

GABRIELE ROMANATO

JavaScript: leggere gli stili CSS degli elementi

Molti sviluppatori che iniziano ad affrontare l'interazione tra JavaScript e CSS di solito pensano che l'oggetto `style` associato a ciascun elemento consenta loro di conoscere gli stili applicati all'elemento. In realtà questo oggetto corrisponde all'omonimo attributo usato sugli elementi. In altre parole non consente di accedere agli stili applicati tramite una regola CSS esterna. Fortunatamente esiste un modo per accedere a questi valori. Vediamo come.

Creiamo un oggetto fittizio per questo scopo:

```
var $ = function(){};
```

Quindi definiamo dei metodi helper per gestire le proprietà CSS in notazione camel-case o separate da trattini:

```
$.prototype.Utils = {  
  
    toCamelCase: function(hyphenatedValue) {  
        var result = hyphenatedValue.replace(/-\w/g,  
function(character) {  
            return character.charAt(1).toUpperCase();  
        });  
        return result;  
    },  
  
    toHyphens: function(camelCaseValue) {  
        var result = camelCaseValue.replace(/[A-Z]/g,  
function(character) {  
            return ('-' + character.charAt(0).toLowerCase());  
        });  
        return result;  
    }  
};
```

```
    }  
  
};
```

Il primo metodo trasforma una proprietà come `background-position` in `backgroundPosition`, mentre il secondo fa l'inverso.

Ora dobbiamo percorrere due strade: una standard per i browser che supportano il metodo del DOM W3C `getComputedStyle()` e una per Internet Explorer, che usa invece la proprietà `currentStyle`:

```
$.prototype.CSS = {  
  
    getAppliedStyle: function(element, styleName) {  
        var style = '';  
        if (window.getComputedStyle) {  
            // W3C: notazione con trattini  
            style =  
element.ownerDocument.defaultView.getComputedStyle(element,  
null).getPropertyValue($.Utils.toHyphens(styleName));  
        } else if (element.currentStyle) {  
            // Internet Explorer: notazione in camel-case  
            style =  
element.currentStyle[$.Utils.toCamelCase(styleName)];  
        }  
  
        return style;  
    }  
  
};
```

Quindi se un elemento ha questa regola:

```
#test {
```

```
width: 200px;  
}
```

avremo:

```
console.log($.CSS.getAppliedStyle(document.getElementById('test'), 'width')); // '200px'
```