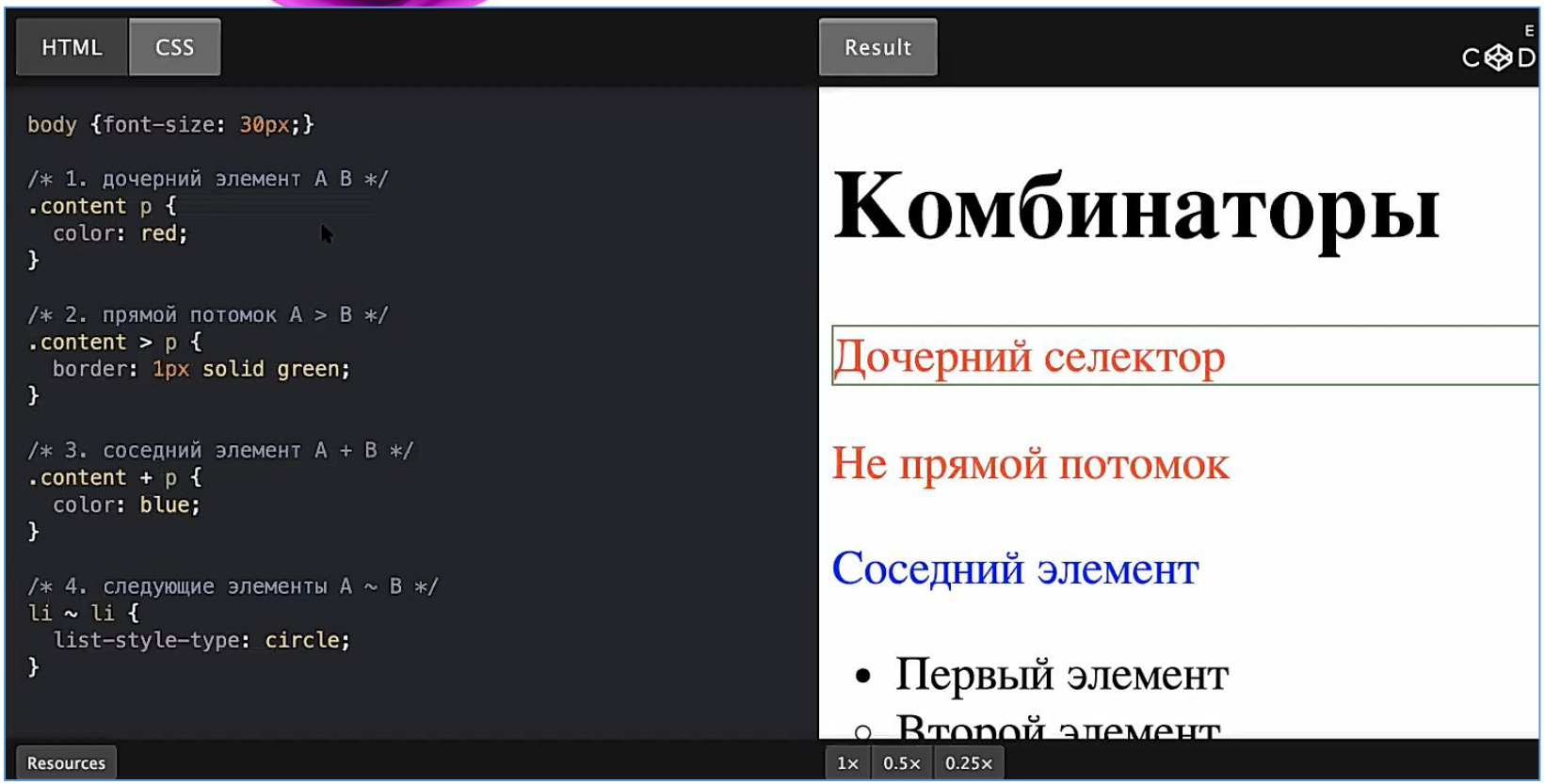
5. Универсальный селектор. * применится ко всем тегам.

