
django-localflavor Documentation

Release 1.0

Django Software Foundation and individual contributors

July 29, 2013

CONTENTS

1	Installation	3
2	Internationalization	5
3	Adding flavors	7
4	Releases	9
5	How to migrate	11
6	Backwards compatibility	13
7	Indices and tables	15
7.1	Argentina (ar)	15
7.2	Austria (at)	15
7.3	Australia (au)	16
7.4	Belgium (be)	17
7.5	Brazil (br)	18
7.6	Canada (ca)	18
7.7	Switzerland (ch)	19
7.8	Chile (cl)	20
7.9	China (cn)	21
7.10	Columbia (co)	22
7.11	Czech Republic (cz)	22
7.12	Germany (de)	23
7.13	Ecuador (ec)	23
7.14	Spain (es)	24
7.15	Finnland (fi)	25
7.16	France (fr)	25
7.17	Great Britain (gb)	26
7.18	Hong Kong (hk)	27
7.19	Croatia (hr)	27
7.20	Indonesia (id)	29
7.21	Ireland (ie)	30
7.22	Israel (il)	30
7.23	India (in)	31
7.24	Iceland (is)	31
7.25	Italy (it)	32
7.26	Japan (jp)	33
7.27	Kuwait (kw)	33

7.28	Lithuania (lt)	33
7.29	Macedonia (mk)	34
7.30	Mexico (mx)	35
7.31	The Netherlands (nl)	37
7.32	Norway (no)	38
7.33	Peru (pe)	38
7.34	Poland (pl)	39
7.35	Portugal (pt)	40
7.36	Paraguay (py)	41
7.37	Romania (ro)	41
7.38	Russia (ru)	43
7.39	Sweden (se)	43
7.40	Slovenia (si)	44
7.41	Slovakia (sk)	45
7.42	Turkey (tr)	46
7.43	United States of America (us)	46
7.44	Urguay (uy)	48
7.45	South Africa (za)	49
7.46	Generic helpers	49

Python Module Index	51
----------------------------	-----------

django-localflavor is a collection of assorted pieces of code that are useful for particular countries or cultures. These are called the “local flavor” add-ons and live in the `localflavor` package.

Inside that package, country- or culture-specific code is organized into subpackages, named using [ISO 3166 country codes](#).

Most of the `localflavor` add-ons are localized form components deriving from the forms framework – for example, a `USStateField` that knows how to validate U.S. state abbreviations, and a `FISocialSecurityNumber` that knows how to validate Finnish social security numbers.

- *Argentina (ar)*
- *Austria (at)*
- *Australia (au)*
- *Belgium (be)*
- *Brazil (br)*
- *Canada (ca)*
- *Switzerland (ch)*
- *Chile (cl)*
- *China (cn)*
- *Columbia (co)*
- *Czech Republic (cz)*
- *Germany (de)*
- *Ecuador (ec)*
- *Spain (es)*
- *Finnland (fi)*
- *France (fr)*
- *Great Britain (gb)*
- *Hong Kong (hk)*
- *Croatia (hr)*
- *Indonesia (id)*
- *Ireland (ie)*
- *Israel (il)*
- *India (in)*
- *Iceland (is)*
- *Italy (it)*
- *Japan (jp)*
- *Kuwait (kw)*
- *Lithuania (lt)*
- *Macedonia (mk)*
- *Mexico (mx)*
- *The Netherlands (nl)*
- *Norway (no)*
- *Peru (pe)*
- *Poland (pl)*
- *Portugal (pt)*
- *Paraguay (py)*
- *Romania (ro)*
- *Russia (ru)*
- *Sweden (se)*
- *Slovenia (si)*
- *Slovakia (sk)*
- *Turkey (tr)*
- *United States of America (us)*
- *Uruguay (uy)*
- *South Africa (za)*

To use one of these localized components, just import the relevant subpackage. For example, here's how you can create a form with a field representing a French telephone number:

```
from django import forms
from localflavor.fr.forms import FRPhoneNumberField

class MyForm(forms.Form):
    my_french_phone_no = FRPhoneNumberField()
```

The `localflavor` package also includes a *generic* subpackage, containing useful code that is not specific to one particular country or culture. Currently, it defines date, datetime and split datetime input fields based on those from the forms, but with non-US default formats. Here's an example of how to use them:

```
from django import forms
from localflavor import generic

class MyForm(forms.Form):
    my_date_field = generic.forms.DateField()
```

INSTALLATION

To install django-localflavor use your favorite packaging tool, e.g.pip:

```
pip install django-localflavor
```

Or download the source distribution from PyPI at <https://pypi.python.org/pypi/django-localflavor>, decompress the file and run `python setup.py install` in the unpacked directory.

Then add 'localflavor' to your `INSTALLED_APPS` setting:

```
INSTALLED_APPS = (  
    # ...  
    'localflavor',  
)
```


INTERNATIONALIZATION

Localflavor has its own catalog of translations, in the directory `localflavor/locale`, and it's not loaded automatically like Django's general catalog in `django/conf/locale`. If you want localflavor's texts to be translated, like form fields error messages, you must include `localflavor` in the `INSTALLED_APPS` setting, so the internationalization system can find the catalog, as explained in *How Django discovers translations*.

ADDING FLAVORS

We'd love to add more of these, so please [create a ticket](#) with any code you'd like to contribute. One thing we ask is that you please use Unicode objects (`u'mystring'`) for strings, rather than setting the encoding in the file. See any of the existing flavors for examples.

If you consider adding a new localflavor for country here are some examples that you might consider implementing:

- form fields and form widgets
 - ID verification
 - tax or social security number validator
 - car registration
 - zip code validation
 - phone number validation
 - country area selects, e.g. cities, counties, states, provinces
- model fields, e.g. for storing any of the above form fields' values
- local translations of English area names. Join your language team at Transifex:
<https://www.transifex.com/projects/p/django-localflavor/>

RELEASES

Due to django-localflavor' history as a former contrib app, the app is required to be working with the actively maintained Django versions. See the documentation about [Django's release process](#) for more information.

django-localflavor releases are not tied to the release cycle of Django. Version numbers follow the appropriate Python standards, e.g. PEPs [386](#) and [440](#).

HOW TO MIGRATE

If you've used the old `django.contrib.localflavor` package or one of the temporary `django-localflavor-*` releases, follow these two easy steps to update your code:

1. Install the third-party `django-localflavor` package.
2. Change your app's import statements to reference the new packages.

For example, change this:

```
from django.contrib.localflavor.fr.forms import FRPhoneNumberField
```

...to this:

```
from localflavor.fr.forms import FRPhoneNumberField
```

Or if you used one of the shortlived `django-localflavor-*` packages change:

```
from django_localflavor_fr.forms import FRPhoneNumberField
```

...to this:

```
from localflavor.fr.forms import FRPhoneNumberField
```

The code in the new package is the same (it was copied directly from Django), so you don't have to worry about backwards compatibility in terms of functionality. Only the imports have changed.

BACKWARDS COMPATIBILITY

We will always attempt to make `localflavor` reflect the officially gazetted policies of the appropriate local government authority. For example, if a government body makes a change to add, alter, or remove a province (or state, or county), that change will be reflected in `localflavor` in the next release.

When a backwards-incompatible change is made (for example, the removal or renaming of a province) the `localflavor` in question will raise a warning when that `localflavor` is imported. This provides a runtime indication that something may require attention.

However, once you have addressed the backwards compatibility (for example, auditing your code to see if any data migration is required), the warning serves no purpose. The warning can then be suppressed. For example, to suppress the warnings raised by the Indonesian `localflavor` you would use the following code:

```
import warnings
warnings.filterwarnings('ignore',
                        category=RuntimeWarning,
                        module='localflavor.id')
from localflavor.id import forms as id_forms
```


INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

7.1 Argentina (ar)

AR-specific Form helpers.

class `localflavor.ar.forms.ARCUITField` (*max_length=None, min_length=None, *args, **kwargs*)

This field validates a CUIT (Código Único de Identificación Tributaria). A CUIT is of the form XX-XXXXXXXX-V. The last digit is a check digit.

clean (*value*)

Value can be either a string in the format XX-XXXXXXXX-X or an 11-digit number.

class `localflavor.ar.forms.ARDNIField` (*max_length=10, min_length=7, *args, **kwargs*)

A field that validates 'Documento Nacional de Identidad' (DNI) numbers.

clean (*value*)

Value can be a string either in the [X]X.XXX.XXX or [X]XXXXXXXX formats.

class `localflavor.ar.forms.ARPostalCodeField` (*max_length=8, min_length=4, *args, **kwargs*)

A field that accepts a 'classic' NNNN Postal Code or a CPA.

See: http://www.correoargentino.com.ar/cpa/que_es http://www.correoargentino.com.ar/cpa/como_escribirlo

class `localflavor.ar.forms.ARProvinceSelect` (*attrs=None*)

A Select widget that uses a list of Argentinean provinces/autonomous cities as its choices.

