

**EMOCON
2017 S/S**

Again, CSS Parser 😞

@cwdxoh

cwdoch

SKplanet

Google Developer Expert for the Web technologies

오늘 할 얘기는...

| CSS를 어떻게 파악하는가는 절대 아닙니다.

CSS Parser?

- CSS를 입력으로 받아 의미있는 단위로 파싱하여 그 결과를 반환

Demo

The screenshot shows a browser window with the title "CSS.stringify()". The address bar indicates the page is at cwidoh.com/cssparser.js/demo/CSS_stringify.html. The page content includes a header with links to Issues, closed Issues, pull requests, closed pull requests, and contributors. Below this, a note states "'CSS.stringify()' supports to transform CSS Document to JSON style." It also mentions the project is based on `cssparser.js` (Project on [github](#), [Project site](#)). A thanks message goes to [JSONEditor](#) for using great Source/JSON Editor at this demo.

The main area displays two side-by-side code editors. The left editor shows the original CSS code:

```
/* Sample at-rule CSS #CSSTricks.com [1]
 *
 * [1] https://css-tricks.com/the-at-rules-of-css/
 */
charset 'UTF-8';

import 'global.css';

/* Namespace for XHTML */
@namespace url(http://www.w3.org/1999/xhtml);

/* Namespace for SVG embedded in XHTML */
@namespace svg url(http://www.w3.org/2000/svg);

@document
/* Rules for a specific page */
@url(http://css-tricks.com/);

/* Rules for pages with a URL that begin with... */
@url-prefix(http://css-tricks.com/snippets/);

/* Rules for any page hosted on a domain */
@domain(css-tricks.com);

/* Rules for all secure pages */
@regexp("https:");
{
    /* Start styling */
    body { font-family: Comic Sans; }

    @font-face {
        font-family: 'MyWebFont';
        src: url('myfont.woff2') format('woff2'),
             url('myfont.woff') format('woff');
    }
}
```

The right editor shows the transformed JSON object:

```
1: {
  2:   "type": "STYLESCHEET",
  3:   "value": [
  4:     {
  5:       "type": "AT_RULE",
  6:       "rule": {
  7:         "type": "ID",
  8:         "value": "charset",
  9:         "prefix": "#"
  10:      },
  11:      "value": {
  12:        "type": "STRING",
  13:        "value": "\\"UTF-8\\"
  14:      }
  15:    },
  16:    {
  17:      "type": "AT_RULE",
  18:      "rule": {
  19:        "type": "ID",
  20:        "value": "Import",
  21:        "prefix": "#"
  22:      },
  23:      "value": {
  24:        "type": "STRING",
  25:        "value": "'global.css'"
  26:      }
  27:    },
  28:    {
  29:      "type": "AT_RULE",
  30:      "rule": {
  31:        "type": "ID",
  32:        "value": "namespaces",
  33:        "prefix": "#"
  34:      },
  35:      "value": {
```

At the bottom, there are three buttons: "Transform (Atomic)", "Transform (Deep)" (which is selected), and "Transform (Simple)". A green box at the bottom indicates "CSS is valid!"



Chang W. Doh

February 19 · ▾

오늘 스터디 간 김에 목표했던 수준까지 [파서](#) 완료. 이번 주에 집에서 할 숙제는 슬슬 코드 정리해서 push 하기. #다트의힘

CSS.stringify()

CWDOH.COM



Kim Jong Man, Jahyoung Koo and 8 others

2 Comments



Like



Comment



Share



Hika Maeng 어디에 쓰면 좋을지 용도도 가르쳐주세요 ^^

[Like](#) · [Reply](#) · February 19 at 11:54pm



Chang W. Doh ...그게 문제입니다. optimizer라도 만들어야 할까봐요. ㅜ

[Like](#) · [Reply](#) · February 19 at 11:55pm



Write a reply...



GIF



왜 만들었나

2013년 어느날

jQuery Mobile 디자인 도구 개발

JS 기반의 에디팅 프레임워크 + 네이티브 기반의 에디터

마크업은 프레임워크에서 처리

- 작업 문서를 네이티브 에디터에서 로딩
- 주입된 프레임워크가 로딩된 문서를 탐색
- 웹뷰 내의 콘텐츠에 대한 UI 인터랙션을 프레임워크가 처리
- 네이티브 에디터에 편집 방법을 전달
- 에디터에서 편집 결과를 프레임워크에 전달/갱신

CSS는 네이티브 에디터에서 처리

| 뭔가 조금 따로 노는 듯한 느낌적인 느낌...