# БАЗОВЫЕ СЕЛЕКТОРЫ



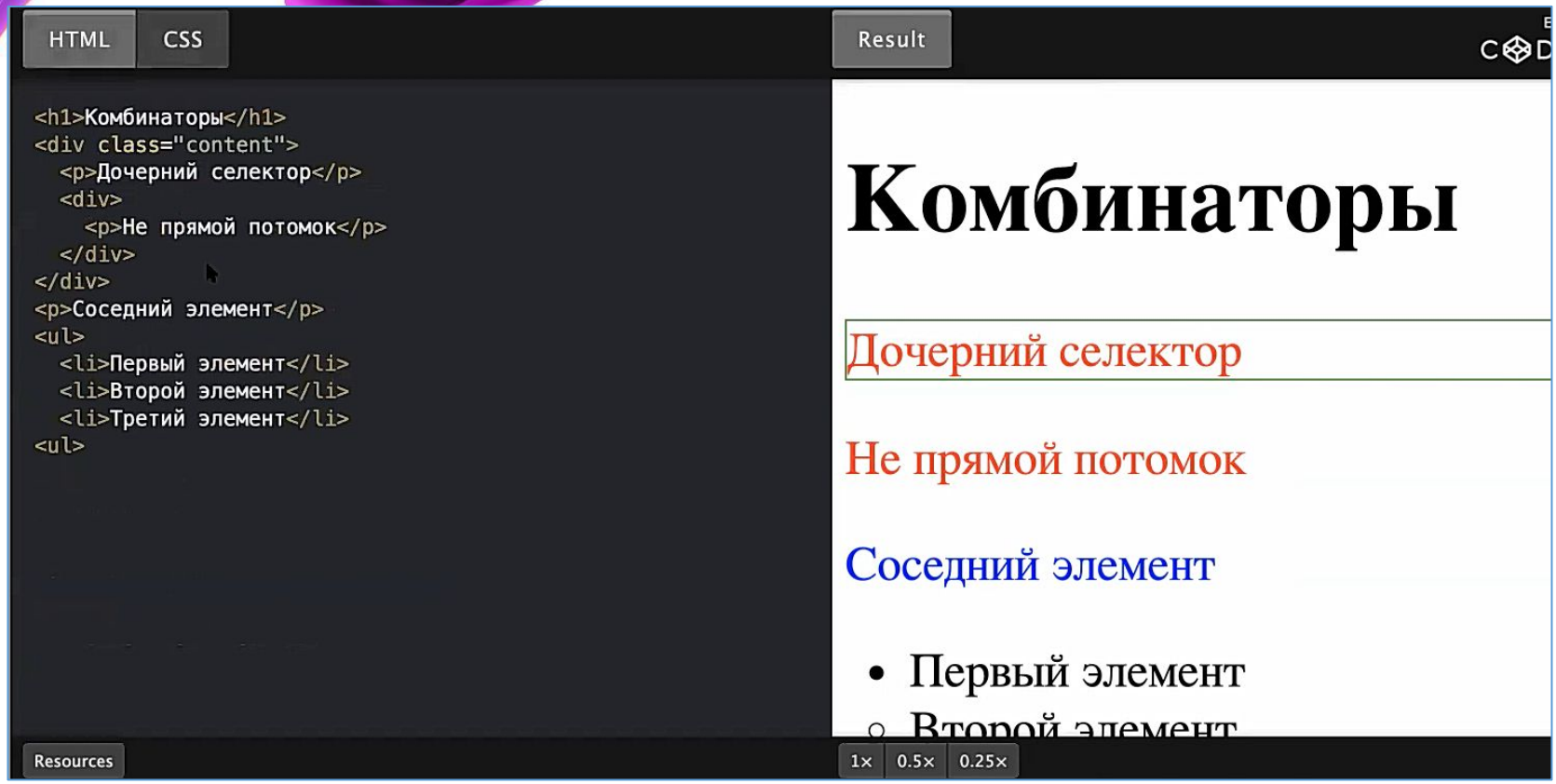

target="_blank", т. е ссылка должна открываться в новой вкладке.

# КОМБИНАТОРЫ



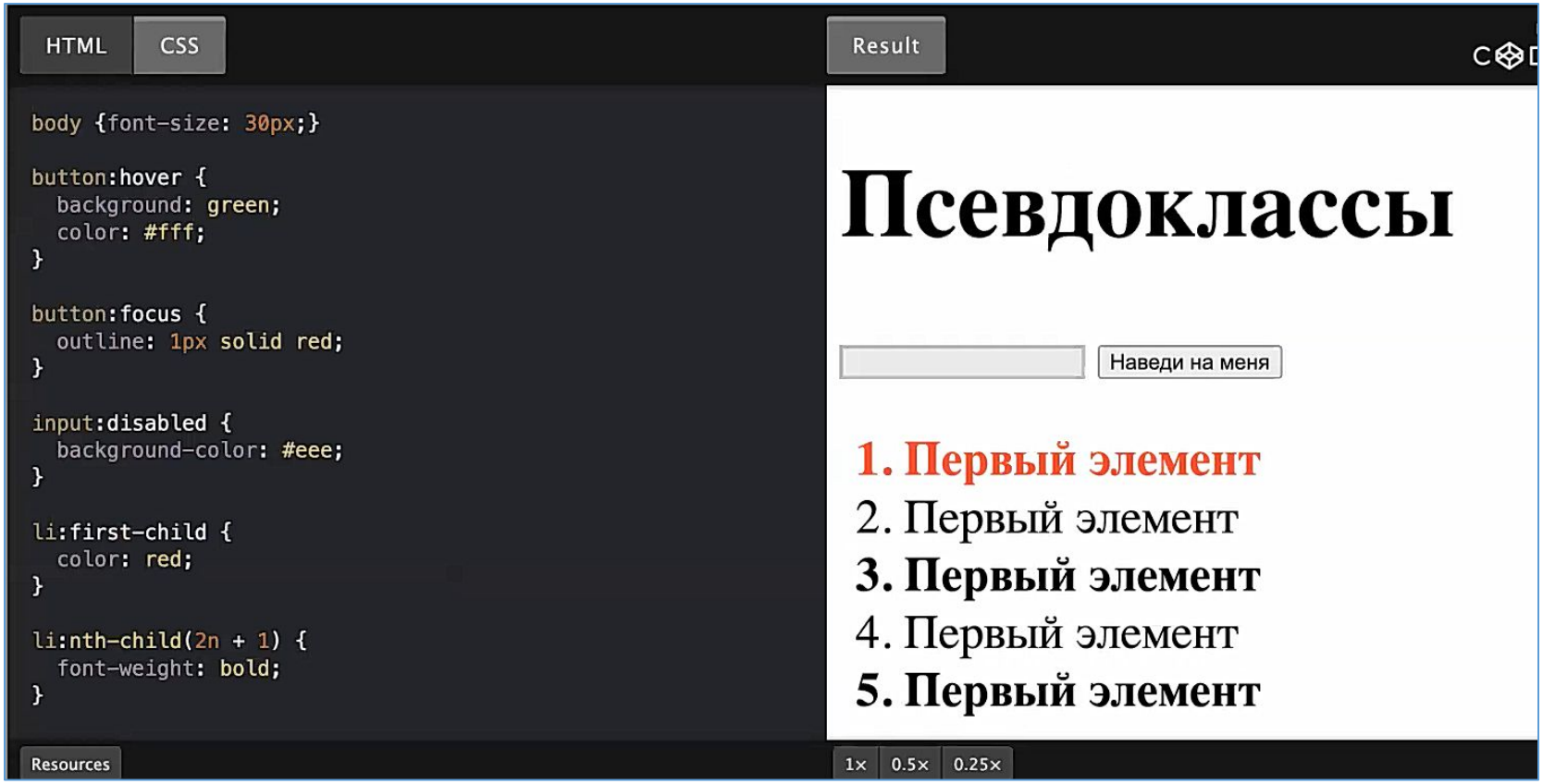

1. Дочерний элемент часто используется. Применить цвет красный (color: red) ко всем элементам «p», которые находятся внутри элементов с классом «.content».
2. Применять к непосредственным детям «.content > p».
3. Применять к соседним элементам «.content + p».
4. Применять к  следующим элементам «li ~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

