CSS Definition Syntax

Роман Дворнов

Москва, ноябрь 2019

- Работаю в Wrike
- Open source проекты:
  csstree, CSSO, discoveryjs, jora, hitext, basis.js, rempl и другие проекты

lahmatiy

rdvornov

Что делает доклад про CSS на JavaScript конференции?

Три причины

- Этот доклад про дизайн языка CSS, в котором есть интересные моменты для инженеров
- В области обработки CSS есть сложные проблемы, которые еще не решены
- Для решения проблем требуются алгоритмы, словари и код (JavaScript)

Quiz

Сколько значений цвета?

```
selector {
    font: 10px arial black;
    animation-name: red;
    whatever: green;
    width: blue;
    color: superpurple;
}
```

Столько значений цвета?

```
selector {
    font: 10px arial black;
    animation-name: red;
    whatever: green;
    width: blue;
    color: superpurple;
}
```

Часть имени шрифта "arial black"

Имя анимации

Неизвестное свойство

В width нельзя указывать цвет

Нет такого цвета

Ни одного!

Из чего состоит значение?

```
background: left 75% url(..) no-repeat #008800;
```

background-position

background-image

background-repeat

background-color

Что является именем анимации?

```
selector {
    animation: infinite forwards normal forwards;
    animation: reverse normal forwards;
    animation: "reverse" normal forwards;
    animation: reverse rotate none;
    animation: "reverse" none rotate;
}
```

Что является именем анимации?

```
selector {
    animation: infinite forwards normal forwards;
    animation: reverse normal forwards;
    animation: "reverse" normal forwards;
    animation: reverse rotate none;
    animation: "reverse" none rotate;
}
```

Вопрос сложный даже для экспертов 🤗

CSS is hard

Для многих вопросы сложные – это нормально

* Будучи «в теме», я сам допускал ошибки в своих докладах про CSS

Как понять как правильно?

Как понимают браузеры?

Как могут понять другие программы?

Источник правды?

CSS Definition Syntax

CSS Definition Syntax

язык для описания синтаксиса CSS

Знакомьтесь, CSS Definition Syntax

Name:	***background***
Value:	*<bg-layer>#* , *<final-bg-layer>*
Initial:	see individual properties
Applies to:	all elements
Inherited:	no
Percentages:	see individual properties
Media:	visual
Computed value:	see individual properties
Animatable:	see individual properties

Where

<bg-layer> = *<bg-image>* || *<position>* [/ *<bg-size>*]? || *<repeat-style>* || *<attachment>* || *<box>* || *<box>*

<final-bg-layer> = *<bg-image>* || *<position>* [/ *<bg-size>*]? || *<repeat-style>* || *<attachment>* || *<box>* || *<box>* || *<'background-color'>*

А где узнать об этом?

- До «CSS3» – фрагменты, разбросанные по спекам
- Полноценное описание начиная с модуля
  CSS Values and Units Module Level 3
  Самое актуальное в последнем Working Draft
- Редкие доклады, например, Типы данных в CSS
  София Валитова, FrontendConf'19
- В этом докладе, конечно ;)
  Материала не так много, в спеках не все очевидно, есть белые пятна

CSS Values and Units Module

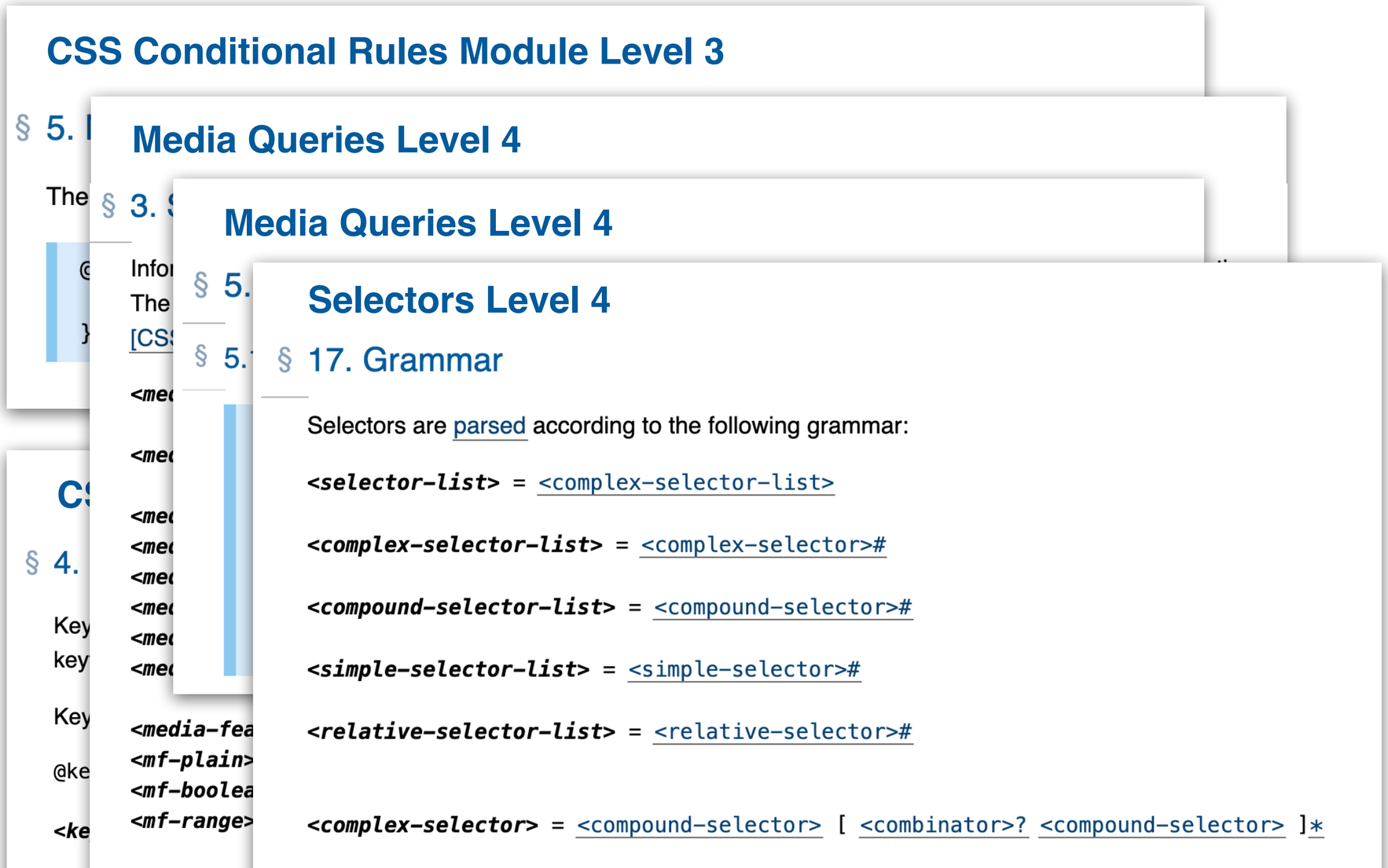
Value, а не CSS!

§ 2. Value Definition Syntax

The ***value definition syntax*** described here is used to define the set of valid values for CSS properties (and the valid syntax of many other parts of CSS). A value so described can have one or more components.

drafts.csswg.org/css-values/#value-defs

«Many other parts of CSS»



- At-rules
- At-rule preludes
- Media features
- Selectors
- …

CSS

~~Value~~ Definition Syntax

Таким название кажется более логичным

☝️ Не только для спек ☝️

* Об этом будет дальше

О чем поговорим?