"CSS도 자바스크립트에서 파싱해서 프레임워크 내로 녹이면 어떨까?"

| ...는 10%였고, 사실 개발이 하고 싶었음

재밌겠다!! 파서를 만들자!!

첫번째 실수

하나 하나 코드를 작성하기는 어려우니 파서 생성기를 찾아보자.

Tiison

| 걸출한 레퍼런스였던...



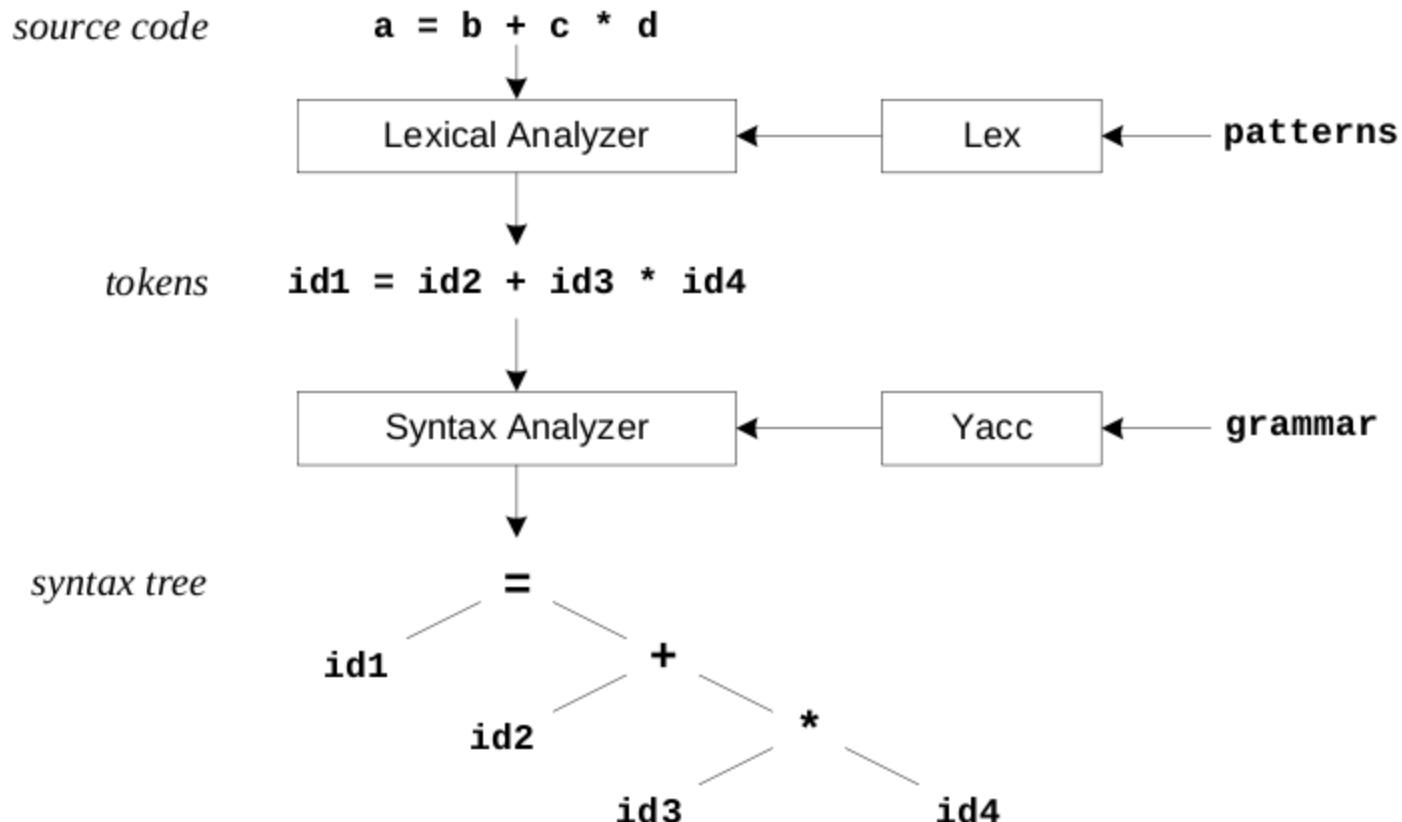
Parsing?

