

## Metodika k lokalizácii a prekladu

Pretože je program vytváraný pre Európsku úniu a jej štáty, je potrebný preklad stránky aj do iných jazykov. Základný jazyk, ktorým sa píše text v programe, je angličtina.

### Ako písať lokalizáciu v TEMPLATE súboroch

Na začiatok všetkých html súborov, za prípadnú definíciu rozšírenia je potrebné vložiť tag `{% load i18n %}`. Toto je potrebné aj v tých, ktoré rozširujú html súbor, v ktorom je už tento tag použitý. Následne môžeme označiť reťazec na preklad.

Pôvodný:

```
<title>Hello world</title>
```

Nový:

```
<title>{% trans "Hello world" %}</title>
```

Použili sme tag `{% trans %}`. Tým sme povedali, že tento reťazec sa bude prekladať. Tento tag sa da využiť aj pri premenných:

```
<title>{% trans myvar %}</title>
```

Existuje aj tag `{% blocktrans %}{% endblocktrans %}`, ktorým je možné prekladať celé vety, obsahujúce reťazce aj premenné.

Toto sú len základy, odporúčam pozrieť dokumentáciu:

<https://docs.djangoproject.com/en/1.11/topics/i18n/translation/#internationalization-in-template-code>

### Ako písať lokalizáciu v PYTHON súboroch

V python súboroch využijeme funkcie knižnice `django.utils.translation`, z ktorej použijeme funkciu `ugettext`.

```
from django.utils.translation import ugettext
```

Následne pri každom stringu, ktorý budeme chcieť prekladať využijeme `ugettext()`:

```
var = ugettext("Hello world")
```

Pre zjednodušenie a zrýchlenie písania použijeme substitúciu:

```
from django.utils.translation import ugettext as _
```

A ďalej píšeme už len:

```
var = _("Hello world")
```

Opakujem a silne odporúčam pozrieť si dokumentáciu, pretože sú tam vysvetlené ďalšie veci ako pluralizácia, kontextové markeri atď, ktoré sa možno nevyužijú, ale budete vedieť, že sa vôbec niečo také dá využiť, a že to knižnica podporuje.

<https://docs.djangoproject.com/en/1.11/topics/i18n/translation/#internationalization-in-python-code>

## Ako aktualizovať/vytvoriť message súboru

Vytvorenie a aktualizovanie message súboru je veľmi podobné, v niektorých krokoch identické. Message súbor je v tomto prípade aj vytvorený. Ale čo ak chceme pridať nový jazyk?

### Vytvorenie:

1. Najprv je potrebné si stiahnuť *gettext*. Návod nájdete na tejto stránke pod nadpisom „gettext On Windows“:

<https://djangobook.com/localization-create-language-files>

2. Následne vytvoríme nový priečinok s názvom “locale“ v priečinku “my\_app”.
3. Stlačíme ctrl+alt+r a do príkazového riadku napíšeme

```
makemessages -l sk_Sk
```

Namiesto sk\_SK dajte požadovaný jazyk. Týmto sa vytvorí .po súbor, ktorý obsahuje všetky stringy na preklad:

```
msgid "Hello world"
```

```
msgstr ""
```

Všade je potrebné pridať preklad do požadovaného jazyka:

```
msgid "Hello world"
```

**msgstr** "Ahoj svet"

4. Ďalej je potrebné skompilovať .po súbor. Využijeme príkaz, ktorý zadáme do príkazového riadku:

*compilemessages*

To vytvorí .mo súbor. Následne by mal byť preklad viditeľný aj v aplikácii.

### **Aktualizovanie:**

Podobné ako pri vytváraní, začíname bodom číslo 3.

Niečo bližšie k message súborom:

<https://djangobook.com/localization-create-language-files>