- Разбор синтаксиса CSS Definition Syntax
- Области применения (кроме спек)
- Как научить программу с ним работать (техническое решение)
- Использование в практических задачах
- Текущие проблемы и будущее

Синтаксис

Важное отступление

- CSS Definition Syntax работает в терминах CSS токенов (не символов!)
- Пробелы и комментарии не учитываются (откидываются на этапе токенизации)

24 типа токенов в CSS

- <ident-token>
- <string-token>, <bad-string-token>, <number-token>, <hash-token>
- <colon-token>, <semicolon-token>, <comma-token>, <[-token>, <]-token>, <(-token>, <)-token>, <{-token>, <}-token>, <delim-token>
- <function-token>, <at-keyword-token>, <percentage-token>, <dimension-token>
- <url-token>, <bad-url-token>
- <whitespace-token>
- <CDO-token>, <CDC-token>

Это не так много

Подробнее про токены и механику

- В спеке CSS Syntax Module Level 3
- В докладе All about the CSS syntax

  Roman Dvornov, pitercss_conf 2018

Базовые компоненты

- Ключевые слова, например, `none` или `auto`
- Функциональные токены, например, `rgb(` или `calc(`
- Литералы: `/ , ( )`
- Экранированные литералы, например, `"+"` или `"-"`
- Базовые типы, например, `<length>` или `<percentage>`
- Свойства, например, `<'background'>`

Определения и ссылки

`<foo> = ...` определение

`<bar> = <foo>` использование

Виды ссылок

`<ident>` тип

`<'ident'>` свойство

`<ident()>` функция (не всегда выносятся)

Комбинаторы

- `A B` все элементы в заданном порядке
- `A | B` один из элементов (enum)
- `A || B` один или более элементов в любом порядке
- `A && B` все элементы в любом порядке
- `[ ]` группировка

Мультипликаторы

- `A?` ноль или одно вхождение (опциональный)
- `A*` ноль и более вхождений
- `A+` одно и более вхождений
- `A{N}` ровно **N** вхождений
- `A{N,M}` не менее **N** и не более **M** вхождений
- `#` одно и более вхождений разделенные запятой, можно ограничивать число вхождений, то есть `#{N}` или `#{N,M}`
- `[ ]!` группа не может производить пустое значение

Примеры

Свойство «color»

```
    <'color'> = <color>
      <color> = <rgb()> | <rgba()> | <hsl()> | <hsla()> | <hex-color> |
                <named-color> | currentcolor | <deprecated-system-color>
      <rgb()> = rgb( <number>#{3} )
     <rgba()> = rgb( <number>#{3} , <alpha-value> )
<alpha-value> = <number> | <percentage>
<named-color> = transparent | aliceblue | antiquewhite | aqua |
                aquamarine | azure | beige | bisque | black | ...
          ...
```

* Синтаксис упрощен для примера, полное описание

Свойство «border»

```
  <'border'> = <line-width> || <line-style> || <color>
<line-width> = <length> | thin | medium | thick
<line-style> = none | hidden | dotted | dashed | solid | double |
               groove | ridge | inset | outset
     <color> = ...
```

Свойство «border»

```
  <'border'> = <line-width> || <line-style> || <color>
<line-width> = <length> | thin | medium | thick
<line-style> = none | hidden | dotted | dashed | solid | double |
               groove | ridge | inset | outset
     <color> = ...
```

Пример валидных значений для border

```
1px solid red
1px red solid
solid 1px red
solid red 1px
red 1px solid
red solid 1px
```

```
1px solid
 1px red
solid 1px
solid red
 red 1px
red solid
```

```
 1px
solid
 red
```

На что похоже?

Ближайшие аналоги

CSS Definition Syntax	RegExp	*BNF
Токены	–	Да
Определения и ссылки	–	Да
Комбинаторы	только |	только |
Мультипликаторы	Да, за исключением #	Аналоги ? и *
Группировка	Да ()	Да ()

В основу взяли *BNF и расширили под домен языка

* Достоверно не известно, на основе «слухов»

BNF

BNF – Backus Naur form

Формальная система … для описания синтаксиса языков программирования, данных, протоколов и т.д.

en.wikipedia.org/wiki/Backus–Naur_form

Пример BNF

```
<postal-address> ::= <name-part> <street-address> <zip-part>

     <name-part> ::= <personal-part> <last-name> <opt-suffix-part> <EOL>
                   | <personal-part> <name-part>

 <personal-part> ::= <initial> "." | <first-name>

<street-address> ::= <house-num> <street-name> <opt-apt-num> <EOL>

      <zip-part> ::= <town-name> "," <state-code> <ZIP-code> <EOL>

<opt-suffix-part> ::= "Sr." | "Jr." | <roman-numeral> | ""
    <opt-apt-num> ::= <apt-num> | ""
```

Пример CSS Definition Syntax

```
<postal-address>     = <name-part> <street-address> <zip-part>

     <name-part>     = <personal-part> <last-name> <opt-suffix-part>? <EOL>
                     | <personal-part> <name-part>

  <personal-part>    = <initial> "." | <first-name>

 <street-address>    = <house-num> <street-name> <opt-apt-num>? <EOL>

       <zip-part>    = <town-name> , <state-code> <ZIP-code> <EOL>

<opt-suffix-part>    = "Sr." | "Jr." | <roman-numeral>
    <opt-apt-num>    = <apt-num>
```

Особые правила
CSS Definition Syntax

«Implicit omissible comma»

Пример определения

```
a? , b?
```

Valid input

```
a, b
a
b
<empty>
```

Invalid input

```
a ,, b
a ,
, b
,
```

«Implicit omissible comma»

Пример определения

```
rgb( <number>#{3} , <alpha-value>? )
```

Valid input

```
rgb(1, 2, 3)
rgb(1, 2, 3, .5)
```

Invalid input

```
rgb(1, 2, 3, )
```

&& и || комбинаторы

Пример определения

```
<line-width> || <line-style> || <color>
```

Valid input

```
1px solid red
solid red 1px
green 1px
dotted
...
```

Invalid input

```
1px solid red 2px
red green
<empty>
...
```

Определения с пониженным приоритетом

`<custom-ident>`

§ 4.2. Author-defined Identifiers: the <custom-ident> type

When parsing positionally-ambiguous keywords in a property value, a <custom-ident> production ***can only claim the keyword if no other unfulfilled production can claim it***.

drafts.csswg.org/css-values-4/#identifier-value

Определения с пониженным приоритетом

Matching <'list-style'> incorrectly due to <custom-ident> #101

Edit | New issue

Closed **lmgonzalves** opened this issue on 9 Jul · 4 comments

lmgonzalves commented on 9 Jul

First of all, great work here! Thanks!

I'm having an issue matching the property `<'list-style'>` with value `inside`.
The token `inside` should be matched as `<'list-style-position'>`, but it is matched as `<'list-style-type'>` instead. See:

Match result 35 iteration(s)

inside
<custom-ident>
<counter-style-name>
<counter-style>
<'list-style-type'>
<'list-style'>

Formal syntax

```
<'list-style-type'> || <'list-style-position'> || <'list-style-image'>
```

This happens because `<'list-style-type'>` could be a `<custom-ident>`, so it will match the `<'list-style-position'>` tokens too (`inside` | `outside`).

Assignees

No one—assign yourself

Labels

bug

lexer

No milestone

Notifications Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

2 participants

Lock conversation

Pin issue

Определения с пониженным приоритетом

<custom-ident>

```
.selector {
    list-style: foo url(..);
}
```

<'list-style-type'>

<custom-ident>

```
.selector {
    list-style: inside url(..);
}
```

<'list-style-position'>

keyword

Определения с пониженным приоритетом

<length>

§ 6. Distance Units: the <length> type

For zero lengths the unit identifier is optional (i.e. can be syntactically represented as the <number> 0).
However, **if a 0 could be parsed as either a <number> or a <length> in a property (such as line-height), it must parse as a <number>.**

drafts.csswg.org/css-values-4/#length-value

Определения с пониженным приоритетом

zero unitless <length>

```
.selector {
    flex: 0 0px;
}
```

<'flex-basis'>

<length>

```
.selector {
    flex: 0 0;
}
```

<'flex-shrink'>

<number>

Определения с пониженным приоритетом

zero unitless <length>

lahmatiy commented on 10 Jul • edited

Author Contributor

@squidfunk Using zero as a unitless value is allowed for `<length>` only (it covers any `<length>` production, e.g. `<length-percentage>`):

> For zero lengths the unit identifier is optional (i.e. can be syntactically represented as the `<number>` `0`).