- 문자열을 문법적으로 의미있는 단위로 분해/분석하는 행위

Parser: How it works

소스의 입력 스트림으로부터 토큰 추출

토큰을 처리하는 규칙(문법)을 통해 계층적인 형태(AST)로 출력



기본 구조

```
%lex
```

```
%%
```

```
// 정규식 기반으로 토큰 정의
```

```
/lex
```

```
// 연산자의 우선순위 및 관계 정의
```

```
%start 시작심볼
```

```
%%
```

```
// 토큰에 대한 유도식(문법) 정의
```

계산기 예제 (1/2)

```
%lex
%%
\s+          /* skip whitespace */
[0-9]+(\. [0-9]+)?\b    return 'NUMBER'
"\*"        return '*'
"/"          return '/'
"\_"        return '_'
"\+"        return '+'
"\^"        return '^'
"\("         return '('
"\)"         return ')'
"PI"         return 'PI'
"E"          return 'E'
<<EOF>>
.

/lex
```

계산기 예제 (2/2)

```
%left '+' '-'
%left '*' '/'
%left '^'
%left UMINUS

%start expressions

%%
expressions
: e EOF      { return $1; }
;
e
: e '+' e -> { op: '+', lhs: $1, rhs: $3 }
| e '-' e -> { op: '-', lhs: $1, rhs: $3 }
| e '*' e -> { op: '*', lhs: $1, rhs: $3 }
| e '/' e -> { op: '/', lhs: $1, rhs: $3 }
| e '^' e -> { op: '^', lhs: $1, rhs: $3 }
| NUMBER -> Number($1)
| E       -> Math.E
| PI     -> Math.PI
;
```

5 * PI ^ 2 + 32 - 11.3 의 결과

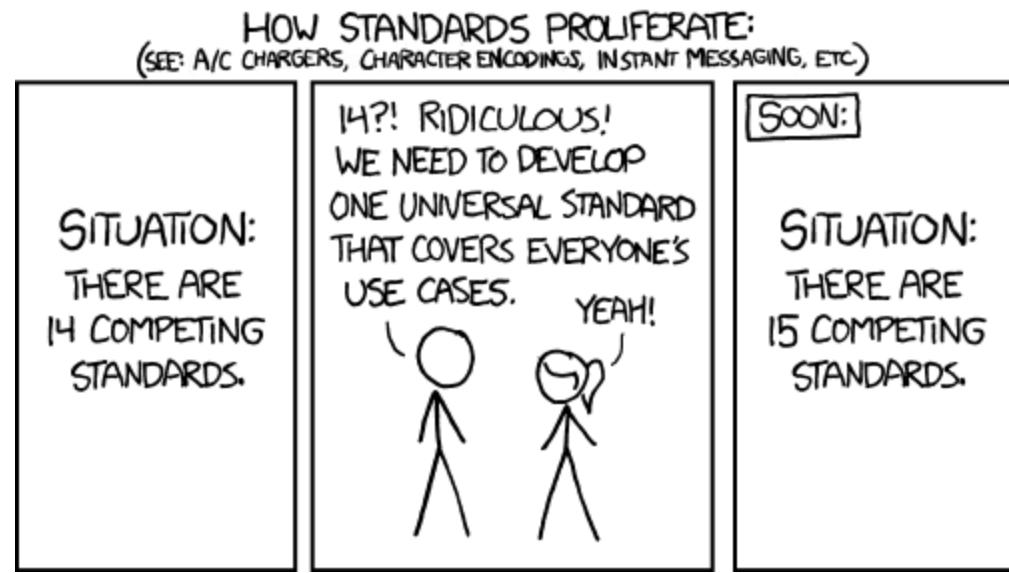
```
{  
  "operator": "-",
  "lhs": {
    "operator": "+",
    "lhs": {
      "operator": "*",
      "lhs": 5,
      "rhs": {
        "operator": "^",
        "lhs": 3.141592653589793,
        "rhs": 2
      }
    },
    "rhs": 32
  },
  "rhs": 11.3
}
```

CSS 룰을 작성하자.

| ~~여기에서 멈췄어야 했다.~~

"W3C CSS 표준안에 따라서 문법을 정의하자."

~~이런 경우는 아니지만...~~



예를 들자면...

- Cascading Style Sheets, level 1 - W3C
- Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) - W3C
- CSS Syntax Module Level 3 - W3C
- Media Queries - W3C
- CSS wiki - W3C
- CSS keyword index

실 사례들도 참고해야 하니까...



| 그 외에도 예제들을 이곳 저곳에서...

생각했던 결과



만들어진 결과



일단은 GitHub에 올려두자.



Issue, 예를 들면...

공백을 토큰으로 받아 처리하면 너무 귀찮으니까,
일단 공백은 무시하고 프로토타입을 만들어야지

```
%lex
{whitespace}+          {}
{vendor_prefix}{ident}  {return 'VENDOR_PREFIX_IDENT';}

%%
```

2014년 어느 날...