`localflavor.ar.ar_provinces.PROVINCE_CHOICES = (('B', 'Buenos Aires'), ('K', 'Catamarca'), ('H', 'Chaco'), ('U', 'Chubut'), ('D', 'Córdoba'), ('F', 'Formosa'), ('G', 'Gérola'), ('I', 'Isla de Tierra del Fuego'), ('J', 'Jujuy'), ('L', 'La Pampa'), ('M', 'Mendoza'), ('N', 'Neuquén'), ('P', 'Palmar Grande'), ('Q', 'Quilmes'), ('R', 'Río Negro'), ('S', 'Salta'), ('T', 'Tucumán'), ('V', 'Villa María'), ('W', 'Wanda')`
A list of Argentinean provinces and autonomous cities as *choices* in a formfield. From <http://www.argentina.gov.ar/argentina/portal/paginas.dhtml?pagina=425>

7.2 Austria (at)

AT-specific Form helpers

```
class localflavor.at.forms.ATSocialSecurityNumberField(required=True, widget=
                                                    get=None, label=None, ini-
                                                    tial=None, help_text=None,
                                                    error_messages=None,
                                                    show_hidden_initial=False,
                                                    validators=[], localize=False)
```

Austrian Social Security numbers are composed of a 4 digits and 6 digits field. The latter represents in most cases the person's birthdate while the first 4 digits represent a 3-digits counter and a one-digit checksum.

The 6-digits field can also differ from the person's birthdate if the 3-digits counter suffered an overflow.

This code is based on information available on <http://de.wikipedia.org/wiki/Sozialversicherungsnummer#.C3.96sterreich>

```
class localflavor.at.forms.ATStateSelect(attrs=None)
    A Select widget that uses a list of AT states as its choices.
```

```
class localflavor.at.forms.ATZipCodeField(max_length=None, min_length=None, *args,
                                          **kwargs)
    A form field that validates its input is an Austrian postcode.
```

Accepts 4 digits (first digit must be greater than 0).

```
localflavor.at.at_states.STATE_CHOICES = (('BL', 'Burgenland'), ('KA', 'Carinthia'), ('NO', 'Lower Austria'), ('O
```

A list of Austrian states according to https://en.wikipedia.org/wiki/States_of_Austria

7.3 Australia (au)

7.3.1 Forms

Australian-specific Form helpers

```
class localflavor.au.forms.AUPhoneNumberField(max_length=None, min_length=None, *args,
                                              **kwargs)
```

A form field that validates input as an Australian phone number.

Valid numbers have ten digits.

```
    clean (value)
```

Validate a phone number. Strips parentheses, whitespace and hyphens.

```
class localflavor.au.forms.AUPostCodeField(max_length=4, min_length=None, *args,
                                           **kwargs)
```

Australian post code field.

Assumed to be 4 digits. Northern Territory 3-digit postcodes should have leading zero.

```
class localflavor.au.forms.AUStateSelect(attrs=None)
```

A Select widget that uses a list of Australian states/territories as its choices.

7.3.2 Models

```
class localflavor.au.models.AUPhoneNumberField(*args, **kwargs)
```

A model field that checks that the value is a valid Australian phone number (ten digits).

```
    description = <django.utils.functional.__proxy__ object at 0x4de31d0>
```

```
    formfield (**kwargs)
```

```
class localflavor.au.models.AUPostCodeField(*args, **kwargs)
    A model field that forms represent as a AUPostCodeField field and stores the four-digit Australian postcode
    in the database.

    description = <django.utils.functional.__proxy__ object at 0x4de3190>

    formfield(**kwargs)
```

```
class localflavor.au.models.AUStateField(*args, **kwargs)
    A model field that is represented with STATE_CHOICES `choices` and stores the three-letter Australian state
    abbreviation in the database.

    description = <django.utils.functional.__proxy__ object at 0x4de3150>
```

7.3.3 Data

```
localflavor.au.au_states.STATE_CHOICES = (('ACT', 'Australian Capital Territory'), ('NSW', 'New South Wales'),
    An alphabetical list of states for use as choices in a formfield.
```

7.4 Belgium (be)

7.4.1 Forms

Belgium-specific Form helpers

```
class localflavor.be.forms.BEPhoneNumberField(max_length=None, min_length=None, *args,
    **kwargs)
    A form field that validates its input as a belgium phone number.

    Landlines have a seven-digit subscriber number and a one-digit area code, while smaller cities have a six-digit
    subscriber number and a two-digit area code. Cell phones have a six-digit subscriber number and a two-digit area
    code preceeded by the number 4. 0d ddd dd dd, 0d/ddd.dd.dd, 0d.ddd.dd.dd, 0ddddd - dialling a bigger city
    Odd dd dd dd, Odd/dd.dd.dd, Odd.dd.dd.dd, 0ddddd - dialling a smaller city 04dd ddd dd dd, 04dd/ddd.dd.dd,
    04dd.ddd.dd.dd, 04ddddd - dialling a mobile number
```

```
class localflavor.be.forms.BEPostalCodeField(max_length=None, min_length=None, *args,
    **kwargs)
    A form field that validates its input as a belgium postal code.

    Belgium postal code is a 4 digits string. The first digit indicates the province (except for the 3ddd numbers that
    are shared by the eastern part of Flemish Brabant and Limburg and the and 1ddd that are shared by the Brussels
    Capital Region, the western part of Flemish Brabant and Walloon Brabant)
```

```
class localflavor.be.forms.BEProvinceSelect(attrs=None)
    A Select widget that uses a list of belgium provinces as its choices.
```

```
class localflavor.be.forms.BERegionSelect(attrs=None)
    A Select widget that uses a list of belgium regions as its choices.
```

7.4.2 Data

```
localflavor.be.be_provinces.PROVINCE_CHOICES = (('VAN', 'Antwerp'), ('BRU', 'Brussels'), ('VOV', 'East Flande
    ISO codes
```

```
localflavor.be.be_regions.REGION_CHOICES = (('BRU', 'Brussels Capital Region'), ('VLG', 'Flemish Region'), ('W
    ISO codes
```

7.5 Brazil (br)

7.5.1 Forms

BR-specific Form helpers

```
class localflavor.br.forms.BRCNPJField(required=True, widget=None, label=None, initial=None, help_text=None, error_messages=None, show_hidden_initial=False, validators=[], localize=False)
```

A form field that validates input as [Brazilian CNPJ](#).

Input can either be of the format XX.XXX.XXX/XXXX-XX or be a group of 14 digits.

clean (value)

Value can be either a string in the format XX.XXX.XXX/XXXX-XX or a group of 14 characters.

```
class localflavor.br.forms.BRCPFField(max_length=14, min_length=11, *args, **kwargs)
```

A form field that validates a CPF number or a CPF string. A CPF number is compounded by XXX.XXX.XXX-VD. The two last digits are check digits.

More information: http://en.wikipedia.org/wiki/Cadastro_de_Pessoas_F%C3%ADsicas

clean (value)

Value can be either a string in the format XXX.XXX.XXX-XX or an 11-digit number.

```
class localflavor.br.forms.BRPhoneNumberField(required=True, widget=None, label=None, initial=None, help_text=None, error_messages=None, show_hidden_initial=False, validators=[], localize=False)
```

A form field that validates input as a Brazilian phone number, with the format XX-XXXX-XXXX.

```
class localflavor.br.forms.BRStateChoiceField(required=True, widget=None, label=None, initial=None, help_text=None)
```

A choice field that uses a list of Brazilian states as its choices.

```
class localflavor.br.forms.BRStateSelect(attrs=None)
```

A Select widget that uses a list of Brazilian states/territories as its choices.

```
class localflavor.br.forms.BRZipCodeField(max_length=None, min_length=None, *args, **kwargs)
```

A form field that validates input as a Brazilian zip code, with the format XXXXX-XXX.

7.5.2 Data

```
localflavor.br.br_states.STATE_CHOICES = (('AC', 'Acre'), ('AL', 'Alagoas'), ('AP', 'Amapá'), ('AM', 'Amazonas'))
```

An alphabetical list of Brazilian states for use as *choices* in a formfield

7.6 Canada (ca)

7.6.1 Forms

Canada-specific Form helpers

```
class localflavor.ca.forms.CAPhoneNumberField(required=True, widget=None,
                                              label=None, initial=None,
                                              help_text=None, error_messages=None,
                                              show_hidden_initial=False, validators=[],
                                              localize=False)
```

Canadian phone number form field.

```
class localflavor.ca.forms.CAPostalCodeField(max_length=None, min_length=None, *args,
                                              **kwargs)
```

Canadian postal code form field.

Validates against known invalid characters: D, F, I, O, Q, U Additionally the first character cannot be Z or W.
For more info see: <http://www.canadapost.ca/tools/pg/manual/PGaddress-e.asp#1402170>

```
class localflavor.ca.forms.CAProvinceField(required=True, widget=None, label=None,
                                           initial=None, help_text=None, error_messages=None,
                                           show_hidden_initial=False, validators=[], localize=False)
```

A form field that validates its input is a Canadian province name or abbreviation. It normalizes the input to the standard two-letter postal service abbreviation for the given province.

```
class localflavor.ca.forms.CAProvinceSelect(attrs=None)
```

A Select widget that uses a list of Canadian provinces and territories as its choices.

```
class localflavor.ca.forms.CASocialInsuranceNumberField(required=True, widget=None,
                                                         label=None, initial=None,
                                                         help_text=None, error_messages=None,
                                                         show_hidden_initial=False,
                                                         validators=[], localize=False)
```

A Canadian Social Insurance Number (SIN).