At one time a unitless zero was valid as `<angle>` value in some syntaxes. However, currently unitless zero as `<angle>` is invalid in general, but valid in some syntaxes due to legacy reasons (spec):

> Note: For legacy reasons, some uses of `<angle>` allow a bare `0` to mean `0deg`. This is not true in general, however, and will not occur in future uses of the `<angle>` type.

`<zero>` introduced to indicate explicitly where it allowed. It appears in Values & Units L4 (it was

github.com/mdn/data/pull/376#issuecomment-510108141

evaluate to a `<number>` with the value `0` (for example, `calc(0)`) do not match `<zero>`; only

Высокоуровневые определения

Определяют общую структуру

```
<stylesheet>
<rule-list>
<declaration-list>
<declaration-value>
<any-value>
```

www.w3.org/TR/css-syntax-3/#rule-defs

Высокоуровневые определения

```
selector {
  --bg-color: , green;
  background: url(...) var(--bg-color,, red);
}
```

Высокоуровневые определения

```
selector {
  --bg-color: , green;
  background: url(...) var(--bg-color,, red);
}
```

Высокоуровневые определения

```
<'--*'> = <declaration-value>?

<var()> = var( <custom-property-name> , <declaration-value>? )
```

Высокоуровневые определения

```
<'--*'> = <declaration-value>?

<var()> = var( <custom-property-name> , <declaration-value>? )
```

§ 8.2. Defining Arbitrary Contents: the <declaration-value> and <any-value> productions

The <declaration-value> production matches **any sequence of one or more tokens**, so long as the sequence does not contain <bad-string-token>, <bad-url-token>, unmatched <)-token>, <]-token>, or <}-token>, or top-level <semicolon-token> tokens or <delim-token> tokens with a value of "!".

www.w3.org/TR/css-syntax-3/#any-value

Высокоуровневые определения

```
<'--*'> = <declaration-value>?

<var()> = var( <custom-property-name> , <declaration-value>? )


selector {
  --name: ,green;                /* OK */
  foo: var(--name,,red);         /* OK */
  bar: var(--name,,);            /* OK */
  baz: var(--name,);             /* не OK */
  qux: var(--name, );            /* OK, есть <whitespace-token> */
  yyy: var(--name);              /* OK, implicit omissible comma */
}
```

Высокоуровневые определения

Имеют особые правила обработки ошибок

(error handling)

www.w3.org/TR/css-syntax-3/#error-handling

CSS?

```
zMN_lc]4U~N,p<l\5mKZZ1<h{3\UZ~;RE8E;int_k\U~Kdsw!U$-ESTI~$Z.wlvZL}.foo:nth-
child(2n+1){!UDcOt%A&HHch;f8vcb@e<F;^U.D22QF;4e-y|l1XUf*a\EhRL+@;color:red!
important;|M0$RE_N?YpfN%nAu\rL$tKN9^^;border:1px solid#abc;!aUV;KPx8WNUt6^!
_MV&lYYF_H?/jbWT~PRoo!I#~5I=g;width:200px;hTkF.%s2!VMtEa_v~~`KKMB2hIc`Pniy-
Q5Y4yImvDPJxG<gP\=}$fF$!DJ!k?&6Kt?!uD72CjO#-xo8S38Y?$%F3^s9bdc81&3e4nklsd*2
```

Да, это CSS

```
zMN_lc]4U~N,p<l\5mKZZ1<h{3\UZ~;RE8E;int_k\U~Kdsw!U$-ESTI~$Z.wlvZL}.foo:nth-
child(2n+1){!UDcOt%A&HHch;f8vcb@e<F;^U.D22QF;4e-y|l1XUf*a\EhRL+@;color:red!
important;|M0$RE_N?YpfN%nAu\rL$tKN9^^;border:1px solid#abc;!aUV;KPx8WNUt6^!
_MV&lYYF_H?/jbWT~PRoo!I#~5I=g;width:200px;hTkF.%s2!VMtEa_v~~`KKMB2hIc`Pniy-
Q5Y4yImvDPJxG<gP\=}$fF$!DJ!k?&6Kt?!uD72CjO#-xo8S38Y?$%F3^s9bdc81&3e4nklsd*2
```

↓

```
.foo:nth-child(2n + 1) {
    color: red !important;
    border: 1px solid #abc;
    width: 200px;
}
```

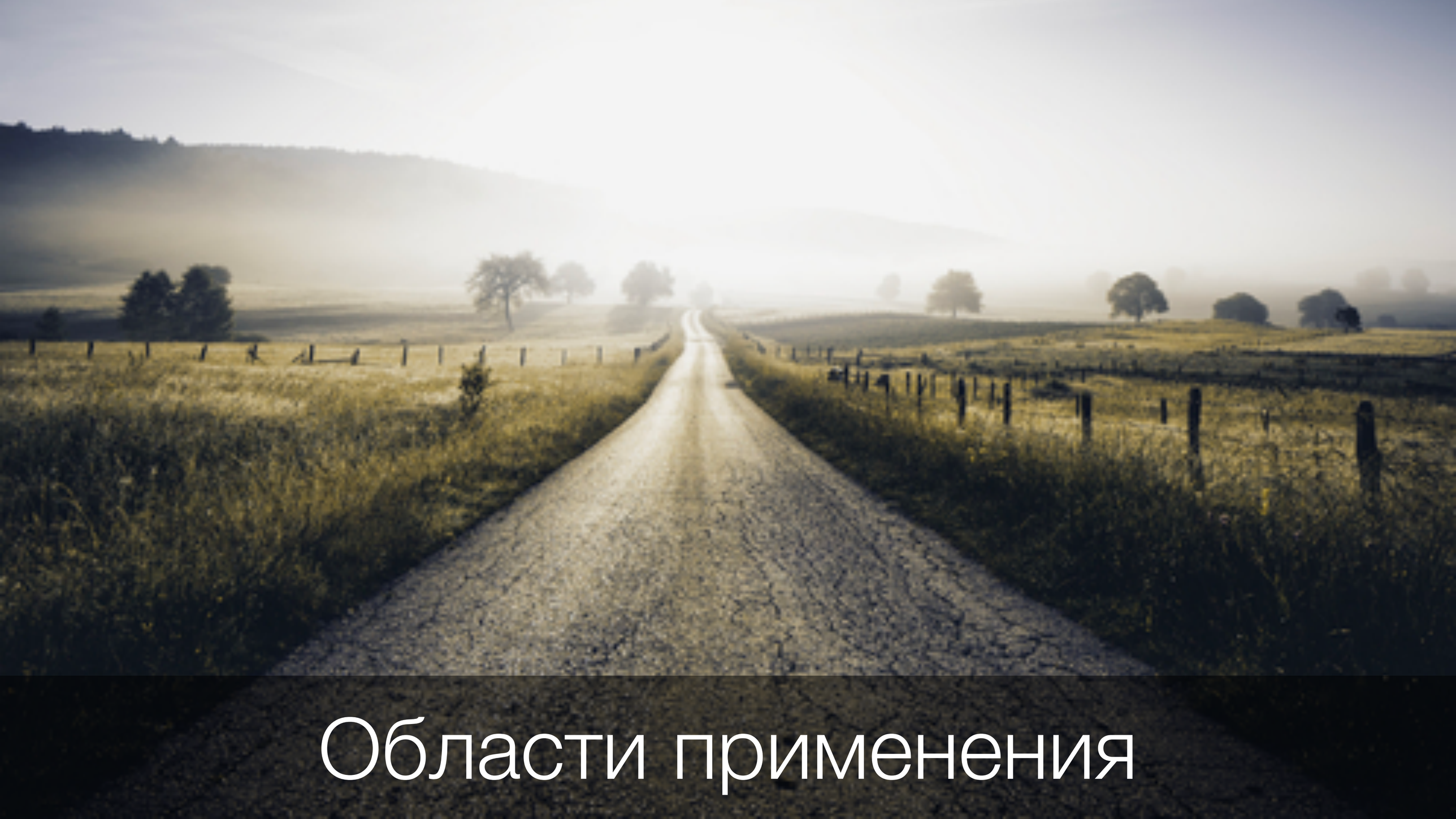
Благодаря правилам обработки ошибок, CSS парсер (браузер) выберет то, что может понять. Может таким будет «CSS10»?