# КОМБИНАТОРЫ




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

# ПСЕВДОКЛАССЫ



# 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>
```

Reset

# ПРОСТОЙ ПРИМЕР ПСЕВДОКЛАССА

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.

# 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

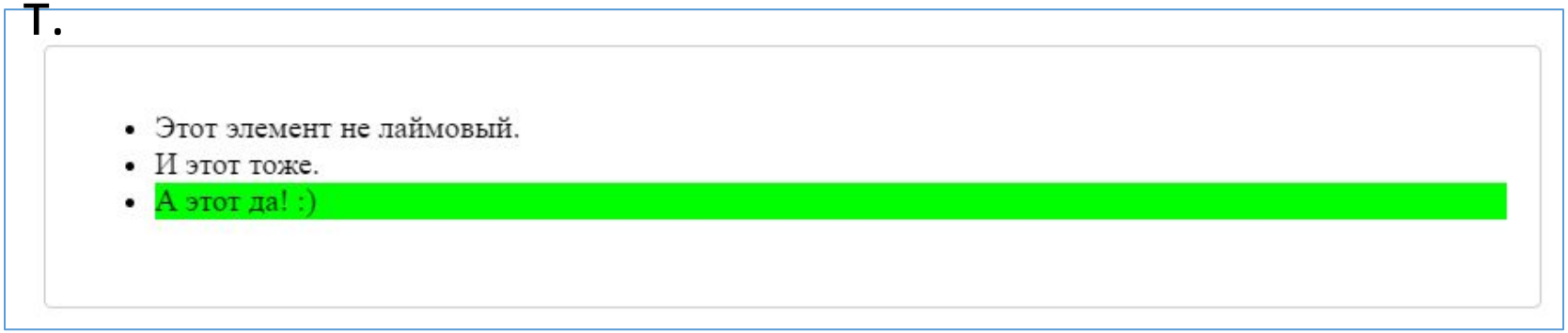
```
<ul>
  <li>Этот элемент не лаймовый.</li>
  <li>И этот тоже.</li>
  <li>А этот да! :)</li>
</ul>
```

## CSS

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

# ПСЕВДОКЛАСС :last-child

Результат.



# 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.

# ПСЕВДОКЛАССС :invalid

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

CSS

```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;
}
```

# ПСЕВДОКЛАССС :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

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

Пример со списком.