Checks the following rules to determine whether the number is valid:

- Conforms to the XXX-XXX-XXX format.
- Passes the check digit process “Luhn Algorithm” See: http://en.wikipedia.org/wiki/Social_Insurance_Number

```
luhn_checksum_is_valid(number)
```

Checks to make sure that the SIN passes a luhn mod-10 checksum See: http://en.wikipedia.org/wiki/Luhn_algorithm

7.6.2 Data

```
localflavor.ca.ca_provinces.PROVINCE_CHOICES = (('AB', 'Alberta'), ('BC', 'British Columbia'), ('MB', 'Manitoba'), ('ON', 'Ontario'), ('QC', 'Quebec'), ('SK', 'Saskatchewan'), ('YT', 'Yukon'))
```

An alphabetical list of provinces and territories for use as *choices* in a formfield. Source: http://www.canada.gc.ca/othergov/prov_e.html

```
localflavor.ca.ca_provinces.PROVINCES_NORMALIZED = {'yk': 'YT', 'ab': 'AB', 'bc': 'BC', 'pq': 'QC', 'ontario': 'ON', 'sk': 'SK', 'manitoba': 'MB', 'alberta': 'AB', 'quebec': 'QC', 'saskatchewan': 'SK', 'yukon': 'YT'}
```

a mapping of province misspellings/abbreviations to normalized abbreviations

7.7 Switzerland (ch)

7.7.1 Forms

Swiss-specific Form helpers

```
class localflavor.ch.forms.CHIdentityCardNumberField(required=True, widget=None,
                                                    get=None, label=None, initial=None,
                                                    tial=None, help_text=None,
                                                    error_messages=None,
                                                    show_hidden_initial=False,
                                                    validators=[], localize=False)
```

A Swiss identity card number.

Checks the following rules to determine whether the number is valid:

- Conforms to the X1234567<0 or 1234567890 format.
- Included checksums match calculated checksums

```
class localflavor.ch.forms.CHPhoneNumberField(required=True, widget=None,
                                              label=None, initial=None,
                                              help_text=None, error_messages=None,
                                              show_hidden_initial=False, validators=[],
                                              localize=False)
```

Validate local Swiss phone number (not international ones) The correct format is '0XX XXX XX XX'. '0XX.XXX.XX.XX' and '0XXXXXXXXX' validate but are corrected to '0XX XXX XX XX'.

```
class localflavor.ch.forms.CHStateSelect(attrs=None)
```

A Select widget that uses a list of CH states as its choices.

```
class localflavor.ch.forms.CHZipCodeField(max_length=None, min_length=None, *args,
                                          **kwargs)
```

A form field that validates input as a Swiss zip code. Valid codes consist of four digits.

7.7.2 Data

```
localflavor.ch.ch_states.STATE_CHOICES = (('AG', 'Aargau'), ('AI', 'Appenzell Innerrhoden'), ('AR', 'Appenzell A
```

An alphabetical list of states

7.8 Chile (cl)

7.8.1 Forms

Chile specific form helpers.

```
class localflavor.cl.forms.CLRegionSelect(attrs=None)
```

A Select widget that uses a list of Chilean Regions (Regiones) as its choices.

```
class localflavor.cl.forms.CLRutField(*args, **kwargs)
```

Chilean "Rol Unico Tributario" (RUT) field. This is the Chilean national identification number.

Samples for testing are available from https://palena.sii.cl/cvc/dte/ee_empresas_emisoras.html

```
clean(value)
```

Check and clean the Chilean RUT.

7.8.2 Data

```
localflavor.cl.cl_regions.REGION_CHOICES = (('RM', 'Región Metropolitana de Santiago'), ('I', 'Región de Tarapa
```

A list of Chilean regions as *choices* in a formfield.

7.9 China (cn)

7.9.1 Forms

Chinese-specific form helpers

```
class localflavor.cn.forms.CNProvinceSelect (attrs=None)
    A select widget with list of Chinese provinces as choices.
```

```
class localflavor.cn.forms.CNPPostCodeField(*args,**kwargs)
    A form field that validates as Chinese post code. Valid code is XXXXXX where X is digit.
```

```
class localflavor.cn.forms.CNIDCardField(max_length=18, min_length=15, *args, **kwargs)
    A form field that validates as Chinese Identification Card Number.
```

This field would check the following restrictions:

- the length could only be 15 or 18.
- if the length is 18, the last digit could be x or X.
- has a valid checksum.(length 18 only)
- has a valid birthdate.
- has a valid location.

The checksum algorithm is described in GB11643-1999.

clean (*value*)
Check whether the input is a valid ID Card Number.

has_valid_birthday (*value*)

This function would grab the birthdate from the ID card number and test whether it is a valid date.

has_valid_checksum(value)

This method checks if the last letter/digit in value is valid according to the algorithm the ID Card follows.

has_valid_location (*value*)

This method checks if the first two digits in the ID Card are valid.

```
class localflavor.cn.forms.CNPhoneNumberField(*args,**kwargs)
    A form field that validates as Chinese phone number
```

A valid phone number could be like: 010-55555555

Considering there might be extension phone numbers, so this could also be: 010-55555555-35

```
class localflavor.cn.forms.CNCellNumberField(*args, **kwargs)
    A form field that validates as Chinese cell number
```

A valid cell number could be like: 13012345678

We used a rough rule here, the first digit should be 1, the second could be 3, 5 and 8, the rest could be what so ever. The length of the cell number should be 11.

7.9.2 Data

`localflavor.cn.cn_provinces.CN_PROVINCE_CHOICES = (('anhui', ''), ('beijing', ''), ('chongqing', ''), ('fujian', ''))`

An alphabetical list of provinces for use as *choices* in a formfield. http://en.wikipedia.org/wiki/ISO_3166-2:CN http://en.wikipedia.org/wiki/Province_%28China%29 http://en.wikipedia.org/wiki/Direct-controlled_municipality http://en.wikipedia.org/wiki/Autonomous_regions_of_China

7.10 Columbia (co)

7.10.1 Forms

Colombian-specific form helpers.

class localflavor.co.forms.CODepartmentSelect (*attrs=None*)
A Select widget that uses a list of Colombian states as its choices.

7.10.2 Data

localflavor.co.co_departments.DEPARTMENT_CHOICES = (('AMA', 'Amazonas'), ('ANT', 'Antioquia'), ('ARA', 'Ar'), ('BOG', 'Bogotá'), ('BOY', 'Boyacá'), ('CAL', 'Caldas'), ('CAQ', 'Cauca'), ('CES', 'Cesar'), ('CHB', 'Chibcha'), ('COR', 'Córdoba'), ('CUN', 'Cundinamarca'), ('GUA', 'Guaviare'), ('HUI', 'Huila'), ('MAG', 'Magdalena'), ('MET', 'Meta'), ('NAR', 'Nariño'), ('NOR', 'Noronja'), ('PUT', 'Putumayo'), ('QUI', 'Quindío'), ('RIS', 'Risaralda'), ('SAB', 'Santander'), ('SAN', 'Santafé de Bogotá'), ('SUC', 'Sucre'), ('TOL', 'Tolima'), ('VAL', 'Valle del Cauca'), ('VIG', 'Vichaco'), ('VIR', 'Viracocha'), ('ZUL', 'Zulia'))
A list of Colombian departaments as *choices* in a formfield.

7.11 Czech Republic (cz)

7.11.1 Forms

Czech-specific form helpers

class localflavor.cz.forms.CZBirthNumberField (*required=True, widget=None, label=None, initial=None, help_text=None, error_messages=None, show_hidden_initial=False, validators=[], localize=False*)
Czech birth number form field.

class localflavor.cz.forms.CZICNumberField (*required=True, widget=None, label=None, initial=None, help_text=None, error_messages=None, show_hidden_initial=False, validators=[], localize=False*)
Czech IC number form field.

class localflavor.cz.forms.CZPostalCodeField (*max_length=None, min_length=None, *args, **kwargs*)
A form field that validates its input as Czech postal code. Valid form is XXXXX or XXX XX, where X represents integer.
clean (*value*)
Validates the input and returns a string that contains only numbers. Returns an empty string for empty values.

class localflavor.cz.forms.CZRegionSelect (*attrs=None*)
A select widget widget with list of Czech regions as choices.

7.11.2 Data

localflavor.cz.cz_regions.REGION_CHOICES = (('PR', 'Prague'), ('CE', 'Central Bohemian Region'), ('SO', 'South Bohemian Region'), ('VY', 'Vysočina Region'), ('ST', 'Středočeský území'), ('MZ', 'Moravskoslezský území'), ('LI', 'Liberecký území'), ('PA', 'Pardubický území'), ('ST', 'Středočeský území'), ('MZ', 'Moravskoslezský území'), ('LI', 'Liberecký území'), ('PA', 'Pardubický území'))
Czech regions, translations get from <http://www.crwflags.com/fotw/Flags/cz-re.html>

7.12 Germany (de)

7.12.1 Forms

DE-specific Form helpers

```
class localflavor.de.forms.DEIdentityCardNumberField(required=True,          wid-
                                                    get=None,   label=None,   ini-
                                                    tial=None,   help_text=None,
                                                    error_messages=None,
                                                    show_hidden_initial=False,
                                                    validators=[], localize=False)
```

A German identity card number.

Checks the following rules to determine whether the number is valid:

- Conforms to the XXXXXXXXXXX-XXXXXXX-XXXXXXX-X format.
- No group consists entirely of zeroes.
- Included checksums match calculated checksums

Algorithm is documented at <http://de.wikipedia.org/wiki/Personalausweis>

```
class localflavor.de.forms.DEStateSelect(attrs=None)
```

A Select widget that uses a list of DE states as its choices.

```
class localflavor.de.forms.DEZipCodeField(max_length=None,   min_length=None,   *args,
                                           **kwargs)
```

A form field that validates input as a German zip code. Valid codes consist of five digits.