Может казаться что с CSS что-то не так, но если знать механику языка все встает на свои места

Промежуточный итог

CSS Definition Syntax

- Имеет общее с другими подобными языками
- Не сложно выучить
  (особенно если знать RegExp и/или BNF)
- Сложнее:
  - Думать в терминах CSS токенов (непривычный словарь)
  - Учитывать специфичные правила



Области применения

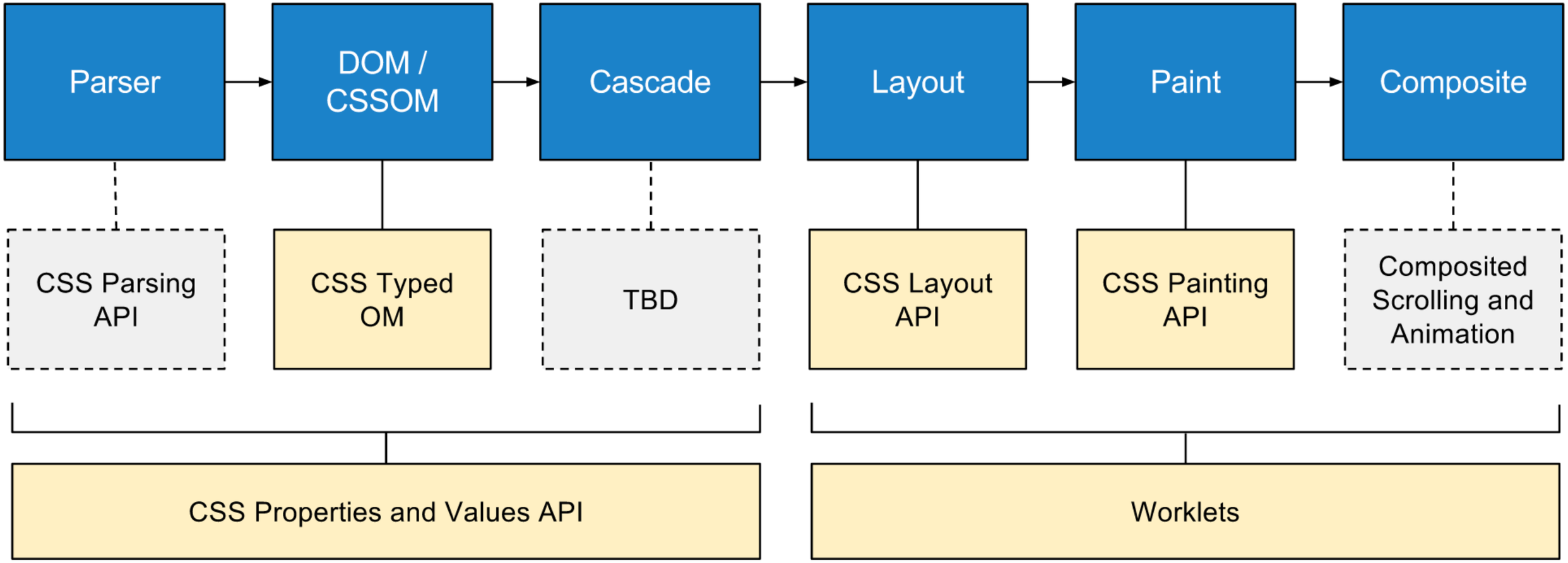
Области применения

- Для написания спецификаций
- CSS Houdini
- Задачи разработчиков?

CSS Houdini

Houdini

Расширение CSS с помощью JavaScript



Custom Property

```javascript
// JavaScript
CSS.registerProperty({
    name: '--sidebar-width',
    syntax: '<length>',
    inherits: true,
    initialValue: '80px'
});
```

```css
/* CSS */
@property --sidebar-width {
    syntax: '<length>';
    inherits: true;
    initial-value: 80px;
}
```

Нужно для анимации (value interpolation) и валидации

CSS Properties and Values API Level 1

Paint Worklet

```
// CSS
.example {
    background-image: paint(demo, red, 1px);
}

// PaintWorklet
registerPaint('demo', class {
    static get inputArguments() { return ['<color>', '<length>']; }
    paint(ctx, size, styleMap, args) { /* paint code here */ }
});
```

CSS Painting API Level 1

Custom Functions

```
CSS.registerFunction({
    name: '--darken',
    inputArguments: ['<color>', '<percentage>'],
}, (color, percent) => {
    const newColor = color.toHSL();
    newColor.lightness *= (1 - percent.value/100);
    return newColor;
});

.element {
    color: --darken(tomato, 12%);
}
```

www.xanthir.com/talks/2019-05-15/

Используется подмножество

CSS Properties and Values API Level 1

§ 5. Syntax Strings

Встроенные определения: <length>, <number>, <percentage>, <length-percentage>, <color>, <image>, <url>, <integer>, <angle>, <time>, <resolution>, <transform-function>, <custom-ident>, <transform-list>

Комбинатор |

Мультипликаторы + и #

Задачи разработчиков

Мой путь в эту тему

- Работа над CSSO (2015)
- Много вопросов «как?»:
  - Безопасная ли трансформация?
  - Из каких частей состоит значение (shorthand свойства, аргументы функции)
  - Упростить добавление новых свойств и конструкций
  - ...

* CSSO – CSS минификатор

Решение:

CSS Definition Syntax

Но как?

Решения?

- Готовые библиотеки

  Отсутствуют 😞

- Браузеры

  Хардкодят – но для них OK

- Другие проекты работающие с CSS

  RegExp или хардкод, не учитывают нюансы

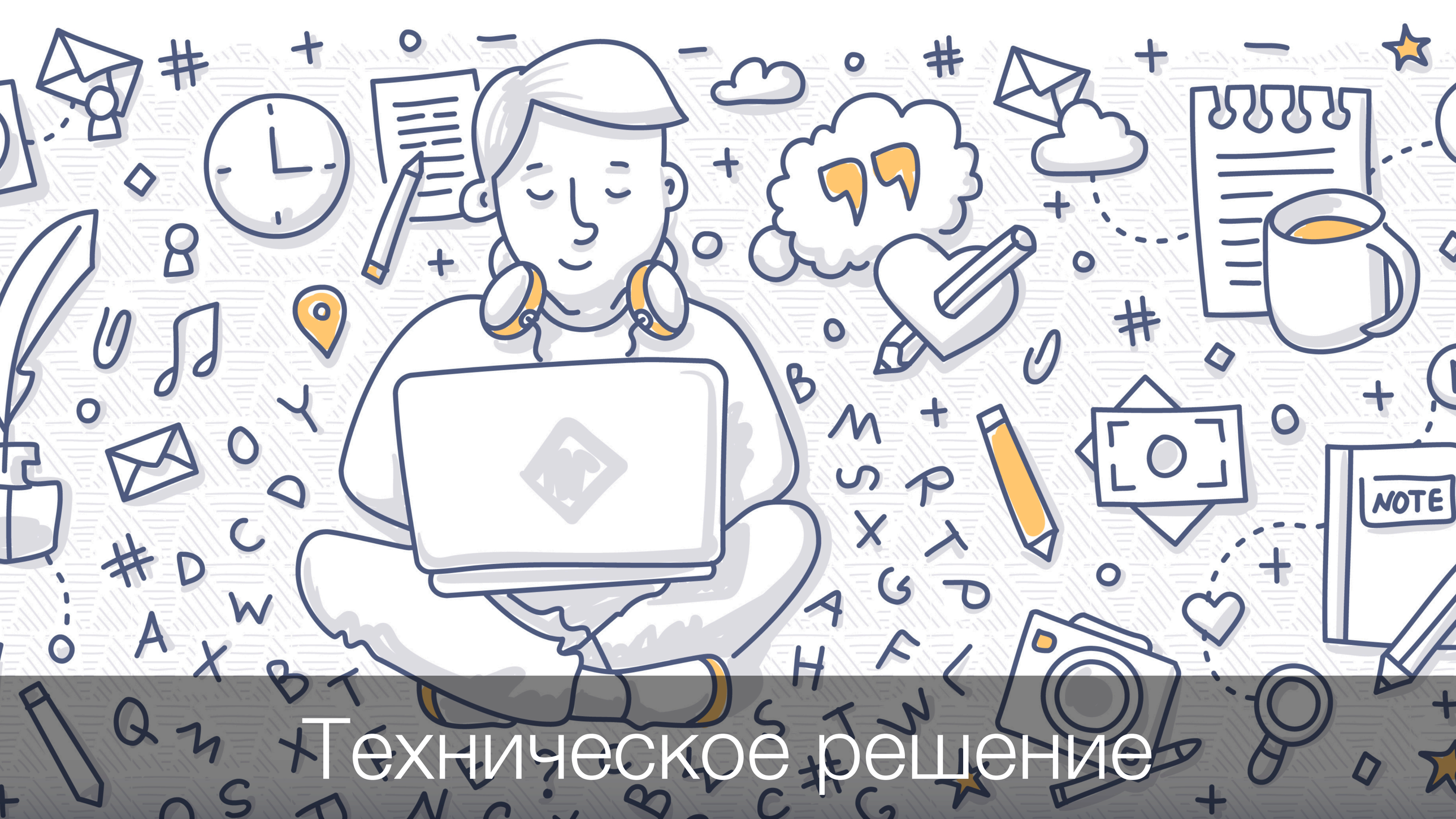
Но представим, что есть решение для работы с CSS Definition Syntax...

