스타일 객체 활용

Feb 26, 2015 - 이준호

스타일 객체???

http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113

Document Object Model (DOM) Level 2 Style Specification

Version 1.0

W3C Recommendation 13 November, 2000

This version:
http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113
(PostScript file , PDF file , plain text , ZIP file)
Latest version:
http://www.w3.org/TR/DOM-Level-2-Style
Previous version:
http://www.w3.org/TR/2000/PR-DOM-Level-2-Style-20000927

스타일 객체??

CSSStyleDeclaration

Overview **Package** Class **Tree** **Index** **Help**
PREV CLASS **NEXT CLASS**
SUMMARY: INNER | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

org.w3c.dom.css
Interface CSSStyleDeclaration

public interface **CSSStyleDeclaration**

The `CSSStyleDeclaration` interface represents a single CSS declaration block. This interface may be used to determine the style properties currently set in a block or to set style properties explicitly within the block.

While an implementation may not recognize all CSS properties within a CSS declaration block, it is expected to provide access to all specified properties in the style sheet through the `CSSStyleDeclaration` interface. Furthermore, implementations that support a specific level of CSS should correctly handle CSS shorthand properties for that level. For a further discussion of shorthand properties, see the `CSS2Properties` interface.

This interface is also used to provide a **read-only** access to the computed values of an element. See also the `ViewCSS` interface.

Note: The CSS Object Model doesn't provide an access to the specified or actual values of the CSS cascade

DOM

DOCUMENT
OBJECT
MODEL

스타일 객체?

- CSSStyleDeclaration
- document.createElement('div').style
- 최종적으로 모든 스타일의 정보가 저장되는 객체
- MSStyleCSSProperties, CSS2Properties

CSSStyleDeclaration

```
document.getElementById('console').innerHTML = '' +
document.getElementsByTagName('head')[0].style + '<br>' +
document.getElementsByTagName('html')[0].style + '<br>' +
document.createElement('script').style + '<br>' +
document.createElement('link').style + '<br>';
```

CSSStyleDeclaration 활용

```
var div_test = document.getElementById('div_test');
div_test.style.left = '100px';
...
```

<style></style>, <link></link>

```
var canvas = document.getElementById('canvas'),
context = canvas.getContext('2d');

context.fillStyle = 'red';
context.fillRect( 0, 0, 100, 100 );

// document.createElement('style').style

// canvas의 context를 얻어와 제어하는 것처럼..
// canvas라는 태그는 실제 동작하는 context를 HTML 문서에 부착하기 위한 공통 인터페이스로 이해할 수 있음.
// style 엘리먼트의 style 객체를 제어하는 것이 아닌.
// style 엘리먼트에 있는 실제 스타일 시트 객체를 얻어와야 한다.
```

스타일 객체(CSSStyleDeclaration)를 찾아서.

- console.dir(document.getElementById('style_test'))
- console.dir(document.getElementById('style_test').sheet)
- console.dir(document.getElementById('style_test').sheet.cssRules)
- console.dir(document.getElementById('style_test').sheet.cssRules[0])
- console.dir(document.getElementById('style_test').sheet.cssRules[0].style)

정리하면

Browser	CSSStyleDeclaration	CSSStyleSheet	CSSRuleList
IE	MSStyleCSSProperties	sheet, styleSheet	cssRules, rules
Chrome	CSSStyleDeclaration	sheet	cssRules, rules
FireFox	CSS2Properties	sheet	cssRules
Safari	CSSStyleDeclaration	sheet	cssRules, rules

selectorText, style

추가, 제거

```
var styleEl = document.createElement('style');
document.getElementsByTagName('head')[0].appendChild(styleEl);

var sheet = styleEl.sheet || styleEl.styleSheet;
var rules = sheet.cssRules || sheet.rules;

sheet.insertRule( 'p{}', 0 ); // sheet.addRule( 'p', ' ' );

var rule = rules[0];
rule.style['color'] = 'red';

sheet.deleteRule(0); // sheet.removeRule(0);
```

This is Test

결론

- 최종적으로 스타일 속성을 저장하고 제어의 대상이 되는 것은 CSSStyleDelaration 객체
- 모든 DOM Element에는 style 속성이 있음. (CSSStyleDelaration)
- 스타일 시트 또한 CSSRule 객체에 style 속성이 있음.
- 따라서 스타일시트 또한 얼마든지 제어가 가능.
- 스타일시트의 CSSRuleList를 제어하게 되면..
  - 런타임에 동적으로 스타일 시트를 변경할 수 있게 되고.
  - 풍부한 프로그래밍 언어로서의 자바스크립트로 스타일 시트를 통제할 수 있음.
  - 인라인이 아니라 스타일 시트에 지정한 스타일이 되므로
  - 스타일시트의 성능과 하드웨어의 도움 등을 그대로 활용 가능.

THANK YOU

http://ligo.kr/akj