

## Równe identycznemu nie równe

Wpisany przez Patryk yarpo Jar  
niedziela, 22 listopada 2009 13:07

---

W JS typ danych jest bardziej podpowiedzią dla programisty niż jakimkolwiek limitem. Na przykład taki kod:

```
{codecitation}var a = 12;  
a = &quot;To jest string&quot;;{/codecitation}
```

Nie wywoła błędu. Zmienna to po prostu pudełko, co w nią włożymy to tam będzie.

Co jeśli będziemy chcieli porównać dwie wartości?

```
{codecitation}var a = 10,  
    b = '10';  
if (a == b) { alert('są równe'); }{/codecitation}
```

Czy oby jednak naprawdę są równe... Jedna zmienna jest liczbą, a druga ciągiem znaków.

I tu właśnie wchodzi operator `identyczności` ===.

```
{codecitation}var a = 10,  
    b = '10';  
if (a == b) { alert('są równe'); }  
if (a === b) { alert('są identyczne'); }{/codecitation}
```

W JS lepiej jest stosować ===, gdyż wtedy dokładnie wiemy co chcemy uzyskać.

### Falsy values

Czyli w skrócie te wartości, które są uważane za nieprawdę

- undefined – niezdefiniowana zmienna
- null
- &quot;&quot; – pusty string
- 0 – wartość liczbowa
- NaN – not a number
- false – typ logiczny

Dla takiego kodu Firefox zwraca wyniki podane jako komentarz:

```
{codecitation}var a = null;  
if (false == a) { alert(a + ' to falsy value 1'); } // &quot;&quot;, 0, false  
if (!a) { alert(a + ' to falsy value 2'); } // wszystkie  
if (false === a) { alert(a + ' to falsy value 3'); } // tylko `false`{/codecitation}
```

Jak widać drugi sposób mówi nam tak naprawdę najmniej. Wiadomo, że coś nie jest prawdą [ale czym tak naprawdę jest to już nikt nie wie].

Jeśli zainteresowało cię porównywanie wartości w JS i chciałbyś się o tym dowiedzieć więcej proponuję strony:

[ferrante.pl/2009/09/05/falsy-values-i-operatory-porownania](http://ferrante.pl/2009/09/05/falsy-values-i-operatory-porownania) - po polsku

[11heavens.com/falsy-and-truthy-in-javascript](http://11heavens.com/falsy-and-truthy-in-javascript) - po angielsku