Какие задачи решаются

- Более надежная трансформация CSS
- Валидация CSS
- Поиск, например, цвета или имена шрифтов
- Подсказки (в редакторах)
- ...

Нет решения – сделай сам!

Так появился CSSTree, CSS парсер
с поддержкой CSS Definition Syntax



Техническое решение

Что нужно?

- Разобрать определения на составные части
  string -> AST
- Сопоставить определение со значением
  матчинг
- Высокоуровневое API для работы с результатом матчинга

AST

String → AST

```
<rgb()> = rgb( <number>#{3} )
```

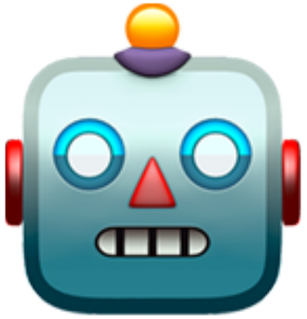
```
{
    "type": "Group",
    "terms": [
        {
            "type": "Function",
            "name": "rgb"
        },
        {
            "type": "Multiplier",
            "comma": true,
            "min": 3,
            "max": 3,
            "term": {
                "type": "Type",
                "name": "number",
                "opts": null
            }
        },
        {
            "type": "Token",
            "value": ")"
```

Вызовы

- Нет грамматики описывающей синтаксис CSS Definition Syntax 🤗
- Нет примера формата AST 🤗
- Словарь определений был заимствован у MDN, но он оказался с ошибками и неполный

Про словарь

- Mozilla Template:CSSData – был служебным шаблоном в недрах MDN
- 17 сентября 2016 анонсирован CSSTree с CSS Definition Syntax на FrontTalks
- 20 сентября 2016 завязалась дискуссия относительно словарей
- Призвали ребят из MDN
- Пропушили их вынести "CSSData" на github
- 19 октября 2016 появился репозиторий mdn/data
- Перенесли большую часть фиксов из CSSTree
- 22 июня 2017 в npm опубликован пакет mdn-data 1.0



Так появился словарь по синтаксису CSS для машин

Сегодня в него контрибьютят представили MDN, разработчики браузеров, авторы спек и все кому не все равно ;)

Словарь все еще не полный

CSS свойств

- mdn/data – **493**
- известных свойств – **1150+**

Чем можно помочь:

- найти неописанное свойство
- нагуглить описание
- сделать PR с описанием в JSON

CSS syntax reference

CSSTree syntax reference

https://csstree.github.io/docs/syntax/

Filter (regexp)

Make report

Property 567

<'--*'>
<'-moz-appearance'>
<'-moz-background-clip'> ADDED
<'-moz-binding'>
<'-moz-border-bottom-colors'>
<'-moz-border-left-colors'>
<'-moz-border-radius-bottomleft'> ADDED
<'-moz-border-radius-bottomright'> ADDED
<'-moz-border-radius-topleft'> ADDED
<'-moz-border-radius-topright'> ADDED
<'-moz-border-right-colors'>
<'-moz-border-top-colors'>
<'-moz-context-properties'>
<'-moz-float-edge'>
<'-moz-force-broken-image-icon'>
<'-moz-image-region'>
<'-moz-orient'>
<'-moz-osx-font-smoothing'> ADDED
<'-moz-outline-radius'>
<'-moz-outline-radius-bottomleft'>

CSS syntax reference (CSSTree & mdn/data)

css-tree	1.0.0-alpha.37
mdn-data	2.0.4

567	350	98	0
Properties	Types	Functions	Problems

Reports

- Syntax stat
- CSSTree difference over mdn-data
- mdn-data syntax code style problems
- Syntaxes used a string in definition

csstree.github.io/docs/syntax

Индекс определений в W3C спеках

CSSWG drafts index

csstree.github.io/csswg-drafts-index/

By Spec | Index

Filter (regexp)

CSS Animations Level 1 9

animation prop
animation-delay prop
animation-direction prop
animation-duration prop
animation-fill-mode prop
animation-iteration-count prop
animation-name prop
animation-play-state prop
animation-timing-function prop

CSS Animations Level 2 1

animation-composition prop

CSS Backgrounds and Borders Module Level 3 41

background prop
background-attachment prop
background-clip prop
background-color prop

Make report

CSSWG drafts index

Source: w3c/csswg-drafts commit d3ef46d870 on 2019-09-09T04:01:40.000Z

97	117	571	256	3
Specs	IDL sections	Definitions	Productions	Problem syntaxes

- Specs and statuses
- Problem syntaxes
- Missed productions
- IDL sections by spec

