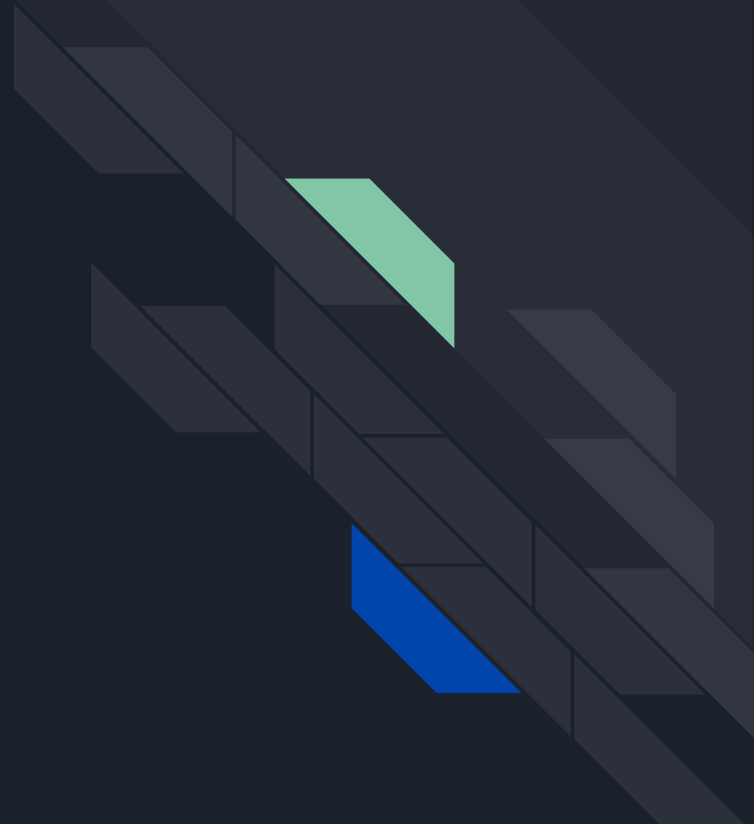




Podstawy internacjonalizacji wtyczek i motywów

Krzysztof Grabania
WordUp Trójmiasto #10, 02.12.2017r.

Internacjonalizacja





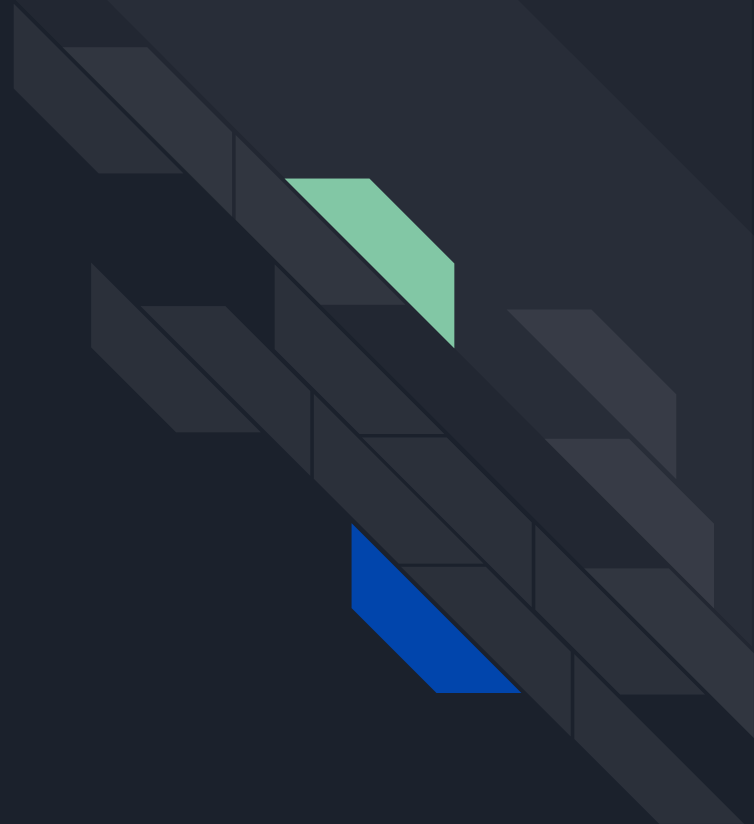
Internacjonalizacja

(i18n, 18 oznacza ilość liter pomiędzy “i” a “n” w angielskim internationalization)

Projektowanie i rozwijanie produktu, aplikacji lub zawartości dokumentu które umożliwiają łatwą lokalizację dla odbiorców docelowych, którzy różnią się co do kultury, regionu, lub języka.