7.12.2 Data

```
localflavor.de.de_states.STATE_CHOICES = (('BW', 'Baden-Wuerttemberg'), ('BY', 'Bavaria'), ('BE', 'Berlin'), ('B
```

An alphabetical list of states

7.13 Ecuador (ec)

7.13.1 Forms

Ecuador-specific form helpers.

```
class localflavor.ec.forms.ECProvinceSelect(attrs=None)
```

A Select widget that uses a list of Ecuador provinces as its choices.

7.13.2 Data

```
localflavor.ec.ec_provinces.PROVINCE_CHOICES = (('A', 'Azua'), ('B', 'Bolívar'), ('F', 'Cañar'), ('C', 'Carchi'), ('
```

A list of Ecuador departaments as *choices* in a formfield.

7.14 Spain (es)

7.14.1 Forms

Spanish-specific Form helpers

class `localflavor.es.forms.ESCCCFIELD` (*max_length=None, min_length=None, *args, **kwargs*)
A form field that validates its input as a Spanish bank account or CCC (Codigo Cuenta Cliente).

Spanish CCC is in format EEEE-OOOO-CC-AAAAAAAAAAAA where:

E = entity O = office C = checksum A = account

It's also valid to use a space as delimiter, or to use no delimiter.

First checksum digit validates entity and office, and last one validates account. Validation is done multiplying every digit of 10 digit value (with leading 0 if necessary) by number in its position in string 1, 2, 4, 8, 5, 10, 9, 7, 3, 6. Sum resulting numbers and extract it from 11. Result is checksum except when 10 then is 1, or when 11 then is 0.

TODO: allow IBAN validation too

class `localflavor.es.forms.ESIdentityCardNumberField` (*only_nif=False, max_length=None, min_length=None, *args, **kwargs*)

Spanish NIF/NIE/CIF (Fiscal Identification Number) code.

Validates three diferent formats:

NIF (individuals): 12345678A CIF (companies): A12345678 NIE (foreigners): X12345678A

according to a couple of simple checksum algorithms.

Value can include a space or hyphen separator between number and letters. Number length is not checked for NIF (or NIE), old values start with a 1, and future values can contain digits greater than 8. The CIF control digit can be a number or a letter depending on company type. Algorithm is not public, and different authors have different opinions on which ones allows letters, so both validations are assumed true for all types.

class `localflavor.es.forms.ESPhoneNumberField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates its input as a Spanish phone number. Information numbers are ommited.

Spanish phone numbers are nine digit numbers, where first digit is 6 (for cell phones), 8 (for special phones), or 9 (for landlines and special phones)

TODO: accept and strip characters like dot, hyphen... in phone number

class `localflavor.es.forms.ESPostalCodeField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates its input as a spanish postal code.

Spanish postal code is a five digits string, with two first digits between 01 and 52, assigned to provinces code.

class `localflavor.es.forms.ESProvinceSelect` (*attrs=None*)

A Select widget that uses a list of spanish provinces as its choices.

class `localflavor.es.forms.ESRegionSelect` (*attrs=None*)

A Select widget that uses a list of spanish regions as its choices.

7.14.2 Data

`localflavor.es.es_provinces.PROVINCE_CHOICES = (('01', 'Araba'), ('02', 'Albacete'), ('03', 'Alacant'), ('04', 'Alm`
 An alphabetical list of provinces

`localflavor.es.es_regions.REGION_CHOICES = (('AN', 'Andalusia'), ('AR', 'Aragon'), ('O', 'Principality of Asturias`
 An alphabetical list of regions

7.15 Finland (fi)

7.15.1 Forms

FI-specific Form helpers

class `localflavor.fi.forms.FIMunicipalitySelect` (*attrs=None*)
 A Select widget that uses a list of Finnish municipalities as its choices.

class `localflavor.fi.forms.FISocialSecurityNumber` (*required=True*, *wid-*
get=None, *label=None*, *ini-*
tial=None, *help_text=None*,
error_messages=None,
show_hidden_initial=False, *val-*
idators=[], *localize=False*)

A form field that validates input as a Finnish social security number.

class `localflavor.fi.forms.FIZipCodeField` (*max_length=None*, *min_length=None*, **args*,
***kwargs*)
 A form field that validates input as a Finnish zip code. Valid codes consist of five digits.

7.15.2 Data

`localflavor.fi.fi_municipalities.MUNICIPALITY_CHOICES = (('akaa', 'Akaa'), ('alajarvi', 'Alajärvi'), ('alaviesk`
 An alphabetical list of Finnish municipalities for use as *choices* in a formfield.

7.16 France (fr)

7.16.1 Forms

FR-specific Form helpers

class `localflavor.fr.forms.FRDepartmentField` (**args*, ***kwargs*)
 A Select Field that uses a FRDepartmentSelect widget.

widget
 alias of `FRDepartmentSelect`

class `localflavor.fr.forms.FRDepartmentSelect` (*attrs=None*)
 A Select widget that uses a list of FR departments as its choices.

class `localflavor.fr.forms.FRPhoneNumberField` (*max_length=14*, *min_length=10*, **args*,
***kwargs*)
 Validate local French phone number (not international ones) The correct format is '0X XX XX XX XX'.
 '0X.XX.XX.XX.XX' and '0XXXXXXXXXX' validate but are corrected to '0X XX XX XX XX'.

```
class localflavor.fr.forms.FRRegionField(*args, **kwargs)
    A Select Field that uses a FRRegionSelect widget.
```

widget
alias of `FRRegionSelect`

```
class localflavor.fr.forms.FRRegionSelect (attrs=None)
    A Select widget that uses a list of FR Regions as its choices.
```

```
class localflavor.fr.forms.FRZipCodeField(max_length=5, min_length=5, *args, **kwargs)
    Validate local French zip code. The correct format is 'XXXX'.
```