```html
<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;
}
```

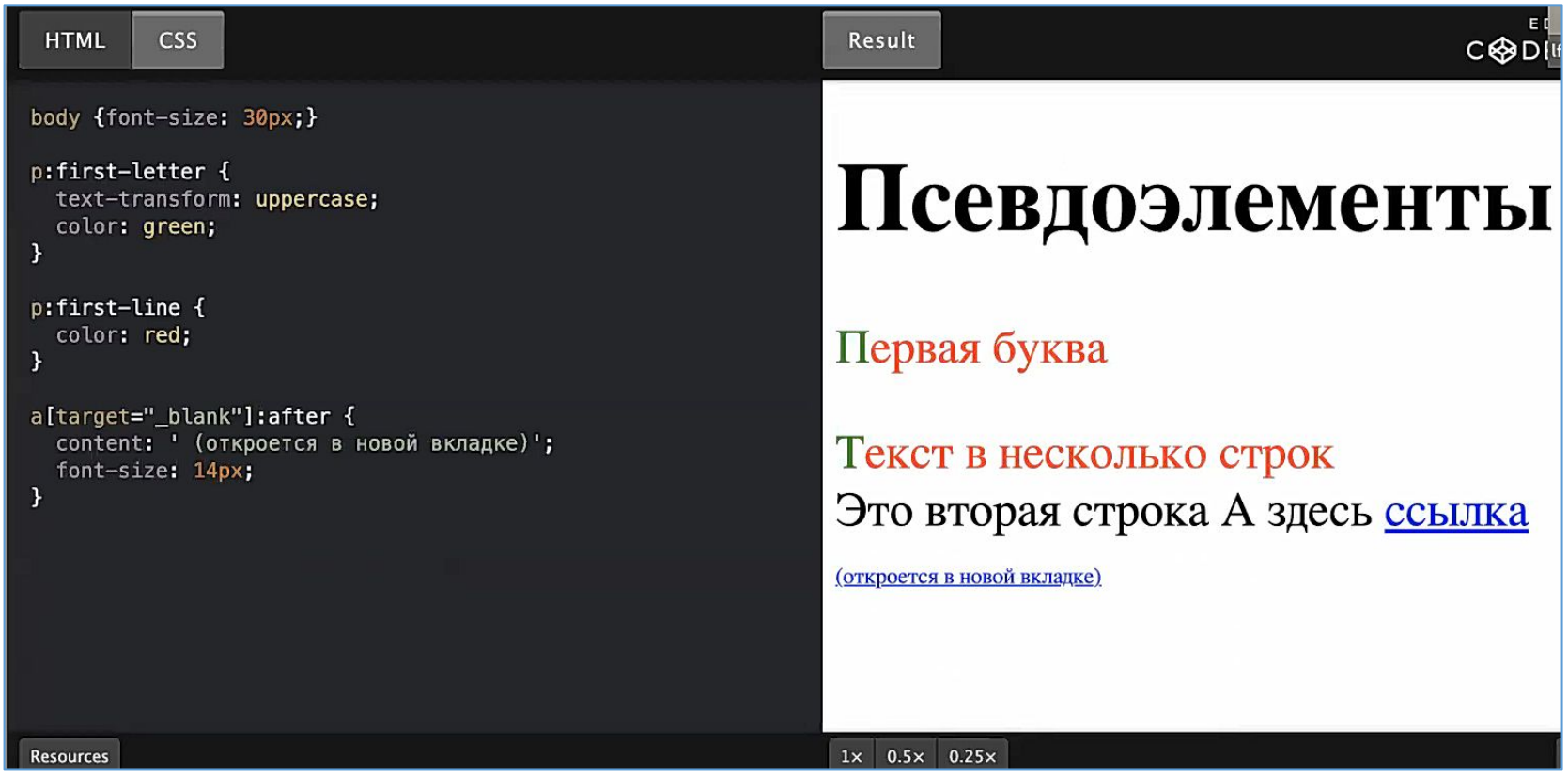
```
<p><a href="">Hover over me</a></p>
```

Reset

# ПСЕВДОКЛАССЫ

1. :active;
2. :link;
3. :focus;
4. :hover;
5. :visites.

# ПСЕВДОЭЛЕМЕНТЫ



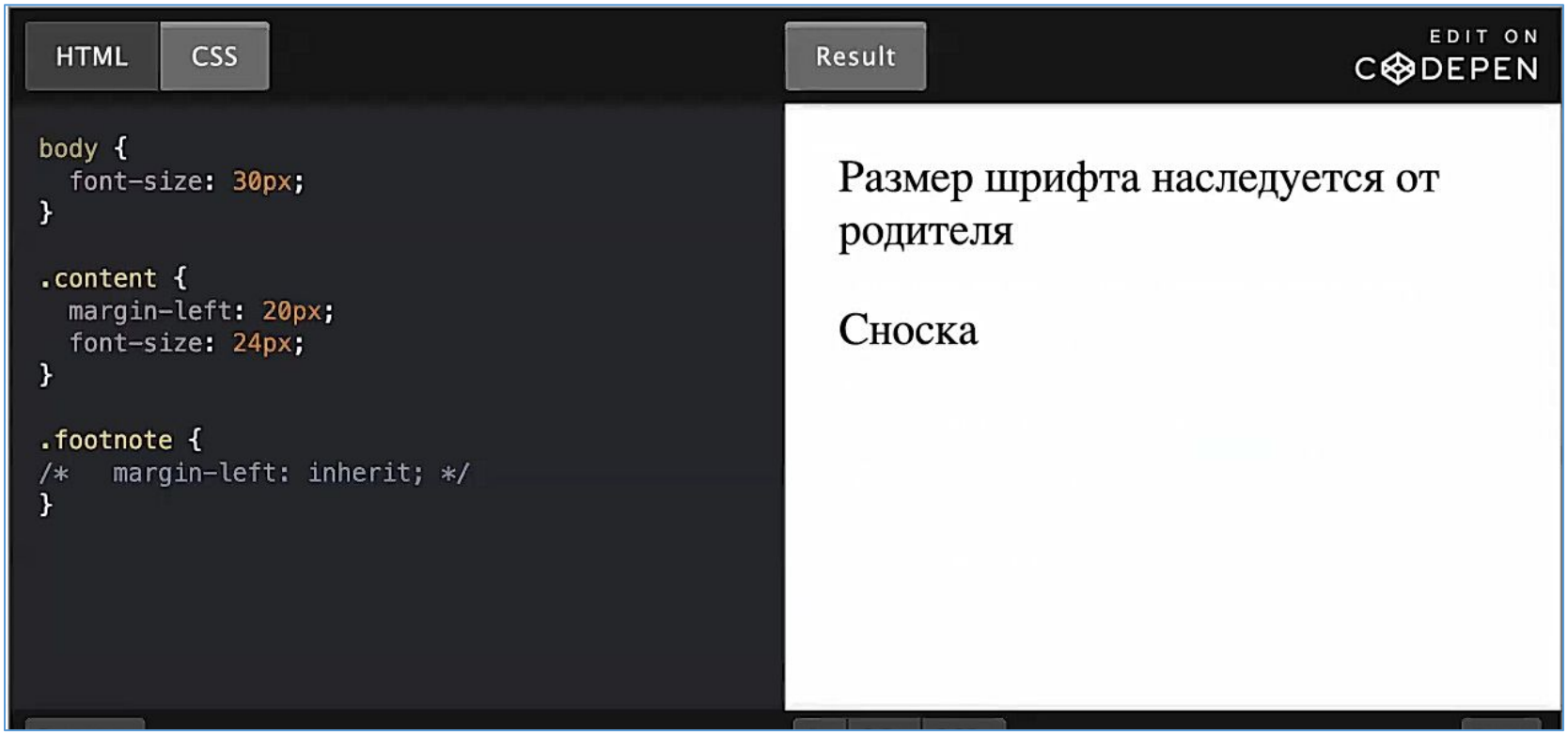
# ПСЕВДОЭЛЕМЕНТЫ

**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 ::.

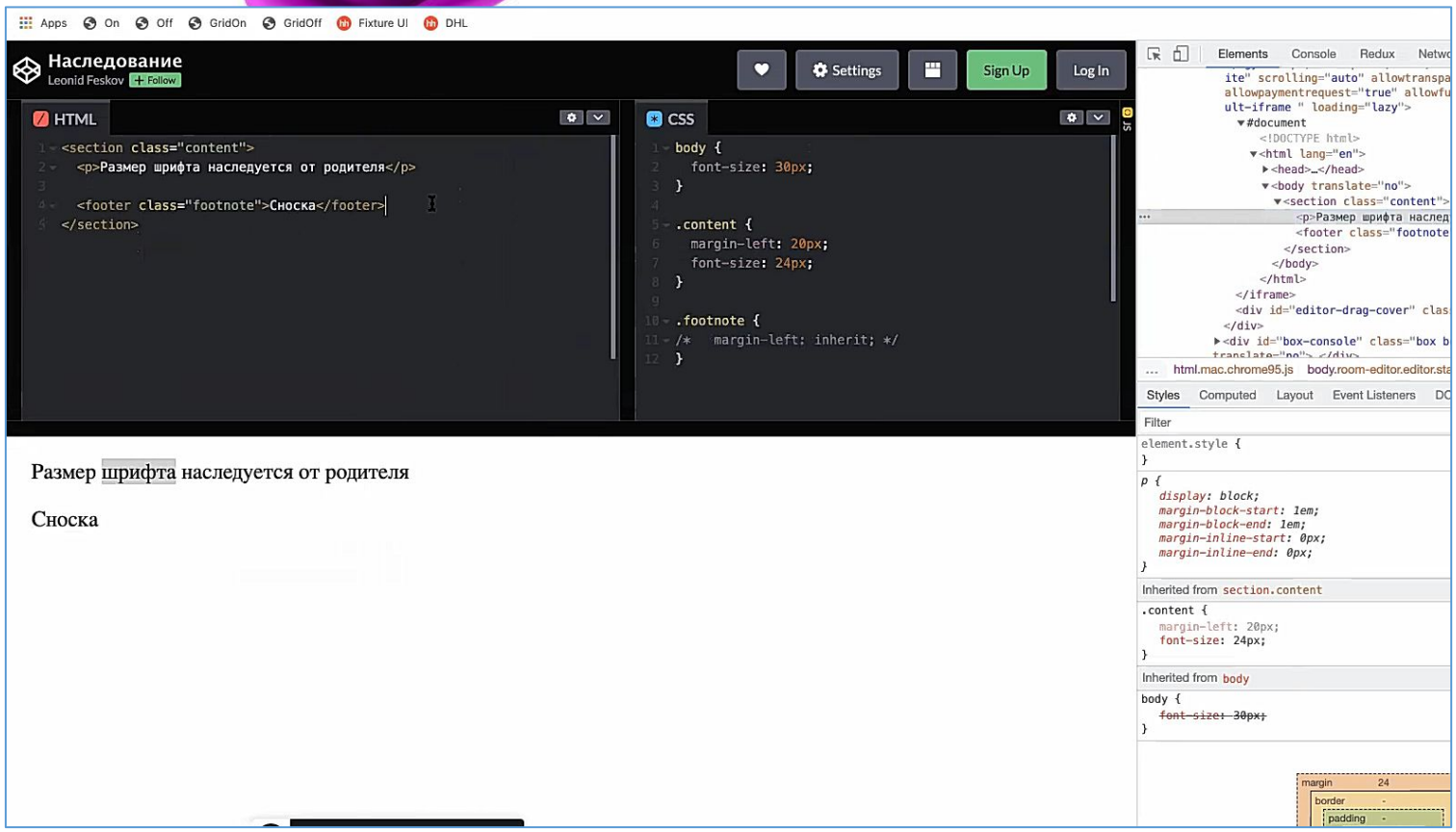
# ПСЕВДОЭЛЕМЕНТЫ

1. :active;
