Stwórzmy sobie obiekt [String]:

{codecitation}var txt = new String('To jest mój napis');{/codecitation}

Obiekt ten posiada kilka metod (funkcji), choćby:

{codecitation}alert(txt.toLowerCase()); // małymi literami{/codecitation}

Dodajmy do niego dodatkową metodę:

{codecitation}txt.letterSpacing = function() {
    var n = this.length;
    var str_result = '';
    for(var i=0; i<n; i++) {
        str_result += this.charAt(i) + ' ';
    }
 return str_result;

}{/codecitation}

Oraz jej wywolanie:

{codecitation} alert(txt.letterSpacing()); {/codecitation}

Zauważ, że jeśli zaraz za tym wyświetlisz ponownie

{codecitation} alert(txt); {/codecitation}

Nadal litery są razem. Dzieje się tak dlatego, że nie zmienialiśmy this, a jedynie przysywaliśmy wszystko do str_result a potem to zwróciliśmy.

Wartym zauważenia jest też to, że jeśli dodasz teraz nowy obiekt String, to nie posiada on tej metody:

{codecitation}var nowy_txt = new String('Poszła baba na targ');

alert(nowy_txt.letterSpacing());{/codecitation}

Taki kod wywoła błąd:

{codecitation}Błąd: nowy_txt.letterSpacing is not a function{/codecitation}

Dlaczego? Dodaliśmy tę metodę dokładnie do tego jednego obiektu, nie do wszystkich obiektów String. To tak jakbyśmy nauczyli czegoś jednego człowieka. Nie możemy później wymagać, aby każdy to potrafił.

Jest jednak i na to sposób, choć o tym w innej poradzie.