csstree.github.io/csswg-drafts-index/

Мопед не мой, я просто пропарсил определения 😳

Matching

Всё* уже придумано за нас

* Почти

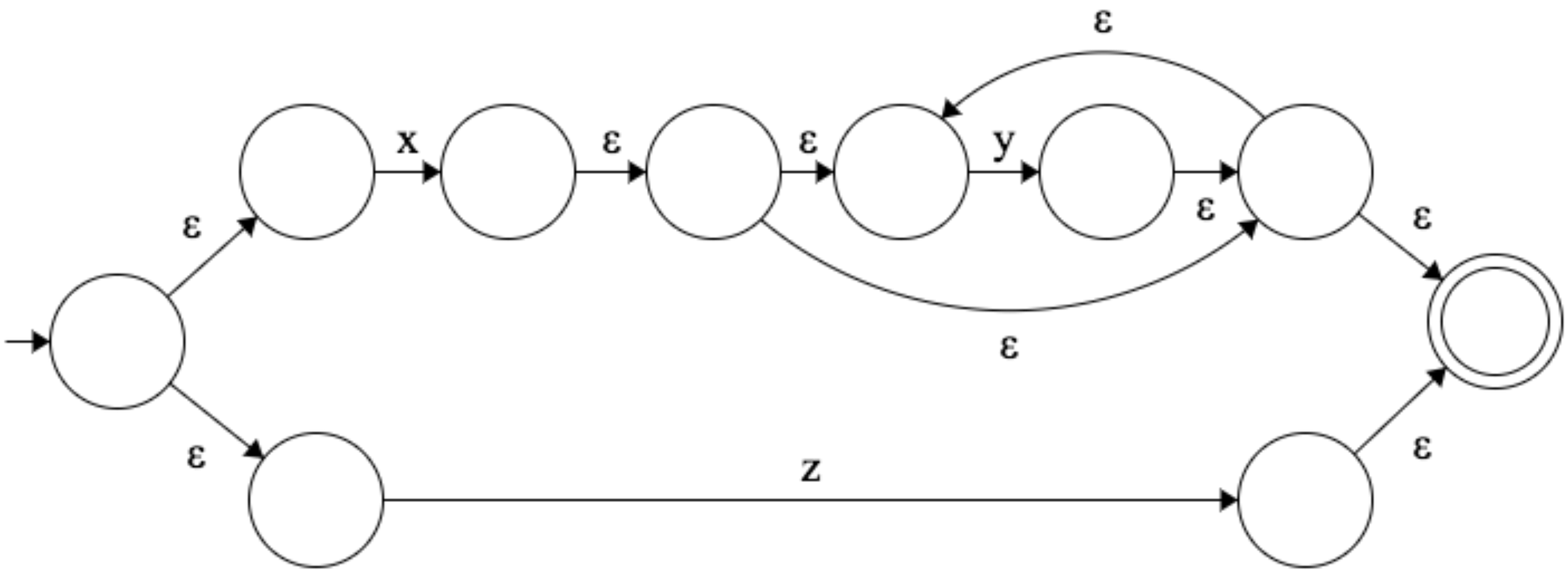
По мотивам RegExp

- Building a RegExp machine – Dmitry Soshnikov
- Regex – Alexander Grebenyuk

Рецепт

- AST → match graph (граф состояний и переходов)
- NFA / DFA ([не]детерминир. конечный автомат)
- DFS / Backtracking (поиск в глубину с восстановлением пути)
- ~~Подпорки и костыли~~ особые правила

x y* | z

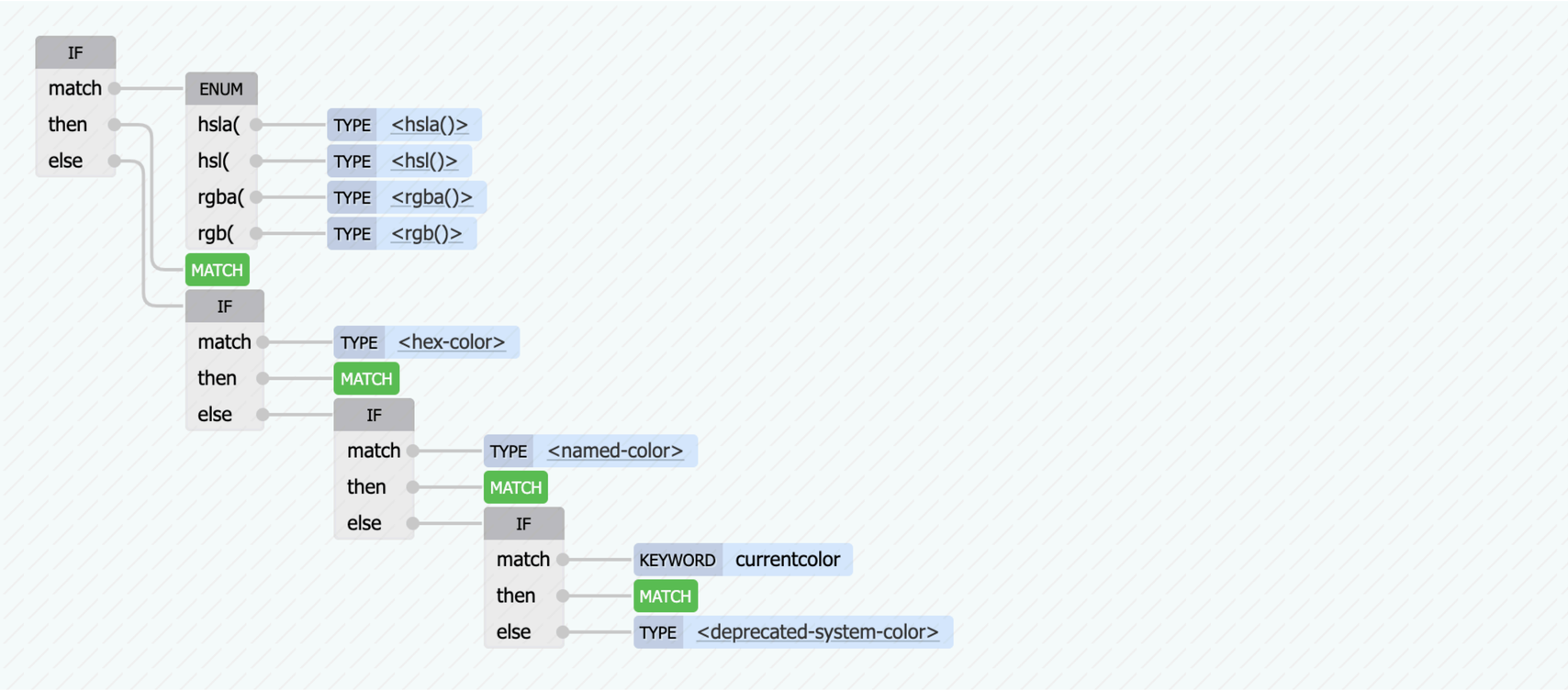


Вид в профиль

<color>

```
<rgb()> | <rgba()> | <hsl()> | <hsla()> | <hex-color> | <named-color> | currentcolor | <deprecated-system-color>
```

Match graph 14 node(s)



Здесь должна быть часть со множеством технических деталей, но время доклада ограничено...

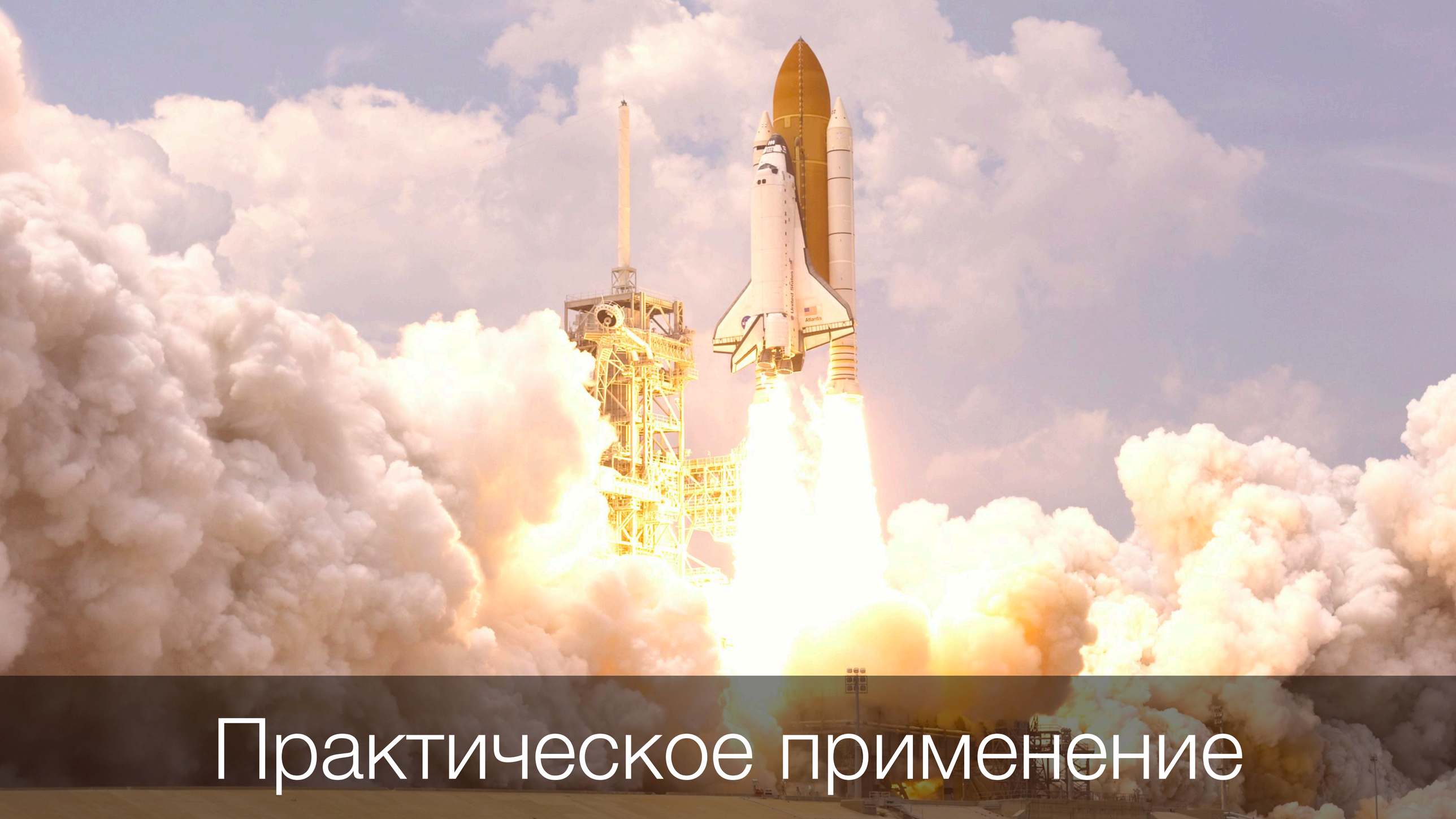
Про основные моменты можно прочитать в release notes CSSTree

Проблема решена?