7.16.2 Data

localflavor.fr.fr_department.DEPARTMENT_CHOICES_PER_REGION=((('01', 'Ain', '82'), ('02', 'Aisne', '22'), ('03', 'Allier', '23'), ('04', 'Alpes de Haute-Provence', '24'), ('05', 'Hautes-Alpes', '25'), ('06', 'Alpes-Maritimes', '26'), ('07', 'Ardennes', '27'), ('08', 'Ardennes', '28'), ('09', 'Ariège', '29'), ('10', 'Aube', '30'), ('11', 'Aube', '31'), ('12', 'Aveyron', '32'), ('13', 'Bouches-du-Rhône', '33'), ('14', 'Calvados', '34'), ('15', 'Cantal', '35'), ('16', 'Charente-Maritime', '36'), ('17', 'Charente-Maritime', '37'), ('18', 'Cher', '38'), ('19', 'Corrèze', '39'), ('20', 'Corse-du-Sud', '40'), ('21', 'Côte-d'Or', '41'), ('22', 'Côtes-d'Armor', '42'), ('23', 'Allier', '43'), ('24', 'Alpes de Haute-Provence', '44'), ('25', 'Hautes-Alpes', '45'), ('26', 'Alpes-Maritimes', '46'), ('27', 'Ardennes', '47'), ('28', 'Ardennes', '48'), ('29', 'Ariège', '49'), ('30', 'Aube', '50'), ('31', 'Aube', '51'), ('32', 'Aveyron', '52'), ('33', 'Bouches-du-Rhône', '53'), ('34', 'Calvados', '54'), ('35', 'Cantal', '55'), ('36', 'Charente-Maritime', '56'), ('37', 'Charente-Maritime', '57'), ('38', 'Cher', '58'), ('39', 'Corrèze', '59'), ('40', 'Corse-du-Sud', '60'), ('41', 'Côte-d'Or', '61'), ('42', 'Côtes-d'Armor', '62'), ('43', 'Allier', '63'), ('44', 'Alpes de Haute-Provence', '64'), ('45', 'Hautes-Alpes', '65'), ('46', 'Alpes-Maritimes', '66'), ('47', 'Ardennes', '67'), ('48', 'Ardennes', '68'), ('49', 'Ariège', '69'), ('50', 'Aube', '70'), ('51', 'Aube', '71'), ('52', 'Aveyron', '72'), ('53', 'Bouches-du-Rhône', '73'), ('54', 'Calvados', '74'), ('55', 'Cantal', '75'), ('56', 'Charente-Maritime', '76'), ('57', 'Charente-Maritime', '77'), ('58', 'Cher', '78'), ('59', 'Corrèze', '79'), ('60', 'Corse-du-Sud', '80'), ('61', 'Côte-d'Or', '81'), ('62', 'Côtes-d'Armor', '82'), ('63', 'Allier', '83'), ('64', 'Alpes de Haute-Provence', '84'), ('65', 'Hautes-Alpes', '85'), ('66', 'Alpes-Maritimes', '86'), ('67', 'Ardennes', '87'), ('68', 'Ardennes', '88'), ('69', 'Ariège', '89'), ('70', 'Aube', '90'), ('71', 'Aube', '91'), ('72', 'Aveyron', '92'), ('73', 'Bouches-du-Rhône', '93'), ('74', 'Calvados', '94'), ('75', 'Cantal', '95'), ('76', 'Charente-Maritime', '96'), ('77', 'Charente-Maritime', '97'), ('78', 'Cher', '98'), ('79', 'Corrèze', '99'), ('80', 'Corse-du-Sud', '100'), ('81', 'Côte-d'Or', '101'), ('82', 'Côtes-d'Armor', '102'), ('83', 'Allier', '103'), ('84', 'Alpes de Haute-Provence', '104'), ('85', 'Hautes-Alpes', '105'), ('86', 'Alpes-Maritimes', '106'), ('87', 'Ardennes', '107'), ('88', 'Ardennes', '108'), ('89', 'Ariège', '109'), ('90', 'Aube', '110'), ('91', 'Aube', '111'), ('92', 'Aveyron', '112'), ('93', 'Bouches-du-Rhône', '113'), ('94', 'Calvados', '114'), ('95', 'Cantal', '115'), ('96', 'Charente-Maritime', '116'), ('97', 'Charente-Maritime', '117'), ('98', 'Cher', '118'), ('99', 'Corrèze', '119'), ('100', 'Corse-du-Sud', '120'), ('101', 'Côte-d'Or', '121'), ('102', 'Côtes-d'Armor', '122'), ('103', 'Allier', '123'), ('104', 'Alpes de Haute-Provence', '124'), ('105', 'Hautes-Alpes', '125'), ('106', 'Alpes-Maritimes', '126'), ('107', 'Ardennes', '127'), ('108', 'Ardennes', '128'), ('109', 'Ariège', '129'), ('110', 'Aube', '130'), ('111', 'Aube', '131'), ('112', 'Aveyron', '132'), ('113', 'Bouches-du-Rhône', '133'), ('114', 'Calvados', '134'), ('115', 'Cantal', '135'), ('116', 'Charente-Maritime', '136'), ('117', 'Charente-Maritime', '137'), ('118', 'Cher', '138'), ('119', 'Corrèze', '139'), ('120', 'Corse-du-Sud', '140'), ('121', 'Côte-d'Or', '141'), ('122', 'Côtes-d'Armor', '142'), ('123', 'Allier', '143'), ('124', 'Alpes de Haute-Provence', '144'), ('125', 'Hautes-Alpes', '145'), ('126', 'Alpes-Maritimes', '146'), ('127', 'Ardennes', '147'), ('128', 'Ardennes', '148'), ('129', 'Ariège', '149'), ('130', 'Aube', '150'), ('131', 'Aube', '151'), ('132', 'Aveyron', '152'), ('133', 'Bouches-du-Rhône', '153'), ('134', 'Calvados', '154'), ('135', 'Cantal', '155'), ('136', 'Charente-Maritime', '156'), ('137', 'Charente-Maritime', '157'), ('138', 'Cher', '158'), ('139', 'Corrèze', '159'), ('140', 'Corse-du-Sud', '160'), ('141', 'Côte-d'Or', '161'), ('142', 'Côtes-d'Armor', '162'), ('143', 'Allier', '163'), ('144', 'Alpes de Haute-Provence', '164'), ('145', 'Hautes-Alpes', '165'), ('146', 'Alpes-Maritimes', '166'), ('147', 'Ardennes', '167'), ('148', 'Ardennes', '168'), ('149', 'Ariège', '169'), ('150', 'Aube', '170'), ('151', 'Aube', '171'), ('152', 'Aveyron', '172'), ('153', 'Bouches-du-Rhône', '173'), ('154', 'Calvados', '174'), ('155', 'Cantal', '175'), ('156', 'Charente-Maritime', '176'), ('157', 'Charente-Maritime', '177'), ('158', 'Cher', '178'), ('159', 'Corrèze', '179'), ('160', 'Corse-du-Sud', '180'), ('161', 'Côte-d'Or', '181'), ('162', 'Côtes-d'Armor', '182'), ('163', 'Allier', '183'), ('164', 'Alpes de Haute-Provence', '184'), ('165', 'Hautes-Alpes', '185'), ('166', 'Alpes-Maritimes', '186'), ('167', 'Ardennes', '187'), ('168', 'Ardennes', '188'), ('169', 'Ariège', '189'), ('170', 'Aube', '190'), ('171', 'Aube', '191'), ('172', 'Aveyron', '192'), ('173', 'Bouches-du-Rhône', '193'), ('174', 'Calvados', '194'), ('175', 'Cantal', '195'), ('176', 'Charente-Maritime', '196'), ('177', 'Charente-Maritime', '197'), ('178', 'Cher', '198'), ('179', 'Corrèze', '199'), ('180', 'Corse-du-Sud', '200'), ('181', 'Côte-d'Or', '201'), ('182', 'Côtes-d'Armor', '202'), ('183', 'Allier', '203'), ('184', 'Alpes de Haute-Provence', '204'), ('185', 'Hautes-Alpes', '205'), ('186', 'Alpes-Maritimes', '206'), ('187', 'Ardennes', '207'), ('188', 'Ardennes', '208'), ('189', 'Ariège', '209'), ('190', 'Aube', '210'), ('191', 'Aube', '211'), ('192', 'Aveyron', '212'), ('193', 'Bouches-du-Rhône', '213'), ('194', 'Calvados', '214'), ('195', 'Cantal', '215'), ('196', 'Charente-Maritime', '216'), ('197', 'Charente-Maritime', '217'), ('198', 'Cher', '218'), ('199', 'Corrèze', '219'), ('200', 'Corse-du-Sud', '220'), ('201', 'Côte-d'Or', '221'), ('202', 'Côtes-d'Armor', '222'), ('203', 'Allier', '223'), ('204', 'Alpes de Haute-Provence', '224'), ('205', 'Hautes-Alpes', '225'), ('206', 'Alpes-Maritimes', '226'), ('207', 'Ardennes', '227'), ('208', 'Ardennes', '228'), ('209', 'Ariège', '229'), ('210', 'Aube', '230'), ('211', 'Aube', '231'), ('212', 'Aveyron', '232'), ('213', 'Bouches-du-Rhône', '233'), ('214', 'Calvados', '234'), ('215', 'Cantal', '235'), ('216', 'Charente-Maritime', '236'), ('217', 'Charente-Maritime', '237'), ('218', 'Cher', '238'), ('219', 'Corrèze', '239'), ('220', 'Corse-du-Sud', '240'), ('221', 'Côte-d'Or', '241'), ('222', 'Côtes-d'Armor', '242'), ('223', 'Allier', '243'), ('224', 'Alpes de Haute-Provence', '244'), ('225', 'Hautes-Alpes', '245'), ('226', 'Alpes-Maritimes', '246'), ('227', 'Ardennes', '247'), ('228', 'Ardennes', '248'), ('229', 'Ariège', '249'), ('230', 'Aube', '250'), ('231', 'Aube', '251'), ('232', 'Aveyron', '252'), ('233', 'Bouches-du-Rhône', '253'), ('234', 'Calvados', '254'), ('235', 'Cantal', '255'), ('236', 'Charente-Maritime', '256'), ('237', 'Charente-Maritime', '257'), ('238', 'Cher', '258'), ('239', 'Corrèze', '259'), ('240', 'Corse-du-Sud', '260'), ('241', 'Côte-d'Or', '261'), ('242', 'Côtes-d'Armor', '262'), ('243', 'Allier', '263'), ('244', 'Alpes de Haute-Provence', '264'), ('245', 'Hautes-Alpes', '265'), ('246', 'Alpes-Maritimes', '266'), ('247', 'Ardennes', '267'), ('248', 'Ardennes', '268'), ('249', 'Ariège', '269'), ('250', 'Aube', '270'), ('251', 'Aube', '271'), ('252', 'Aveyron', '272'), ('253', 'Bouches-du-Rhône', '273'), ('254', 'Calvados', '274'), ('255', 'Cantal', '275'), ('256', 'Charente-Maritime', '276'), ('257', 'Charente-Maritime', '277'), ('258', 'Cher', '278'), ('259', 'Corrèze', '279'), ('260', 'Corse-du-Sud', '280'), ('261', 'Côte-d'Or', '281'), ('262', 'Côtes-d'Armor', '282'), ('263', 'Allier', '283'), ('264', 'Alpes de Haute-Provence', '284'), ('265', 'Hautes-Alpes', '285'), ('266', 'Alpes-Maritimes', '286'), ('267', 'Ardennes', '287'), ('268', 'Ardennes', '288'), ('269', 'Ariège', '289'), ('270', 'Aube', '290'), ('271', 'Aube', '291'), ('272', 'Aveyron', '292'), ('273', 'Bouches

```
localflavor.fr.fr_department.DEPARTMENT_CHOICES = (('01', '01 - Ain'), ('02', '02 - Aisne'), ('03', '03 - Allier'), ('04', '04 - Ardennes'), ('05', '05 - Aube'), ('06', '06 - Alpes-Maritimes'), ('07', '07 - Calvados'), ('08', '08 - Nord'), ('09', '09 - Haute-Garonne'), ('10', '10 - Côte-d\'Or'), ('11', '11 - Charente-Maritime'), ('12', '12 - Gers'), ('13', '13 - Bouches-du-Rhône'), ('14', '14 - Mayenne'), ('15', '15 - Corrèze'), ('16', '16 - Charente'), ('17', '17 - Dordogne'), ('18', '18 - Creuse'), ('19', '19 - Haute-Loire'), ('20', '20 - Corse-du-Sud'), ('21', '21 - Côte-d\'Or'), ('22', '22 - Finistère'), ('23', '23 - Allier'), ('24', '24 - Gironde'), ('25', '25 - Doubs'), ('26', '26 - Drôme'), ('27', '27 - Eure'), ('28', '28 - Eure-et-Loir'), ('29', '29 - Finistère'), ('30', '30 - Gard'), ('31', '31 - Haute-Garonne'), ('32', '32 - Ariège'), ('33', '33 - Gironde'), ('34', '34 - Hérault'), ('35', '35 - Ille-et-Vilaine'), ('36', '36 - Indre-et-Loire'), ('37', '37 - Maine-et-Loire'), ('38', '38 - Savoie'), ('39', '39 - Haute-Savoie'), ('40', '40 - Landes'), ('41', '41 - Loir-et-Cher'), ('42', '42 - Loire'), ('43', '43 - Haute-Loire'), ('44', '44 - Loire-Atlantique'), ('45', '45 - Loiret'), ('46', '46 - Lot'), ('47', '47 - Lot-et-Garonne'), ('48', '48 - Lozère'), ('49', '49 - Mayenne'), ('50', '50 - Manche'), ('51', '51 - Marne'), ('52', '52 - Haute-Marne'), ('53', '53 - Mayenne'), ('54', '54 - Meurthe-et-Moselle'), ('55', '55 - Meuse'), ('56', '56 - Morbihan'), ('57', '57 - Moselle'), ('58', '58 - Cher'), ('59', '59 - Nord'), ('60', '60 - Oise'), ('61', '61 - Sarthe'), ('62', '62 - Pas-de-Calais'), ('63', '63 - Puy-de-Dôme'), ('64', '64 - Pyrénées-Atlantiques'), ('65', '65 - Hautes-Pyrénées'), ('66', '66 - Pyrénées-Orientales'), ('67', '67 - Bas-Rhin'), ('68', '68 - Haut-Rhin'), ('69', '69 - Rhône'), ('70', '70 - Vosges'), ('71', '71 - Saône-et-Loire'), ('72', '72 - Mayenne'), ('73', '73 - Savoie'), ('74', '74 - Haute-Savoie'), ('75', '75 - Paris'), ('76', '76 - Seine-Maritime'), ('77', '77 - Marne-la-Vallée'), ('78', '78 - Yvelines'), ('79', '79 - Deux-Sèvres'), ('80', '80 - Somme'), ('81', '81 - Tarn'), ('82', '82 - Tarn-et-Garonne'), ('83', '83 - Var'), ('84', '84 - Vaucluse'), ('85', '85 - Loire-Atlantique'), ('86', '86 - Vienne'), ('87', '87 - Dordogne'), ('88', '88 - Vosges'), ('89', '89 - Aisne'), ('90', '90 - Territoire de Belfort'), ('91', '91 - Essonne'), ('92', '92 - Hauts-de-Seine'), ('93', '93 - Seine-Saint-Denis'), ('94', '94 - Val-de-Marne'), ('95', '95 - Val-d\'Oise')
A list of departments
```