Źródło: <https://www.w3.org/International/questions/qa-i18n>

Lokalizacja





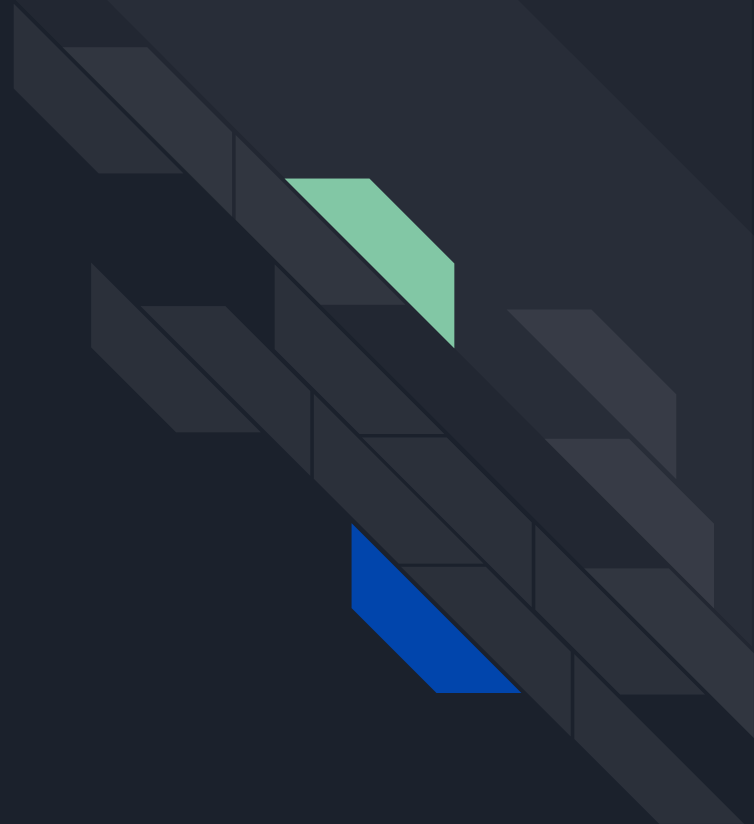
Lokalizacja

(l10n, l0 oznacza ilość liter pomiędzy “l” a “n” w angielskim localization)

Adaptowanie produktu, aplikacji lub zawartości dokumentu do specyficznego rynku pod względem językowym, kulturowym i innymi wymaganiami.

Źródło: <https://www.w3.org/International/questions/qa-i18n>

Pliki tłumaczeń





Plik .pot

(Portable Object Template)

Szablon tłumaczeń zawierający wszystkie ciągi znaków w formie **nieprzetłumaczonej**, zrozumiałej dla ludzi (najczęściej angielskiej).

```
#: includes/wc-attribute-functions.php:196  
msgid "Select"  
msgstr ""
```

Fragment pliku woocommerce.pot



Plik .po

(Portable Object)

Plik tłumaczeń zawierający wszystkie ciągi znaków w formie **przetłumaczonej**, zrozumiałej dla ludzi. Dla każdego języka tworzy się osobny plik .po.

```
#: includes/wc-attribute-functions.php:196  
msgid "Select"  
msgstr "Wybierz"
```

Fragment pliku woocommerce-pl_PL.po



Plik .mo

(Machine Object)

Plik tłumaczeń zawierający “skompilowaną” wersję tłumaczeń, zrozumiałą dla WordPressa (funkcje gettext).

```
de12·0495·0000·0000·8d13·0000·1c00·0000  
849c·0000·111a·0000·ec38·0100·0000·0000  
30a1·0100·1300·0000·31a1·0100·1800·0000  
45a1·0100·0e00·0000·5ea1·0100·0e00·0000  
6da1·0100·0b00·0000·7ca1·0100·1b00·0000  
88a1·0100·2f00·0000·a4a1·0100·1200·0000  
d4a1·0100·1e00·0000·e7a1·0100·1900·0000  
06a2·0100·0b00·0000·20a2·0100·2a00·0000
```

Fragment pliku woocommerce-pl_PL.mo w widoku tekstowym



Nazewnictwo plików

	Wzór	Przykład
Plik .pot	[slug].pot	woocommerce.pot
Plik .po	[slug]-[lang]_[country].po	woocommerce-pl_PL.po
Plik .mo	[slug]-[lang]_[country].mo	woocommerce-pl_PL.mo

Kod kraju nie zawsze jest wymagany.