2. :link;
3. :focus;
4. :hover;
5. :visites.

# НАСЛЕДОВАНИЕ



# НАСЛЕДОВАНИЕ



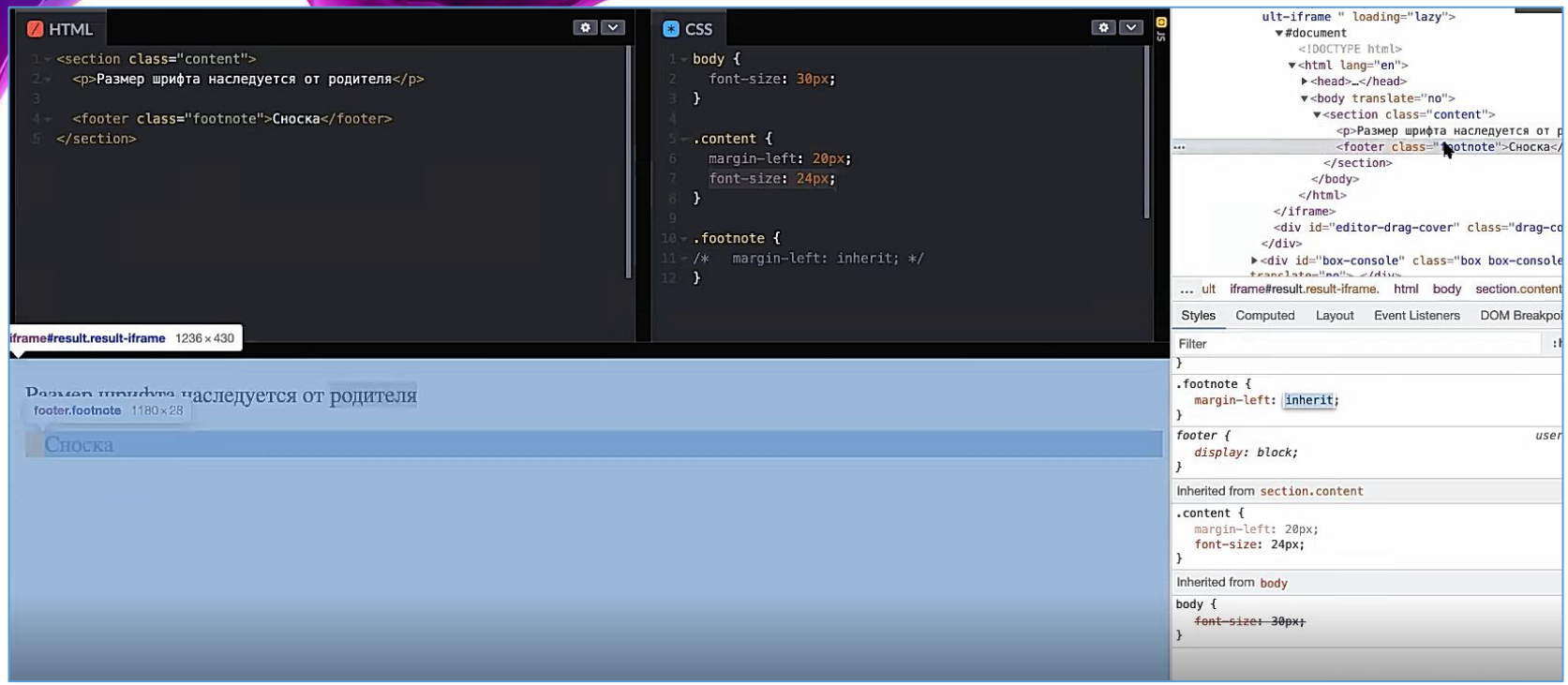

Во всех современных браузерах есть просмотр DOM дерева, инспектор HTML.

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

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

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

# НАСЛЕДОВАНИЕ




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
```