~~너는 테스트를 소중히 여기지 않았지~~



tomgenoni commented on 22 Dec 2014



If I parse the string:

```
.foo .bar { bar:foo; }
```

The data object contains:

```
selector: ".foo.bar"
```

when it should be:

```
selector: ".foo .bar"
```

지키지도 못할 약속



cwdoh commented on 3 Jun 2014

Owner



I reviewed css grammar rules in this project and decided to rewrite them all.
Currently it can't avoid conflicts with simple modifications. :(

p.s. I know this is very late reply to your request. Sorry for that. :)

~~다시 만든다고는 했지만 언제라고는 얘기안했다.~~

2015년, 어라??

깜빡하고 그대로 쭈욱~~

<input type="checkbox"/>  Error on comma after pseudo element	 2
#11 by pe6o was closed on 24 Feb	
<input type="checkbox"/>  Browser hangs when parsing imports 	
#9 by ben8p was closed on 6 Aug 2015	
<input type="checkbox"/>  Error with selector that contains `()`  	  2
#8 by mahdaen was closed on 24 Feb 	Rewrite parser
<input type="checkbox"/>  Parsing type option is ignored  	  2
#7 by maxnordlund was closed 27 days ago	
<input type="checkbox"/>  whitespace in selectors is removed 	 4
#6 by tomgenoni was closed on 25 Apr 2015	
<input type="checkbox"/>  browser dead when parse comment string 	  3
#5 by ben7th was closed on 6 Aug 2015	
<input type="checkbox"/>  Support for sibling selector ~	 1
#4 by frequent was closed on 4 Aug 2014	
<input type="checkbox"/>  grammar conflicts 	  3
#2 by mainiak was closed on 24 Feb 	Rewrite parser
<input type="checkbox"/>  convert back to css 	  6
#1 by jdolitsky was closed on 17 Feb 	Rewrite parser

2017년 어느날

아, 맞다!



인간의 욕심은 끝이 없고, 같은 실수를 반복한다.

- 파서 생성기를 쓰자.
- CSS 룰을 새로 작성하자.
- "표준은 W3C에 있는 것을 따르면 되겠다."
- 실 사례들도 참고해야 하니까...

아차!!!

공백을 토큰으로 받아 처리하면 너무 귀찮으니까,
일단 공백은 무시하고 프로토타입을 만들어야지
응??

규칙을 잘 정의해서 회피하자.

기존 토큰

```
[#]{ident}    {return 'HASH_STRING';}  
[*]          {return 'ASTERISK';}  
{ident}       {return 'GENERAL_IDENT';}
```

그냥 공백이 포함된 토큰을 추가 정의하자

```
[.]{ident}{WS}+  {return 'SELECTOR_CLASS_WITH_WS')}  
[#]{ident}{WS}+  {return 'SELECTOR_ID_WITH_WS')}  
[*]{WS}+         {return 'ASTERISK_WITH_WS')}  
{ident}{WS}+     {return 'SELECTOR_TYPE_WITH_WS')}
```

값(Value) 규칙에 추가 토큰 처리를 기술

```
HashVal
: HASH_STRING
| HEXA_NUMBER
| SELECTOR_ID_WITH_WHITESPACE    -> $1.trimRight()
;

IDENT
: GENERAL_IDENT
| VENDOR_PREFIX_IDENT
| SELECTOR_TYPE_WITH_WHITESPACE  -> $1.trimRight()
;
```

공백이 뒤따르는 셀렉터(DescendantSelector)에 대한 규칙 정의

DescendantSelector

```
: ASTERISK_WITH_WHITESPACE      -> ...
| SELECTOR_TYPE_WITH_WHITESPACE -> ...
| SELECTOR_CLASS_WITH_WHITESPACE -> ...
| SELECTOR_ID_WITH_WHITESPACE   -> ...
;
```

SelectorGroup

```
: Selector
| Selector SelectorGroup
| Selector SelectorCombinator SelectorGroup
| DescendantSelector
| DescendantSelector SelectorGroup
| DescendantSelector SelectorCombinator SelectorGroup
;
```

CSS

```
.meetat-card-container .mdl-card {  
    min-height: 336px;  
}
```

Result (type: Simple)

```
{  
  "type": "rule",  
  "selectors": [  
    ".meetat-card-container .mdl-card"  
  ],  
  "declarations": {  
    "min-height": "336px"  
  }  
}
```

| 이런 작업들을 찾아서 계속한 결과...



두 마리가 되었...

결론은...

누가 어디에 쓰면 좋을지 아이디어 좀 주세요.

버그 리포트도 좋고요. ~~PR 주시면 더 고맙...~~

Thanks

cwdoh

changwook.doh@gmail.com