localflavor.fr.fr_region.**REGION_CHOICES** = (('01', 'Guadeloupe'), ('02', 'Martinique'), ('03', 'Guyane'), ('04', 'La
See the “Code officiel géographique” on the INSEE website <www.insee.fr>.

7.17 Great Britain (gb)

7.17.1 Forms

GB-specific Form helpers

```
class localflavor.gb.forms.GBCountySelect (attrs=None)
    A Select widget that uses a list of UK Counties/Regions as its choices.
```

```
class localflavor.gb.forms.GBNationSelect (attrs=None)
    A Select widget that uses a list of UK Nations as its choices.
```

```
class localflavor.gb.forms.GBPpostcodeField(max_length=None, min_length=None, *args,
                                             **kwargs)
    A form field that validates its input is a UK postcode.
```

The regular expression used is sourced from the schema for British Standard BS7666 address types: <http://www.govtalk.gov.uk/gdsc/schemas/bs7666-v2-0.xsd>

The value is uppercased and a space added in the correct place, if required.

7.17.2 Data

localflavor.gb.gb_regions.ENGLAND_REGION_CHOICES = (('Bedfordshire', 'Bedfordshire'), ('Berkshire', 'Berkshire'))

English regions http://en.wikipedia.org/wiki/List_of_ceremonial_counties_of_England

```
localflavor.gb.gb_regions.NORTHERN_IRELAND_REGION_CHOICES = (('County Antrim', 'County Antrim'), ('County Londonderry', 'County Londonderry'), ('County Tyrone', 'County Tyrone'))
Northern Ireland regions: http://en.wikipedia.org/wiki/List\_of\_Irish\_counties\_by\_area
```

```
localflavor.gb.gb_regions.WALES_REGION_CHOICES = (('Clwyd', 'Clwyd'), ('Dyfed', 'Dyfed'), ('Gwent', 'Gwent'),
    Welsh regions: http://en.wikipedia.org/wiki/Preserved\_counties\_of\_Wales
```

```
localflavor.gb.gb_regions.SCOTTISH_REGION_CHOICES = (('Borders', 'Borders'), ('Central Scotland', 'Central Scotland'))
```

Scottish regions: http://en.wikipedia.org/wiki/Regions_and_districts_of_Scotland


```
class localflavor.hr.forms.HRLicensePlateField(required=True, widget=None,
                                              label=None, initial=None,
                                              help_text=None, error_messages=None,
                                              show_hidden_initial=False, validators=[]),
                                              localize=False)
```

Vehicle license plate of Croatia field. Normalizes to the specific format below. Suffix is constructed from the shared letters of the Croatian and English alphabets.

Format examples: SB 123-A (but also supports more characters) ZG 1234-AA

Used for standardized license plates only.

```
class localflavor.hr.forms.HRLicensePlatePrefixSelect(attrs=None)
    A Select widget that uses a list of vehicle license plate prefixes of Croatia as its choices.
```

```
class localflavor.hr.forms.HROIBField(min_length=11, max_length=11, *args, **kwargs)
    Personal Identification Number of Croatia (OIB) field.
```

<http://www.oib.hr/>

```
class localflavor.hr.forms.HRPhoneNumberField(required=True, widget=None,
                                              label=None, initial=None,
                                              help_text=None, error_messages=None,
                                              show_hidden_initial=False, validators=[]),
                                              localize=False)
```

Phone number of Croatia field. Format: Complete country code or leading zero, area code prefix, 6 or 7 digit number. Validates fixed, mobile and FGSM numbers. Normalizes to a full number with country code (+385 prefix).

```
class localflavor.hr.forms.HRPhoneNumberPrefixSelect(attrs=None)
    A Select widget that uses a list of phone number prefixes of Croatia as its choices.
```

```
class localflavor.hr.forms.HRPostalCodeField(required=True, widget=None,
                                              label=None, initial=None,
                                              help_text=None, error_messages=None,
                                              show_hidden_initial=False, validators=[]),
                                              localize=False)
```

Postal code of Croatia field. It consists of exactly five digits ranging from 10000 to possibly less than 60000.

<http://www.posta.hr/main.aspx?id=66>

7.19.2 Data

```
localflavor.hr.hr_choices.HR_COUNTY_CHOICES = (('GZG', 'Grad Zagreb'), ('BBŽ', 'Bjelovarsko-bilogorska župa'),
        Croatian Counties: http://en.wikipedia.org/wiki/ISO\_3166-2:HR Croatia doesn't have official abbreviations for counties. The ones provided are in common use.
```

```
localflavor.hr.hr_choices.HR_LICENSE_PLATE_PREFIX_CHOICES = (('BJ', 'BJ'), ('BM', 'BM'), ('ČK', 'ČK'), ('I'),
        Only common license plate prefixes are provided. Special cases and obsolete prefixes are omitted.
        http://hr.wikipedia.org/wiki/Dodatak:Popis\_registracijskih\_oznaka\_za\_cestovna\_vozila\_u\_Hrvatskoj
```

```
localflavor.hr.hr_choices.HR_PHONE_NUMBER_PREFIX_CHOICES = (('1', '01'), ('20', '020'), ('21', '021'), ('22', '02'),
        The list includes county and cellular network phone number prefixes.
```


7.20 Indonesia (id)

7.20.1 Forms

ID-specific Form helpers

```
class localflavor.id_.forms.IDLicensePlateField(required=True,                widget=None,
                                                label=None,                initial=None,
                                                help_text=None,    error_messages=None,
                                                show_hidden_initial=False, validators=[
                                                    ], localize=False)
```

An Indonesian vehicle license plate field.

http://id.wikipedia.org/wiki/Tanda_Nomor_Kendaraan_Bermotor

Plus: "B 12345 12"

```
class localflavor.id_.forms.IDLicensePlatePrefixSelect(attrs=None)
    A Select widget that uses a list of vehicle license plate prefix code of Indonesia as its choices.
```

http://id.wikipedia.org/wiki/Tanda_Nomor_Kendaraan_Bermotor

```
class localflavor.id_.forms.IDNationalIdentityNumberField(required=True,
                                                            widget=None,        la-
                                                            bel=None,    initial=None,
                                                            help_text=None,    er-
                                                            ror_messages=None,
                                                            show_hidden_initial=False,
                                                            validators=[        ], localize=False)
```

An Indonesian national identity number (NIK/KTP#) field.

http://id.wikipedia.org/wiki/Nomor_Induk_Kependudukan

xx.xxxx.ddmmyy.xxxx - 16 digits (excl. dots)

```
class localflavor.id_.forms.IDPhoneNumberField(required=True,                widget=None,
                                                label=None,                initial=None,
                                                help_text=None,    error_messages=None,
                                                show_hidden_initial=False, validators=[
                                                    ], localize=False)
```

An Indonesian telephone number field.

http://id.wikipedia.org/wiki/Daftar_kode_telepon_di_Indonesia

```
class localflavor.id_.forms.IDPostCodeField(required=True,                widget=None,
                                              label=None,                initial=None,
                                              help_text=None,    error_messages=None,
                                              show_hidden_initial=False, validators=[
                                                  ], localize=False)
```

An Indonesian post code field.

http://id.wikipedia.org/wiki/Kode_pos

```
class localflavor.id_.forms.IDProvinceSelect(attrs=None)
    A Select widget that uses a list of provinces of Indonesia as its choices.
```

```
localflavor.id.id_choices.LICENSE_PLATE_PREFIX_CHOICES = (('A', 'Banten'), ('AA', 'Magelang'), ('AB', 'Yogyakarta'))
```

7.21 Ireland (ie)

7.21.1 Forms

class localflavor.ie.forms.**IECountySelect** (*attrs=None*)
A Select widget that uses a list of Irish Counties as its choices.