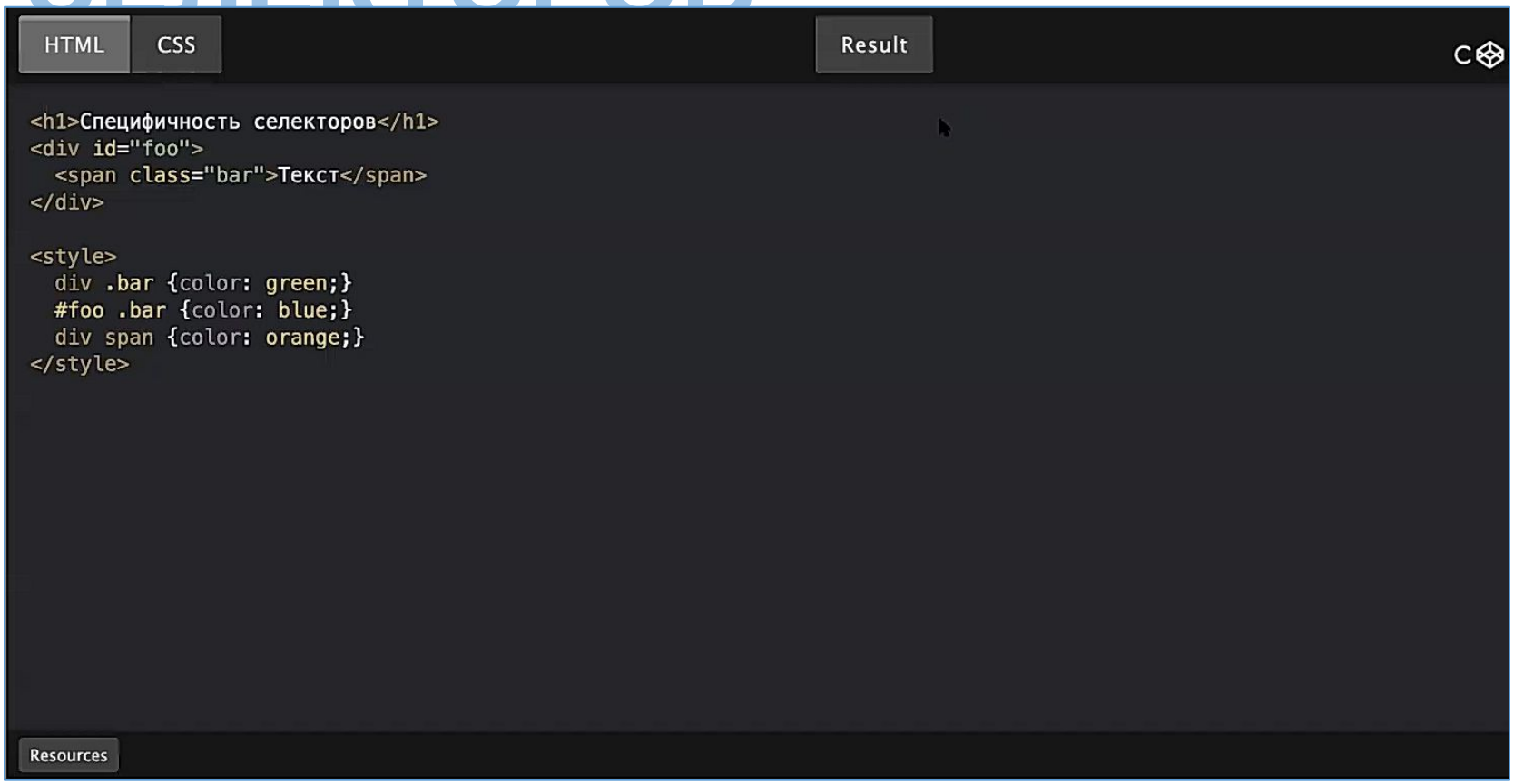
Это стили, которые относятся: к шрифту, к размеру, к цветам текста.