Вызовы

- Производительность (можно лучше)
- Ограничения*:
  - Не все специальные правила учитываются
  - Не поддерживается `var()`
  - Не вычисляется `calc()`
  - Не применяется ко всей таблице стилей
  - ...

* На данный момент, но в планах исправить...



Практическое применение

Решаемые задачи

- Надежная трансформация CSS
- Валидация CSS
- Поиск
- Расширение синтаксиса CSS или его подмена
- ...

Надежность трансформаций

«Stack trace» для значений

<'background'>

Test a value:

top 100% url(..) no-repeat #008800

Match:

top	100%	url(..)	no-repeat	#008800

Match trace (click to pin trace block)

<percentage>

<length> | <percentage>

[center && [left | right | top | bottom] <length-percentage>?] | [[left | right] <length-percentage>?] && [[top | bottom] <length-percentage>?] | [[left | center | right | <length-percentage>] || [top | center | bottom | <length-percentage>]]

<bg-image> || <position> [/ <bg-size>]? || <repeat-style> || <attachment> || <box> || <box> || <'background-color'>

[<bg-layer> ,]* <final-bg-layer>

Опыт CSSO: минификация цвета

- Ищем идентификаторы
- Если идентификатор из словаря – заменяем на короткую запись, например, `black` -> `#000`

Вспомним пример

```
selector {
    font: 10px arial black;
}
```

Часть имени шрифта
"arial black"

Опыт CSSO: минификация цвета

- Ищем идентификаторы
- Если идентификатор из словаря – заменяем на короткую запись, например, `black` -> `#000`
- Черный список свойств*, в которых не делаем замен (`font`, `font-family`, ...)

* Так делают все минификаторы CSS

CSS Definition Syntax API

```
csstree.lexer
    .matchDeclaration(declarationNode)
    .isType(ident, 'color')
```

Используется в CSSO при минификации значений цвета

Опыт CSSO: минификация цвета

- Ищем идентификаторы
- Если идентификатор из словаря – заменяем на короткую запись, например, `black` -> `#000`
- ~~Черный список свойств, в которых не делаем замен (`font`, `font-family`, ...)~~

Опыт CSSO: unitless zero length

- Ищем `0px` / `0%`
- Удаляем `px` / `%`

Вспоминаем пример

zero unitless <length>

```
.selector {
    flex: 0 0px;
}
```

<'flex-basis'>

<length>

```
.selector {
    flex: 0 0;
}
```

<'flex-shrink'>

<number>

Опыт CSSO: unitless zero length

- Ищем `0px` / `0%`
- Удаляем `px` / `%`
- Черный список свойств* (`flex`, ...)

* Так делают все минификаторы CSS

CSS Definition Syntax API

```
csstree.lexer
    .matchDeclaration(declarationNode)
    .isType(node, 'length')
```

Опыт CSSO: unitless zero length

- Ищем `0px` / `0%`
- Удаляем `px` / `%`
- ~~Черный список свойств (`flex`, ...)~~
- Проверяем после трансформации, что `0` это по прежнему длина

Валидация

Проверка на ошибки

```
const { error } = csstree.lexer.matchProperty(
    'background',
    'top left url(image.png) no-repeat greeen'
);
```

<'background'>

Test value:

top left url(image.png) no-repeat greeen

```
Uncomplete match
  syntax: [ <bg-layer> , ]* <final-bg-layer>
   value: top left url(image.png) no-repeat greeen
  ----------------------------------------^
```

Плагины/пакеты

- csstree-validator – npm пакет + консольная команда
- stylelint-csstree-validator – плагин для stylelint
- gulp-csstree – плагин для gulp
- SublimeLinter-contrib-csstree – плагин для Sublime Text
- vscode-csstree – плагин для VS Code
- atom-plugin – плагин для Atom

Огромное спасибо Сергею Мелюкову за его вклад и поддержку!

Поиск

Как найти цвет?

Синтаксис цвета (без L4)

<color>

Formal syntax:

```
<rgb()> | <rgba()> | <hsl()> | <hsla()> | <hex-color> | <named-color> | currentcolor | <deprecated-system-color>
```

```
<rgb()> = rgb( [ [ <percentage>{3} | <number>{3} ] [ / <alpha-value> ]? ] | [ [ <percentage>#{3} | <number>#{3} ] , <alpha-value>? ] )
<alpha-value> = <number> | <percentage>
<rgba()> = rgba( [ [ <percentage>{3} | <number>{3} ] [ / <alpha-value> ]? ] | [ [ <percentage>#{3} | <number>#{3} ] , <alpha-value>? ] )
<hsl()> = hsl( [ <hue> <percentage> <percentage> [ / <alpha-value> ]? ] | [ <hue> , <percentage> , <percentage> , <alpha-value>? ] )
<hue> = <number> | <angle>
<hsla()> = hsla( [ <hue> <percentage> <percentage> [ / <alpha-value> ]? ] | [ <hue> , <percentage> , <percentage> , <alpha-value>? ] )
<named-color> = transparent | aliceblue | antiquewhite | aqua | aquamarine | azure | beige | bisque | black | blanchedalmond | blue | blueviolet | brown | burlywood |
cadetblue | chartreuse | chocolate | coral | cornflowerblue | cornsilk | crimson | cyan | darkblue | darkcyan | darkgoldenrod | darkgray | darkgreen | darkgrey | darkkhaki |
darkmagenta | darkolivegreen | darkorange | darkorchid | darkred | darksalmon | darkseagreen | darkslateblue | darkslategray | darkslategrey | darkturquoise | darkviolet |
deeppink | deepskyblue | dimgray | dimgrey | dodgerblue | firebrick | floralwhite | forestgreen | fuchsia | gainsboro | ghostwhite | gold | goldenrod | gray | green | greenyellow |
grey | honeydew | hotpink | indianred | indigo | ivory | khaki | lavender | lavenderblush | lawngreen | lemonchiffon | lightblue | lightcoral | lightcyan | lightgoldenrodyellow |
lightgray | lightgreen | lightgrey | lightpink | lightsalmon | lightseagreen | lightskyblue | lightslategray | lightslategrey | lightsteelblue | lightyellow | lime | limegreen | linen |
magenta | maroon | mediumaquamarine | mediumblue | mediumorchid | mediumpurple | mediumseagreen | mediumslateblue | mediumspringgreen | mediumturquoise |
mediumvioletred | midnightblue | mintcream | mistyrose | moccasin | navajowhite | navy | oldlace | olive | olivedrab | orange | orangered | orchid | palegoldenrod | palegreen |
paleturquoise | palevioletred | papayawhip | peachpuff | peru | pink | plum | powderblue | purple | rebeccapurple | red | rosybrown | royalblue | saddlebrown | salmon |
sandybrown | seagreen | seashell | sienna | silver | skyblue | slateblue | slategray | slategrey | snow | springgreen | steelblue | tan | teal | thistle | tomato | turquoise | violet |
wheat | white | whitesmoke | yellow | yellowgreen | <-non-standart-color>
<-non-standart-color> = -moz-ButtonDefault | -moz-ButtonHoverFace | -moz-ButtonHoverText | -moz-CellHighlight | -moz-CellHighlightText | -moz-Combobox | -moz-
ComboboxText | -moz-Dialog | -moz-DialogText | -moz-dragtargetzone | -moz-EvenTreeRow | -moz-Field | -moz-FieldText | -moz-html-CellHighlight | -moz-html-CellHighlightText |
-moz-mac-accentdarkestshadow | -moz-mac-accentdarkshadow | -moz-mac-accentface | -moz-mac-accentlightesthighlight | -moz-mac-accentlightshadow | -moz-mac-
accentregularhighlight | -moz-mac-accentregularshadow | -moz-mac-chrome-active | -moz-mac-chrome-inactive | -moz-mac-focusring | -moz-mac-menuselect | -moz-mac-
menushadow | -moz-mac-menutextselect | -moz-MenuHover | -moz-MenuHoverText | -moz-MenuBarText | -moz-MenuBarHoverText | -moz-nativehyperlinktext | -moz-
OddTreeRow | -moz-win-communicationstext | -moz-win-mediatext | -moz-activehyperlinktext | -moz-default-background-color | -moz-default-color | -moz-hyperlinktext | -moz-
visitedhyperlinktext | -webkit-activelink | -webkit-focus-ring-color | -webkit-link | -webkit-text
<deprecated-system-color> = ActiveBorder | ActiveCaption | AppWorkspace | Background | ButtonFace | ButtonHighlight | ButtonShadow | ButtonText | CaptionText | GrayText |
Highlight | HighlightText | InactiveBorder | InactiveCaption | InactiveCaptionText | InfoBackground | InfoText | Menu | MenuText | Scrollbar | ThreeDDarkShadow | ThreeDFace |
ThreeDHighlight | ThreeDLightShadow | ThreeDShadow | Window | WindowFrame | WindowText
```