Lista dostępnych kodów języków: <https://make.wordpress.org/polyglots/teams/>



Foldery tłumaczeń

Wtyczki	Motywy
<code>wp-content/languages/plugins</code>	<code>wp-content/languages/themes</code>
<code>wp-content/plugins/[path]</code> (np. <code>my-plugin/languages</code>)	<code>wp-content/themes/[path]</code> (np. <code>my-theme/languages</code>)

Domyślnie, od wersji 4.6, WordPress w pierwszej kolejności ładuje pliki z folderu `wp-content/languages`



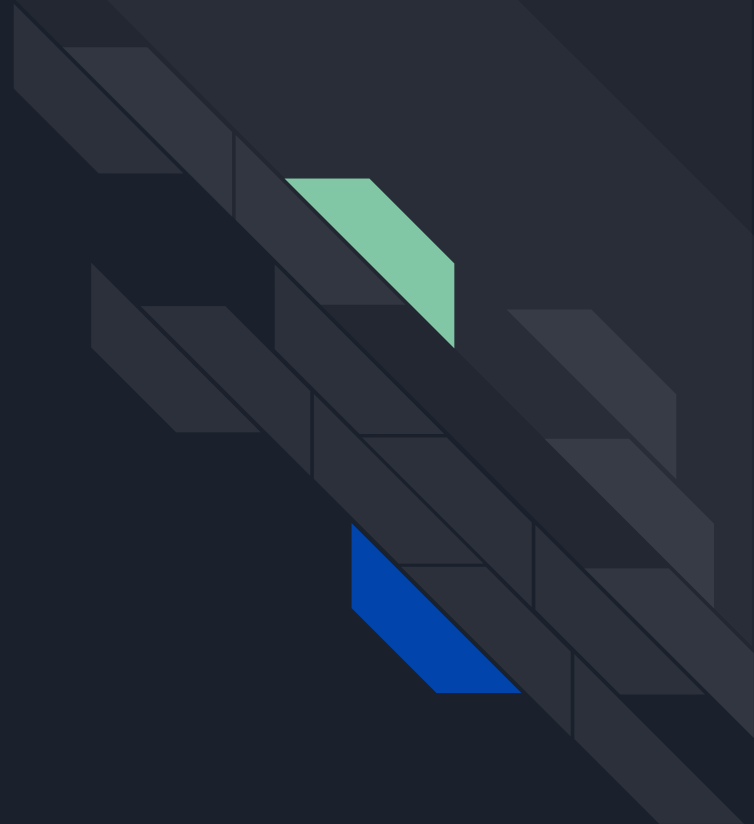
Ładowanie tłumaczenia

```
$path = basename(dirname(__FILE__)) . '/languages/';  
load_plugin_textdomain('my-plugin', FALSE, $path);
```

```
$path = get_template_directory() . '/languages';  
load_theme_textdomain('my-theme', $path);
```

Ładowanie tłumaczeń należy “podpiąć” do akcji “plugins_loaded” w przypadku wtyczki lub do “after_setup_theme” w przypadku motywu.

Funkcje





Podstawowe funkcje

Definicje	<pre>__(string, textDomain); _e(string, textDomain);</pre>
Przykłady	<pre>__(`My first plugin`, `plugin-name`); _e(`My first plugin`, `plugin-name`);</pre>
Plik .po	<pre>#: path/to/file:1 msgid "My first plugin" msgstr "Moja pierwsza wtyczka"</pre>

Funkcja `_e()` jest równoznaczna z zapisem `echo __()`



Liczby mnogie

Definicja	<code>_n(singular, plural, count, textDomain);</code>
Przykład	<code>\$count = 1; _n(`%d comment`, `%d comments`, \$count, `plugin-name`);</code>
Plik .po	<code>#: path/to/file.php:1 msgid "%d comment" msgid_plural "%d comments" msgstr[0] "%d komentarz" msgstr[1] "%d komentarze" msgstr[2] "%d komentarzy"</code>