# 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

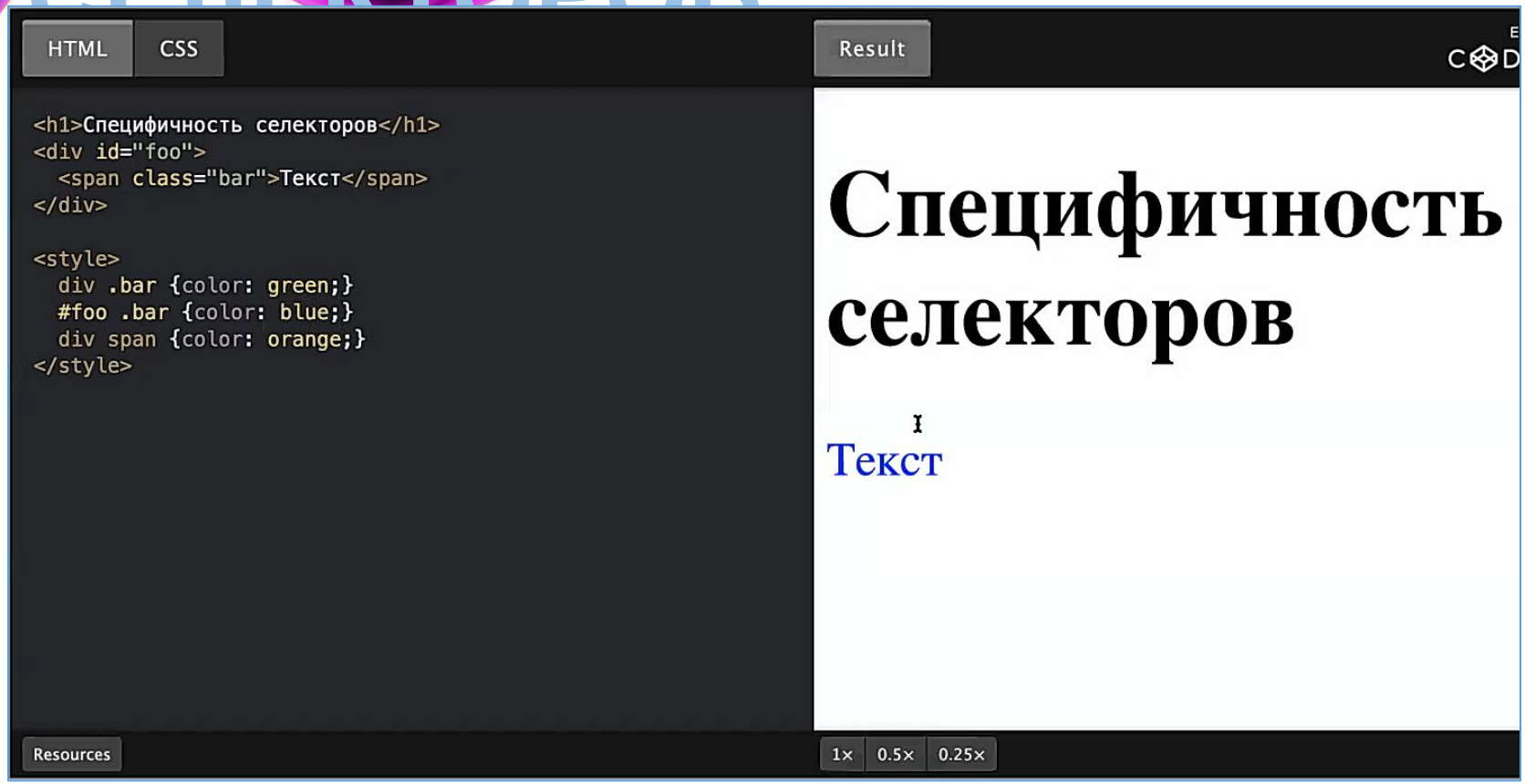
Это стили, которые относятся: к размеру, отступам, фоновые картинки.

# СПЕЦИФИЧНОСТЬ СЕЛЕКТОРОВ




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




Цвет текста синий, так как для определение 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.

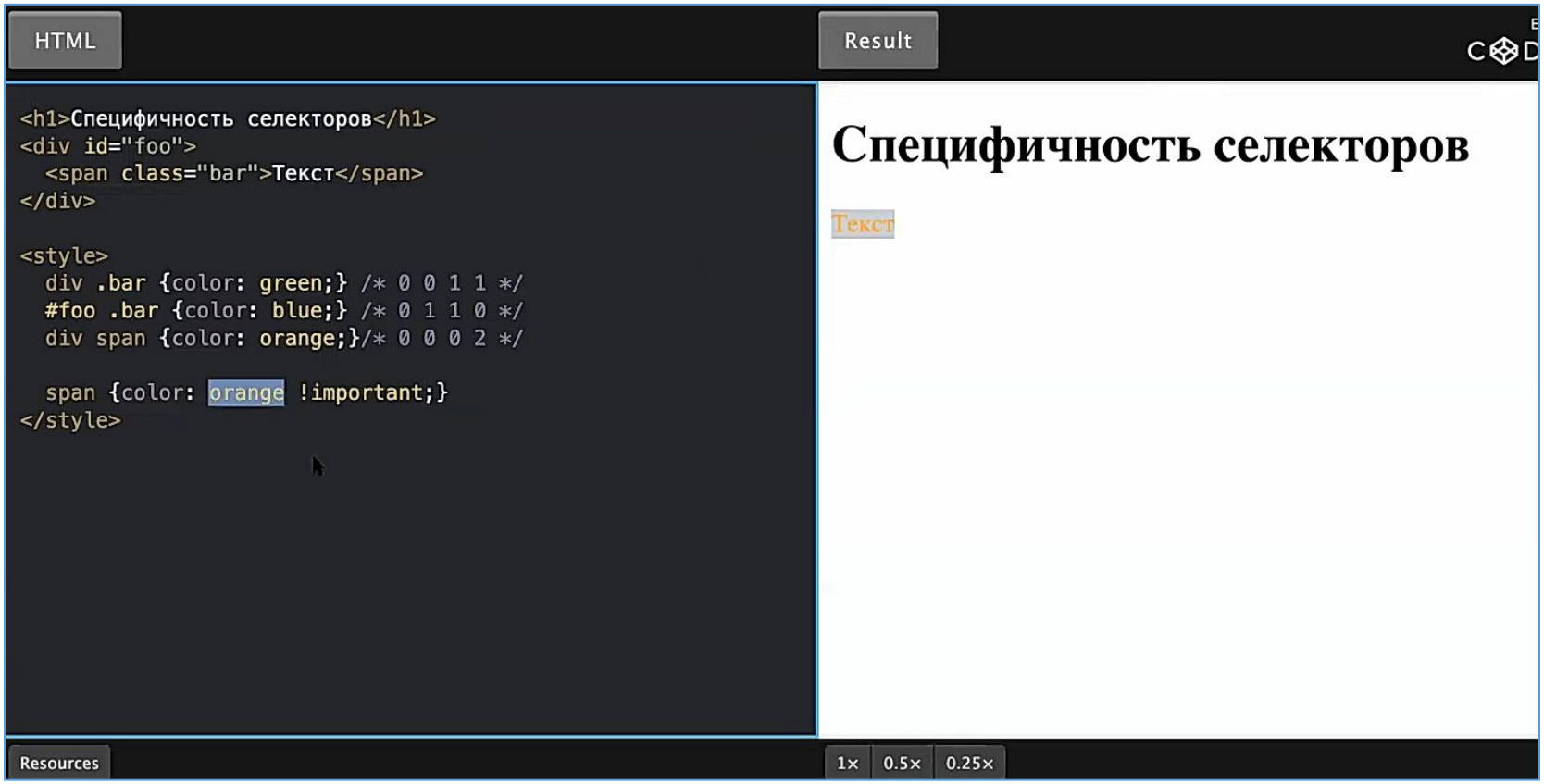
| Селектор | Специфичность |
| --- | --- |
| тег, псевдо-элемент(span, ::after) | 0 0 0 1 |
| класс, псевдокласс, атрибут (.header, :hover) | 0 0 1 0 |
| идентификатор (#footer) | 0 1 0 0 |
| инлайн-стили (style="color: red;") | 1 0 0 0 |

