

{TITLE}

# Spis treści

- Po staremu
- `Intl` na ratunek!
- Formatujemy datę

[h1=start]Data po polsku[/h1]

Dzisiaj będzie krótko, rzeczowo i na temat (co dość rzadko mi się zdarza). Otóż: jak **dobrze** zrobić datę po polsku w JS i się przy tym nie narobić.

## Po staremu

Dotąd jedynym właściwym sposobem były różne dziwne kombinacje typu "stwórzmy sobie tablicę z polskimi nazwami miesięcy i dni tygodnia, a następnie podstawiamy pod dane, zwracane przez `new Date`". Mogą one wyglądać rozwlekle i odpychająco (<http://webmade.org/porady/data-po-polsku-js.php>) (a ich autorem nie jest Polak). Mogą być też wzorowane na innych językach (<http://phpjs.org/functions/strftime/>), stając się jeszcze bardziej rozwlekle i... nie JS-owe. Mogą w końcu zmienić się po prostu w wywołanie odpowiedniej, krótkiej funkcjki w PHP, po stwierdzeniu, że JS się do tego nie nadaje.

## `Intl` na ratunek!

Na szczęście się to zmieniło. I to nie znowu tak niedawno (bo w 2012). Wtedy powstał standard ECMA-402 (ECMAScript Internationalization API) (<http://www.ecma-international.org/ecma-402/1.0/>). Oczywiście na odzew ze strony producentów przeglądarek trzeba było ciut poczekać, niemniej – jeśli wierzyć MDN ([https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Intl#Browser\\_Compatibility](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl#Browser_Compatibility)) – działa dziś wszędzie, oprócz Safari i – co prawdopodobnie może być o wiele ważniejsze – node.js (jednak od czego jest niezawodny GitHub i polyfill (<https://github.com/andyearnshaw/Intl.js>); oczywiście w node.js/io.js obsługę można sobie wkompiłować (<https://github.com/nodejs/io.js#intl-ecma-402-support>)).

Cały standard składa się z 3 głównych "klas", zamkniętych w krótkiej i schludnej przestrzeni nazw – `Intl`. Są to:

- `Intl.Collator` – służący do porównywania tekstu
- `Intl.NumberFormat` – służący do formatowania wszelkiej maści liczb (między innymi walut czy liczebników porządkowych)
- `Intl.DateTimeFormat` – tak, tego właśnie użyjemy

# Formatujemy datę

Żeby wyświetlić ładną datę po polsku, musimy stworzyć nowy obiekt "klasy"

`Intl.DateTimeFormat`. Jako pierwszy parametr konstruktor bierze nazwę języka (oczywiście zgodną ze standardem ISO, zatem dwuliterowy skrót):

```
var formatter = new Intl.DateTimeFormat( 'pl' );
```

Już! Pierwszy krok za nami. Mamy obiekt wyświetlający datę po polsku. Udostępnia on aż jedną metodę – `format` – która formatuje przekazaną jej datę (jeśli jej nie podamy, zastosuje aktualną – przynajmniej w Chrome; polecam jednak ją podawać dla zwiększenia czytelności kodu).

```
formatter.format( new Date() ); // 23.7.2016
```

Jak widać, mało imponujące. Równie dobrze można to uznać za datę po angielsku. Na pomoc przychodzi nam drugi parametr konstruktora `Intl.DateTimeFormat` – jest to obiekt opcji. Przyjrzyjmy się takiemu przykładowi:

```
var formatter = new Intl.DateTimeFormat( 'pl', {  
  day: 'numeric',  
  month: 'long',  
  year: 'numeric'  
} );  
formatter.format( new Date() ); // 23 lipca 2016
```

*Voiła!* To jest dokładnie to, o co nam chodziło. Oczywiście opcji jest więcej, po dokładny ich spis zapraszam na MDN ([https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/DateTimeFormat#Parameters](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/DateTimeFormat#Parameters)).

Copyright © by Comandeer (<https://www.comandeer.pl>).