

Jest kilka sposobów na łączenie ciągów znaków. W tym artykule opiszę 3.

Sposób I - "optymistyczny";

Optymistyczny dlatego, że używa się samych plusów.

```
{codecitation class='brush: js'}var string_1 = 'Ala ma ';
```

```
var string_2 = 'kota';
```

```
alert(string_1 + string_2);{/codecitation}
```

Korzystając z tej metody pamiętaj o [różnych niespodziankach](#) związanych z wykorzystywaniem operatora '+', który jest też wykorzystywany do dodawania liczb.

Sposób II - obiektowy

Wykorzystamy tu wbudowaną metodę obiektu String.

```
{codecitation class='brush: js'}var string_1 = 'Ala ma ';
```

Różne sposoby konkatenacji ciągów znaków

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 23:16

```
string_1.concat('kota');
```

```
alert(string_1);{/codecitation}
```

Metoda ta, na pierwszy rzut oka wydaje się mniej intuicyjna. Z [moich testów wynika](#), że także nie musi (a w przypadku IE zdecydowanie nie jest) być najbardziej wydajna. Choć to dziwne.

Sposób III - własny

Stwórzmy szybko prosty obiekt YString (wielka litera z przodu nie przez przypadek - konwencja. Obiekty wymagające operatora new staraj się nazywać wielką literą. Obiekty, które nie wymagają new, małą literą):

```
{/codecitation class="brush: js"}var YString = function (sep) {
```

```
    this.__strings__ = [];
```

```
    this.separator = sep || '"';
```

```
}
```

```
YString.prototype.append = function(str) {
```

Różne sposoby konkatenacji ciągów znaków

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 23:16

```
this.__strings__.push(str);
```

```
return this;
```

```
};
```

```
YString.prototype.toString = function() {
```

```
    return this.__strings__.join(this.separator);
```

```
};{/codecitation}
```

I przykład zastosowania takiego kodu:

```
{codecitation class='brush: js'}var oString = new YString('&quot; &quot;);
```

Różne sposoby konkatencji ciągów znaków

Wpisany przez Patryk yarpo Jar
piątek, 25 września 2009 23:16

```
oString.append("Ala ma").append("kota");
```

```
alert(oString.toString());{/codecitation}
```

Dodatkową ciekawostką jest zastosowanie tu wzorca łańcuchowego (jak np. w jQuery). Po wywołaniu metody `append()`, możemy po kropce wywołać kolejną metodę tego obiektu, będzie ona działać a tej samej instancji obiektu, ponieważ w metodzie `append()` zawsze zwracamy referencję obiektu, na którym pracujemy.