1000 – инлайн-стили добавляет единицу в 4 разряд.

# РАСЧЕТ СПЕЦИФИЧНОСТЬ

«#foo .bar» больше, чем другие селекторы.
Свойство «important» позволяет повысить специфичность.

# РАСЧЕТ СПЕЦИФИЧНОСТЬ




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

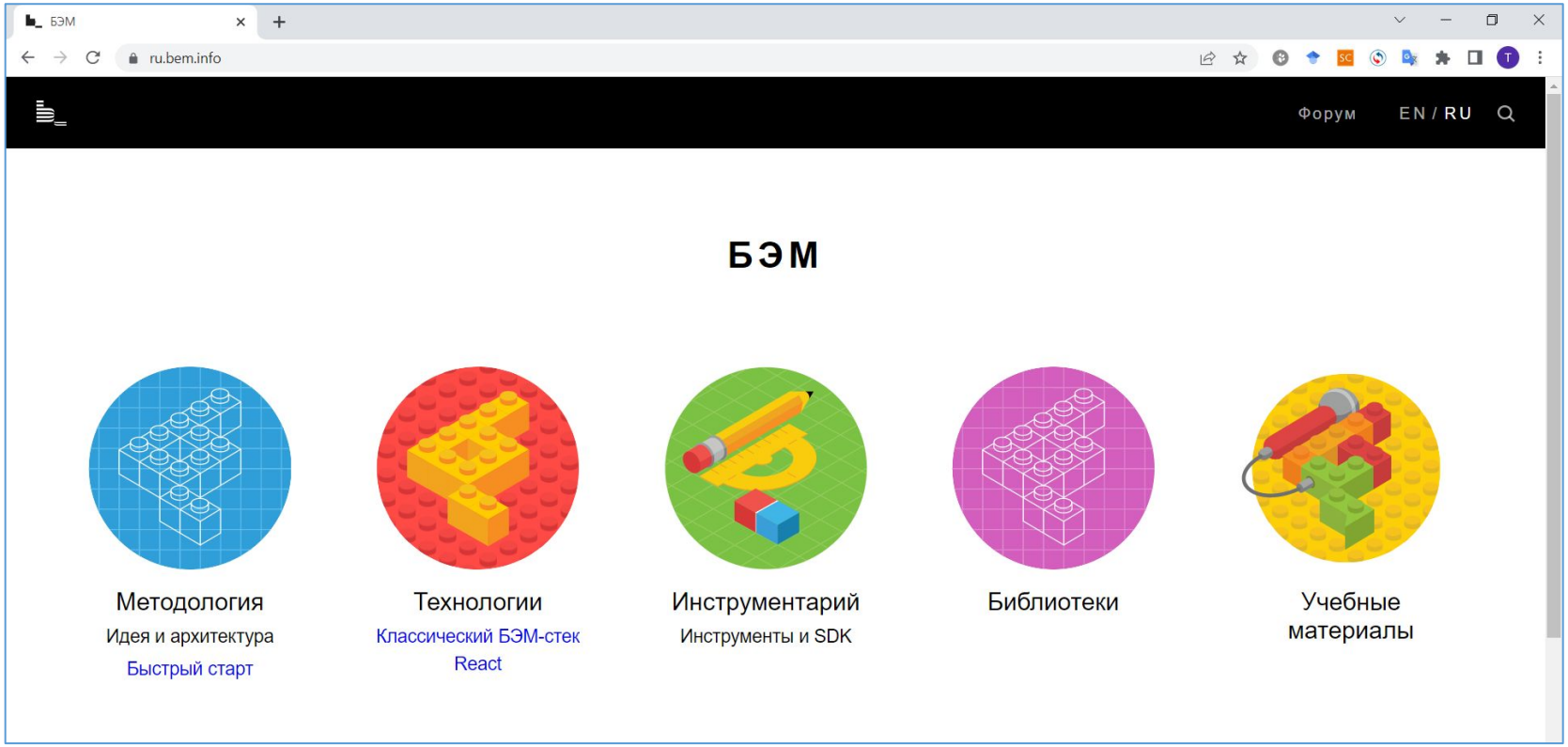
Селектор .text #name div.red.small { … }
Если захотите переиспользовать какие-то стили, которые описаны тут { … },
чтобы они использовались в другом месте вашего кода, то необходимо повторить структуру .text #name div.red.small.
Если переопределяете, то пишите более сложный селектор.

# 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 АТРИБУТОВ

1. Текстовые: "так" или 'так';
2. Размеры:
   - 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».