Kontekst fraz

Definicje	<pre>_x(string, context, textDomain); _ex(string, context, textDomain); _nx(singular, plural, count, context, textDomain);</pre>
Przykłady	<pre>_x(`Post`, `noun`, `plugin-name`); _ex(`Post`, `verb`, `plugin-name`);</pre>
Plik .po	<pre>#: path/to/file:1 msgctxt "verb" msgid "Post" msgstr "Wpis"</pre>



Dodatkowe funkcje

Definicje	<pre>_n_noop(singular, plural, textDomain); _nx_noop(singular, plural, context, textDomain); translate_nooped_plural(translation, count, textDomain);</pre>
Przykład	<pre>\$noop = _n_noop(`%d comment`, `%d comments`, `my-plugin`); ... \$count = 1; echo translate_nooped_plural(\$noop, \$count, `my-plugin`);</pre>

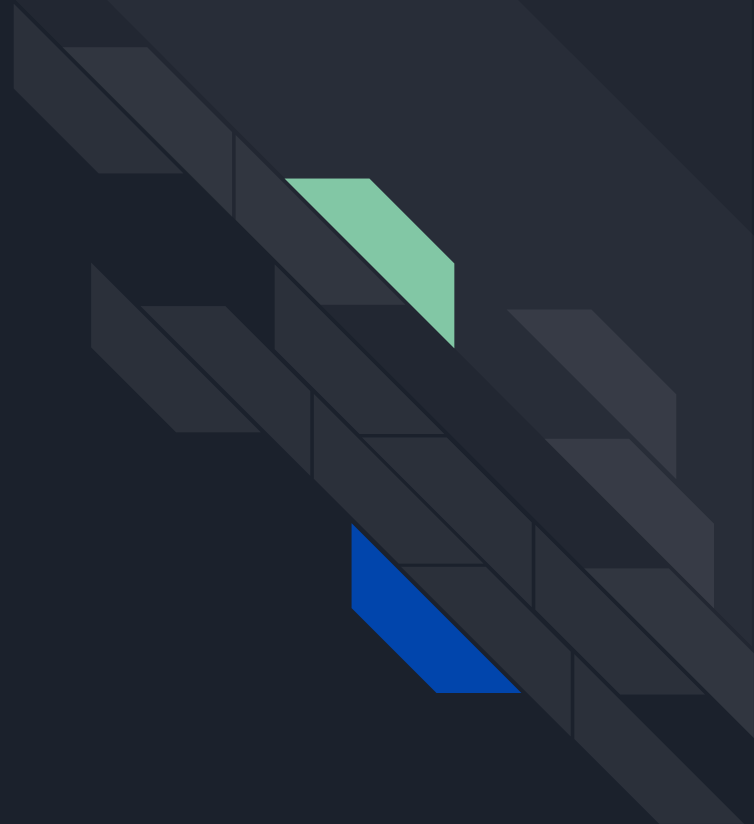


Funkcje sanityzujące

- `esc_html_()`
- `esc_html_e()`
- `esc_html_x()`
- `esc_attr_()`
- `esc_attr_e()`
- `esc_attr_x()`

Wszystkie frazy znajdujące się w atrybutach HTML powinny być sanityzowane.

Frazy ze zmiennymi





Frazy ze zmiennymi

```
echo 'You have' . $count . ' orders.';
```

```
echo "You have $count orders.";
```

```
echo "You have {$count} orders.";
```



Frazy ze zmiennymi

```
echo 'You have' . $count . ' orders.';
```

```
echo "You have $count orders.";
```

```
echo "You have {$count} orders.";
```



Frazy ze zmiennymi

```
$count = 5;
_e( "You have $count orders." );
// You have 5 orders.
```

- brak podziału ze względu na ilość (potrzebne warunki logiczne, problemy z obsługą różnych odmian językowych)
- w tłumaczonych frazach niepotrzebnie pojawia się nazwa zmiennej
- dajemy duże pole do opisu osobom które chciałyby nabroić w naszej wtyczce/motywie



printf i sprintf

```
printf(· 'You have %d orders.', · $count · );
```

```
echo sprintf(  
    'Your name is %s and you have %d orders.',  
    $name,  
    $count  
);
```