7.21.2 Data

```
localflavor.ie.ie_counties.IE_COUNTY_CHOICES = (('carlow', 'Carlow'), ('cavan', 'Cavan'), ('clare', 'Clare'), ('co
```

Irish Counties: http://en.wikipedia.org/wiki/Counties_of_Ireland

7.22 Israel (i1)

7.22.1 Forms

http://he.wikipedia.org/wiki/%D7%9E%D7%A1%D7%A4%D7%A8_%D7%96%D7%94%D7%95%D7%AA_(%D7%99%D7%
http://he.wikipedia.org/wiki/%D7%A1%D7%A4%D7%A8%D7%AA_%D7%91%D7%99%D7%A7%D7%95%D7%A8%D7%A
http://he.wikipedia.org/wiki/%D7%A7%D7%99%D7%93%D7%95%D7%9E%D7%AA_%D7%98%D7%9C%D7%A4%D7%95

A form field that validates its input as an Israeli postal code. Valid form is XXXXX where X represents integer.

7.23 India (in)

7.23.1 Forms

India-specific Form helpers.

```
class localflavor.in_.forms.INPhoneNumberField(max_length=None,    min_length=None,
                                                *args, **kwargs)
```

INPhoneNumberField validates that the data is a valid Indian phone number, including the STD code. It's normalised to 0XXX-XXXXXXX or 0XXX XXXXXXXX format. The first string is the STD code which is a '0' followed by 2-4 digits. The second string is 8 digits if the STD code is 3 digits, 7 digits if the STD code is 4 digits and 6 digits if the STD code is 5 digits. The second string will start with numbers between 1 and 6. The separator is either a space or a hyphen.

```
class localflavor.in_.forms.INStateField(required=True,  widget=None,  label=None,  ini-
                                          tial=None,  help_text=None,  error_messages=None,
                                          show_hidden_initial=False,  validators=[],  local-
                                          ize=False)
```

A form field that validates its input is a Indian state name or abbreviation. It normalizes the input to the standard two-letter vehicle registration abbreviation for the given state or union territory

```
class localflavor.in_.forms.INStateSelect(attrs=None)
```

A Select widget that uses a list of Indian states/territories as its choices.

```
class localflavor.in_.forms.INZipCodeField(max_length=None,  min_length=None,  *args,
                                           **kwargs)
```

A form field that validates input as an Indian zip code, with the format XXXXXXXX.

7.23.2 Data

```
localflavor.in_.in_states.STATE_CHOICES = (('KA', 'Karnataka'), ('AP', 'Andhra Pradesh'), ('KL', 'Kerala'), ('TN', 'Tamil Nadu'))
```

A list of states

```
localflavor.in_.in_states.STATES_NORMALIZED = {'uttarakhand': 'UA', 'ch': 'CH', 'karnataka': 'KA', 'tamilnad': 'TN'}
```

Normalized state names

7.24 Iceland (is)

7.24.1 Forms

Iceland specific form helpers.

```
class localflavor.is_.forms.ISIdNumberField(max_length=11,    min_length=10,    *args,
                                              **kwargs)
```

Icelandic identification number (kennitala). This is a number every citizen of Iceland has.

```
class localflavor.is_.forms.ISPhoneNumberField(max_length=8,    min_length=7,    *args,
                                              **kwargs)
```

Icelandic phone number. Seven digits with an optional hyphen or space after the first three digits.

```
class localflavor.is_.forms.ISPostalCodeSelect(attrs=None)
```

A Select widget that uses a list of Icelandic postal codes as its choices.

7.25 Italy (it)

7.25.3 Data

```
localflavor.it.it_region.REGION_CHOICES = (('ABR', 'Abruzzo'), ('BAS', 'Basilicata'), ('CAL', 'Calabria'), ('CAM
```

An alphabetical list of regions

7.26 Japan (jp)

7.26.1 Forms

JP-specific Form helpers

```
class localflavor.jp.forms.JPPostalCodeField(max_length=None, min_length=None, *args,  
                                              **kwargs)
```

A form field that validates its input is a Japanese postcode.

Accepts 7 digits, with or without a hyphen.

```
clean(value)
```

Validates the input and returns a string that contains only numbers. Returns an empty string for empty values.

```
class localflavor.jp.forms.JPPrefectureSelect(attrs=None)
```

A Select widget that uses a list of Japanese prefectures as its choices.

7.26.2 Data

```
localflavor.jp.jp_prefectures.JP_PREFECTURES = (('hokkaido', 'Hokkaido'), ('aomori', 'Aomori'), ('iwate', 'Iwa
```

A list of prefectures

7.27 Kuwait (kw)

7.27.1 Forms

Kuwait-specific Form helpers

```
class localflavor.kw.forms.KWCivilIDNumberField(required=True, widget=None,  
                                                  label=None, initial=None,  
                                                  help_text=None, error_messages=None,  
                                                  show_hidden_initial=False, validators=[  
                                                  ], localize=False)
```

Kuwaiti Civil ID numbers are 12 digits, second to seventh digits represents the person's birthdate.

Checks the following rules to determine the validity of the number:

- The number consist of 12 digits.
- The birthdate of the person is a valid date.
- The calculated checksum equals to the last digit of the Civil ID.

7.28 Lithuania (lt)

7.28.1 Forms

```
class localflavor.lt.forms.LTCountySelect(attrs=None)
```

A select field with the Lithuanian counties as choices

```
class localflavor.lt.forms.LTIDCodeField(*args, **kwargs)
```

A form field that validates as Lithuanian ID Code.

Checks:

- Made of exactly 11 decimal numbers.
- Checksum is correct.
- ID contains valid date.

valid_date (*value*)

Check if date in ID code is valid. We won't check for dates in future as it would become too restrictive.

class localflavor.lt.forms.**LTMunicipalitySelect** (*attrs=None*)

A select field with the Lithuanian municipalities as choices

class localflavor.lt.forms.**LTPostalCodeField** (**args, **kwargs*)

A form field that validates as Lithuanian postal code

Lithuanian postal codes in following forms accepted:

- XXXXX
- LT-XXXXX

7.28.2 Data

localflavor.lt.lt_choices.**COUNTY_CHOICES** = (('alytus', 'Alytus'), ('kaunas', 'Kaunas'), ('klaipeda', 'Klaipda'), ('m'

Alphabetically sorted lists of Lithuanian counties.

localflavor.lt.lt_choices.**MUNICIPALITY_CHOICES** = (('akmene', 'Akmen district'), ('alytus_c', 'Alytus city'), ('al'

Alphabetically sorted lists of Lithuanian municipalities.

7.29 Macedonia (mk)

7.29.1 Forms

class localflavor.mk.forms.**MKIdentityCardNumberField** (**args, **kwargs*)

A Macedonian ID card number. Accepts both old and new format.

class localflavor.mk.forms.**MKMunicipalitySelect** (*attrs=None*)

A form Select widget that uses a list of Macedonian municipalities as choices. The label is the name of the municipality and the value is a 2 character code for the municipality.

class localflavor.mk.forms.**UMCNField** (**args, **kwargs*)

A form field that validates input as a unique master citizen number.

The format of the unique master citizen number has been kept the same from Yugoslavia. It is still in use in other countries as well, it is not applicable solely in Macedonia. For more information see: https://secure.wikimedia.org/wikipedia/en/wiki/Unique_Master_Citizen_Number

A value will pass validation if it complies to the following rules:

- Consists of exactly 13 digits
- The first 7 digits represent a valid past date in the format DDMMYYYY
- The last digit of the UMCN passes a checksum test

7.29.2 Models

class `localflavor.mk.models.MKIdentityCardNumberField(*args, **kwargs)`

A form field that validates input as a Macedonian identity card number. Both old and new identity card numbers are supported.

class `localflavor.mk.models.MKMunicipalityField(*args, **kwargs)`

A form field that validates input as a Macedonian identity card number. Both old and new identity card numbers are supported.

class `localflavor.mk.models.UMCNField(*args, **kwargs)`

A form field that validates input as a unique master citizen number.