Цвет это ...

- HexColor
- rgb(), rgba(), hls(), hlsa()
- named colors
- deprecated system colors
- vendor named colors
- ...
- CSS Color Module Level 4
    - изменения в HexColor, rgb(a)/hls(a)
    - новое: hwb(), lab(), lch(), color(), device-cmyk()

Если руками,

то это очень много кода...

Поиск с CSS Definition Syntax

```
// найти все цвета
csstree.lexer.matchAllFragments(ast, 'Type', 'color')
```

Поиск с CSS Definition Syntax

```
// найти все цвета
csstree.lexer.matchAllFragments(ast, 'Type', 'color')

// ... имена анимаций
csstree.lexer.matchAllFragments(ast, 'Type', 'keyframes-name')

// ... имена шрифтов
csstree.lexer.matchAllFragments(ast, 'Type', 'family-name')
```

Расширение синтаксиса

Откуда дровишки?

- Используется словарь mdn-data
- Дополняется/патчится в CSSTree
- Словарь можно изменить и/или расширить

Откуда дровишки?

- Используется словарь mdn-data
- Дополняется/патчится в CSSTree
- Словарь **можно изменить и/или расширить**

Меняем словарь

```
const mySyntax = csstree.fork({
    properties: {
        'my-prop': '<length> <my-type>',
        ...
    },
    types: {
        'my-type': 'foo | bar',
        ...
    }
})
```

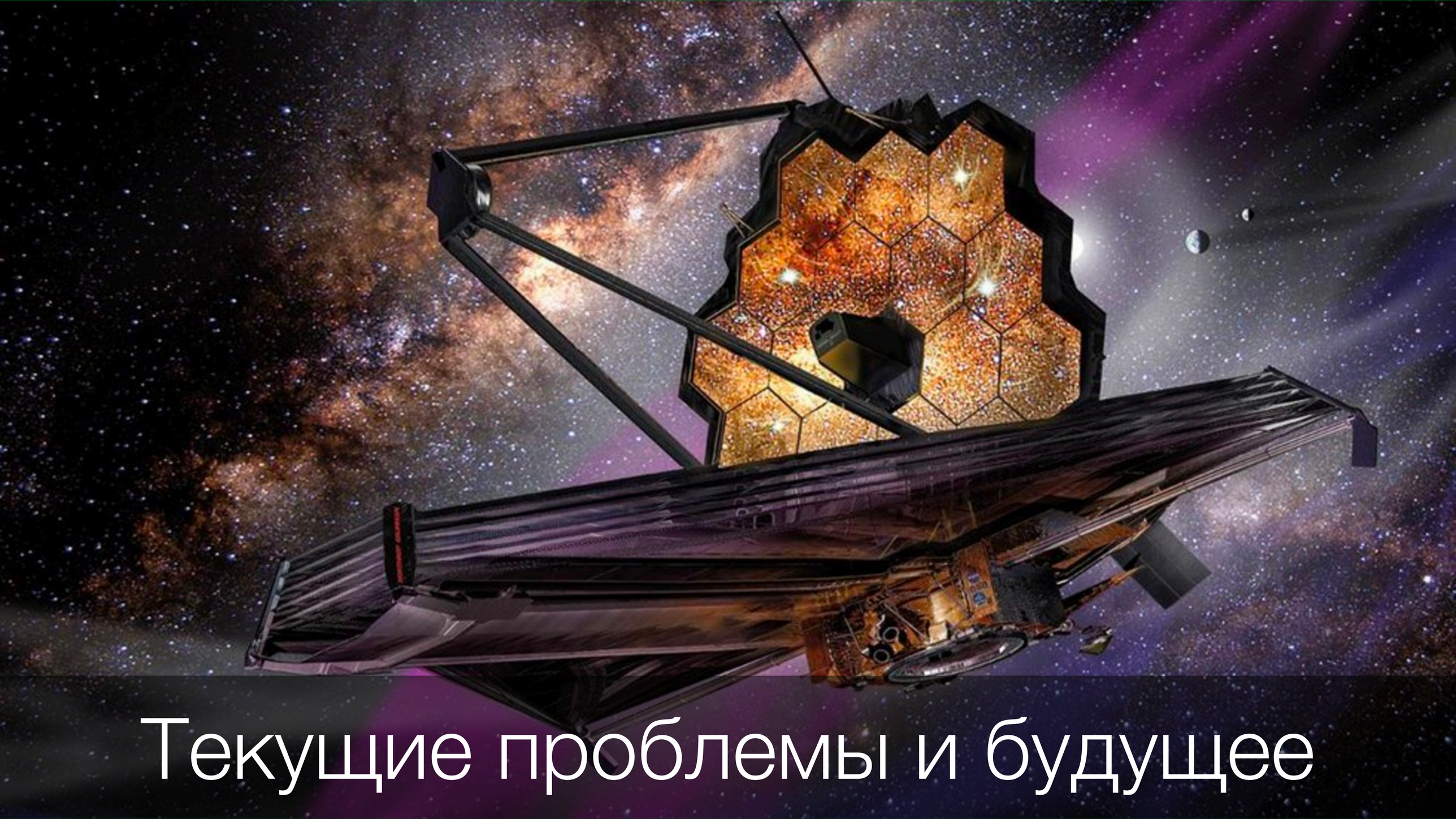
Все работает так же

```
// матчинг/валидация
mySyntax.lexer.matchDeclaration('my-prop', '...')

// поиск
csstree.lexer.matchAllFragments(ast, 'Type', 'my-type')
```

Так можно добавить поддержку нестандарного/экспериментального синтаксиса

CSS Modules, плагины к PostCSS, ...



Текущие проблемы и будущее

Проблемы

- Пробелы в спеках
- Неполные словари
- Не все выражено в определениях (часть в тестах рядом)
- ...

У CSS Definition Syntax большой потенциал

CSS вынужден расширяться, будет вынужден расширяться и Definition Syntax

Пробелы в спеках / неполные словари

PR в соответствующую репу!

CSS спеки: github.com/w3c/csswg-drafts

Словарь: github.com/mdn/data

Расширение определений

возможно!

Например, range definition notation, вроде <length[0,∞]>

Все решаемо!

Нужны ваши опыт и фидбек

Попробуйте ;)

Спасибо!

Рома Дворнов lahmatiy rdvornov