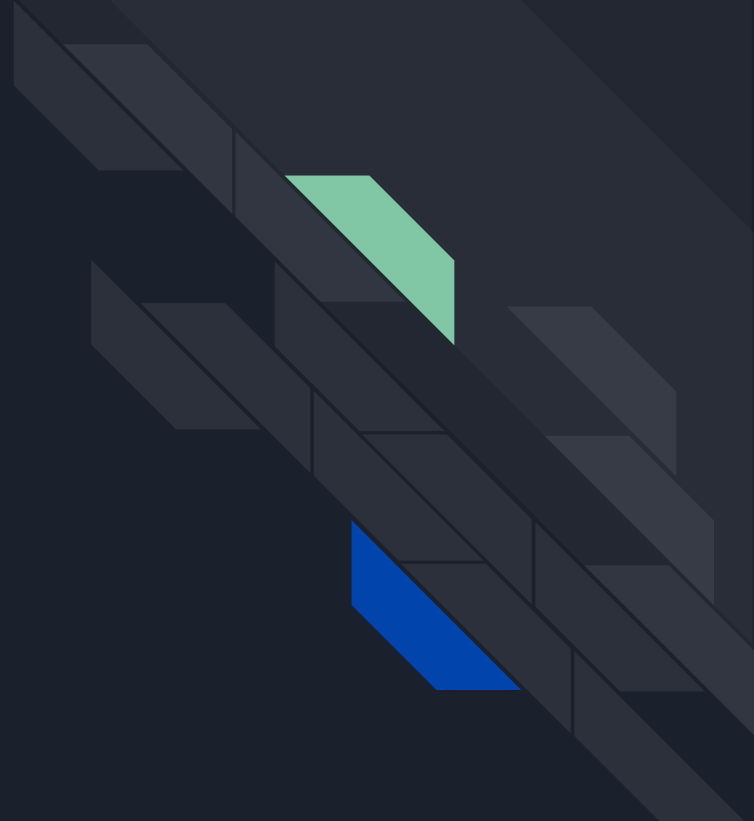
Dokumentacja: <http://php.net/manual/pl/function.sprintf.php>



printf + i18n

```
1 <?php
2 printf(
3     _n(
4         'You have %d order.',
5         'You have %d orders.',
6         $count,
7         'text-domain'
8     ),
9     $count
10 );
```


Dodatkowe informacje





Nagłówek Text Domain

WordPress korzysta z nagłówka Text Domain aby wyodrębnić tłumaczenia danej wtyczki nawet gdy jest ona wyłączona.

Ma to zastosowanie w tłumaczeniu podstawowych informacji o wtyczce. W skład tych informacji wchodzi jej nazwa, opis oraz autor.

```
1 <?php
2 /*
3  * Plugin Name: My plugin
4  * Version: 1.0
5  * Author: Me
6  * Text Domain: my-plugin
7  */
```

Nagłówek pliku my-plugin/my-plugin.php



Zasady text domain

1. Text domain musi być taki sam jak “slug” wtyczki lub motywu (nazwa głównego pliku lub folderu)
2. Spacje powinny być zastępowane myślnikami “-”, nie podkreślnikami “_”
3. Należy zawsze stosować text domain, nawet jeśli uważasz, że jest niepotrzebny
4. Nie wolno zapisywać text domain do zmiennej i wykorzystywać jej w funkcjach do tłumaczenia

Źródło:

<https://developer.wordpress.org/plugins/internationalization/how-to-internationalize-your-plugin/>



Text Domain w zmiennej

```
$textDomain = 'my-extra-long-and-unique-text-domain';  
__( 'Hello world!', $textDomain );  
__( 'Another world!', $textDomain );
```

O ile ten zapis jest poprawny, o tyle dokumentacja WordPressa zabrania takich praktyk. Należy za każdym razem pisać Text Domain, w każdej funkcji internacjonalizującej.

Źródło:

<https://developer.wordpress.org/plugins/internationalization/how-to-internationalize-your-plugin/>



Nagłówek Domain Path

```
1 <?php
2 /*
3  * Plugin Name: My plugin
4  * Version: 1.0
5  * Author: Me
6  * Text Domain: my-plugin
7  * Domain Path: /my-language-folder
8  */
```

Nagłówek Domain Path wskazuje na folder w którym znajdują się tłumaczenia danej wtyczki.

Jego zastosowanie jest takie samo jak Text Domain - WordPress korzysta z niego gdy wtyczka jest wyłączona.

Dziękuję za uwagę!





Krzysztof Grabania

<https://github.com/Dartui>

krzysztof@grabania.pl