

## Dopisywanie własnych metod do obiektów

Wpisany przez Patryk yarpo Jar  
wtorek, 22 września 2009 18:04

---

W poradzie dotyczącej [dodawania nowych metod](#) do istniejących obiektów wspomniałem, że da się dodawać metody do wszystkich obiektów - nie tylko do tego jednego wybranego.

Przykładowy kod:

```
{codecitation} String.prototype.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}
```

W powyższym kodzie dodaliśmy nową metodą do wszystkich obiektów tworzonych według prototypu obiektu String. Dzięki temu, każdy obiekt String będzie posiadał tę metodę:

```
{codecitation}var oNapis1 = new String(&quot;Mój pierwszy napis&quot;);  
var oNapis2 = new String(&quot;Kolejny tekst&quot;);  
  
alert(oNapis1.letterSpacing());  
alert(oNapis2.letterSpacing());{/codecitation}
```

Gdyby teraz do jednemu z tych obiektów nadpisać metodę letterSpacing:

```
{codecitation}oNapis1.letterSpacing = function() {  
    return this.toLowerCase(); // zwraca wszystko małymi literami
```

## Dopisywanie własnych metod do obiektów

Wpisany przez Patryk yarpo Jar  
wtorek, 22 września 2009 18:04

---

```
}
```

```
alert(oNapis1.letterSpacing());  
alert(oNapis2.letterSpacing());{/codecitation}
```

Pierwszy alert da w wyniku to, co zakodowaliśmy w definicji `letterSpacing` przypisanej do prototypu obiektu `String`. Drugi alert wyświetli wynik działania metody nadpisanej (czyli `toLowerCase()`).

Zrozumienie tego początkowo nie jest łatwe, ale można się w końcu przyzwyczaić.