The format of the unique master citizen number is not unique to Macedonia. For more information see: https://secure.wikimedia.org/wikipedia/en/wiki/Unique_Master_Citizen_Number

A value will pass validation if it complies to the following rules:

- Consists of exactly 13 digits
- The first 7 digits represent a valid past date in the format DDMMYYYY
- The last digit of the UMCN passes a checksum test

7.29.3 Data

`localflavor.mk.mk_choices.MK_MUNICIPALITIES = (('AD', 'Aerodrom'), ('AR', 'Aračinovo'), ('BR', 'Berovo'), ('TL', 'Tetovo'))`
Macedonian municipalities per the reorganization from 2004.

7.30 Mexico (mx)

7.30.1 Forms

Mexican-specific form helpers.

`localflavor.mx.forms.CURP_INCONVENIENT_WORDS = ('BACA', 'BAKA', 'BUEI', 'BUEY', 'CACA', 'CACO', 'CA'`

This is the list of inconvenient words according to the *Anexo 2* of the document described in the next link: <http://portal.veracruz.gob.mx/pls/portal/url/ITEM/444112558A57C6E0E040A8C02E00695C>

class `localflavor.mx.forms.MXCURPField(min_length=18, max_length=18, *args, **kwargs)`

A field that validates a Mexican Clave Única de Registro de Población.

The CURP is integrated by a juxtaposition of characters following the next pattern:

Index	Format	Accepted Characters
1	X	Any letter
2	X	Any vowel
3-4	XX	Any letter
5-10	YYMMDD	Any valid date
11	X	Either <i>H</i> or <i>M</i> , depending on the person's gender
12-13	XX	Any valid acronym for a state in Mexico
14-16	XXX	Any consonant
17	X	Any number between 0 and 9 or any letter
18	X	Any number between 0 and 9

More info about this: <http://www.condusef.gob.mx/index.php/clave-unica-de-registro-de-poblacion-curp>

class `localflavor.mx.forms.MXRFCField` (*min_length=9, max_length=13, *args, **kwargs*)
 A form field that validates a Mexican *Registro Federal de Contribuyentes* for either *Persona física* or *Persona moral*.

The *Persona física* RFC string is integrated by a juxtaposition of characters following the next pattern:

Index	Format	Accepted Characters
1	X	Any letter
2	X	Any vowel
3-4	XX	Any letter
5-10	YYMMDD	Any valid date
11-12	XX	Any letter or number between 0 and 9
13	X	Any digit between 0 and 9 or the letter A

The *Persona moral* RFC string is integrated by a juxtaposition of characters following the next pattern:

Index	Format	Accepted Characters
1-3	XXX	Any letter including & and Ñ chars
4-9	YYMMDD	Any valid date
10-11	XX	Any letter or number between 0 and 9
12	X	Any number between 0 and 9 or the letter A

More info about this: [http://es.wikipedia.org/wiki/Registro_Federal_de_Contribuyentes_\(M%C3%A9xico\)](http://es.wikipedia.org/wiki/Registro_Federal_de_Contribuyentes_(M%C3%A9xico))

class `localflavor.mx.forms.MXSocialSecurityNumberField` (*min_length=11, max_length=11, *args, **kwargs*)

A field that validates a Mexican Social Security Number.

The Social Security Number is integrated by a juxtaposition of digits following the next pattern:

In- dex	Required numbers
1-2	The number of the branch office where the Social Security Number was designated.
3-4	The year of inscription to the Social Security.
5-6	The year of birth of the Social Security Number owner.
7-10	The progressive number of procedure for the IMSS. (This digit is provided exclusively by the Institute as it regards the Folio number of such procedure).
11	The verification digit.

class `localflavor.mx.forms.MXStateSelect` (*attrs=None*)

A Select widget that uses a list of Mexican states as its choices.

class `localflavor.mx.forms.MXZipCodeField` (**args, **kwargs*)

A form field that accepts a Mexican Zip Code.

More info about this: http://en.wikipedia.org/wiki/List_of_postal_codes_in_Mexico

`localflavor.mx.forms.RFC_INCONVENIENT_WORDS = ('BUEI','BUEY','CACA','CACO','CAGA','CAGO','CAI`

This is the list of inconvenient words according to the *Anexo IV* of the document described in the next link:

http://www.sisi.org.mx/jspsi/documentos/2005/seguimiento/06101/0610100162005_065.doc

7.30.2 Models

class `localflavor.mx.models.MXCURPField` (**args, **kwargs*)

A model field that forms represent as a `forms.MXCURPField` field and stores the value of a valid Mexican CURP.


```
class localflavor.mx.models.MXRFCField(*args, **kwargs)
    A model field that forms represent as a forms.MXRFCField field and stores the value of a valid Mexican RFC.

class localflavor.mx.models.MXSocialSecurityNumberField(*args, **kwargs)
    A model field that forms represent as a forms.MXSocialSecurityNumberField field and stores the value of a
    valid Mexican Social Security Number.

class localflavor.mx.models.MXStateField(*args, **kwargs)
    A model field that stores the three-letter Mexican state abbreviation in the database.

class localflavor.mx.models.MXZipCodeField(*args, **kwargs)
    A model field that forms represent as a forms.MXZipCodeField field and stores the five-digit Mexican zip code.
```

7.30.3 Data

```
localflavor.mx.mx_states.STATE_CHOICES = (('AGU', 'Aguascalientes'), ('BCN', 'Baja California'), ('BCS', 'Baja C
    All 31 states, plus the Distrito Federal.
```

7.31 The Netherlands (nl)

7.31.1 Forms

NL-specific Form helpers

```
class localflavor.nl.forms.NLPhoneNumberField(required=True, widget=None,
                                              label=None, initial=None,
                                              help_text=None, error_messages=None,
                                              show_hidden_initial=False, validators=[],
                                              localize=False)

    A Dutch telephone number field.
```

```
class localflavor.nl.forms.NLProvinceSelect(attrs=None)
    A Select widget that uses a list of provinces of the Netherlands as its choices.
```

```
class localflavor.nl.forms.NLSoFiNumberField(required=True, widget=None,
                                              label=None, initial=None,
                                              help_text=None, error_messages=None,
                                              show_hidden_initial=False, validators=[],
                                              localize=False)

    A Dutch social security number (SoFi/BSN) field.
```

<http://nl.wikipedia.org/wiki/Sofinummer>

```
class localflavor.nl.forms.NLZipCodeField(required=True, widget=None, label=None,
                                           initial=None, help_text=None, er-
                                           ror_messages=None, show_hidden_initial=False,
                                           validators=[], localize=False)

    A Dutch postal code field.
```

7.31.2 Data

```
localflavor.nl.nl_provinces.PROVINCE_CHOICES = (('DR', 'Drenthe'), ('FL', 'Flevoland'), ('FR', 'Fryslân'), ('GL',
    An alphabetical list of provinces
```

7.32 Norway (no)

7.32.1 Forms

Norwegian-specific Form helpers

class localflavor.no.forms.**NOMunicipalitySelect** (*attrs=None*)

A Select widget that uses a list of Norwegian municipalities (fylker) as its choices.

class localflavor.no.forms.**NOPhoneNumberField** (*max_length=None, min_length=None, *args, **kwargs*)

Field with phonenummer validation. Requires a phone number with 8 digits and optional country code

class localflavor.no.forms.**NOSocialSecurityNumber** (*required=True, get=None, label=None, initial=None, help_text=None, error_messages=None, show_hidden_initial=False, validators=[], localize=False*)

Algorithm is documented at <http://no.wikipedia.org/wiki/Personnummer>

class localflavor.no.forms.**NOZipCodeField** (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates input as a Norwegian zip code. Valid codes have four digits.

7.32.2 Data

localflavor.no.no_municipalities.**MUNICIPALITY_CHOICES** = (('akershus', 'Akershus'), ('austagder', 'Aust-Agder'), ...)

An alphabetical list of Norwegian municipalities (fylker) for use as *choices* in a formfield.

7.33 Peru (pe)

7.33.1 Forms

PE-specific Form helpers.

class localflavor.pe.forms.**PEDNIField** (*max_length=8, min_length=8, *args, **kwargs*)

A field that validates Documento Nacional de Identidad (DNI) numbers.

clean (*value*)

Value must be a string in the XXXXXXXX formats.

class localflavor.pe.forms.**PERUCField** (*max_length=11, min_length=11, *args, **kwargs*)

This field validates a RUC (Registro Unico de Contribuyentes). A RUC is of the form XXXXXXXXXXXXX.

clean (*value*)

Value must be an 11-digit number.

class localflavor.pe.forms.**PERRegionSelect** (*attrs=None*)

A Select widget that uses a list of Peruvian Regions as its choices.

7.33.2 Data

`localflavor.pe.pe_region.REGION_CHOICES = (('AMA', 'Amazonas'), ('ANC', 'Ancash'), ('APU', 'Apurímac'), ('AR`
 A list of Peru regions as *choices* in a formfield.

7.34 Poland (p1)

7.34.1 Forms

Polish-specific form helpers

class `localflavor.pl.forms.PLCountySelect` (*attrs=None*)

A select widget with list of Polish administrative units as choices.

class `localflavor.pl.forms.PLNIPField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates as Polish Tax Number (NIP). Valid forms are: XXX-YYY-YY-YY, XXX-YY-YY-YYY or XXXYYYYYYYY.

Checksum algorithm based on documentation at <http://wipos.p.lodz.pl/zylla/ut/nip-rego.html>

has_valid_checksum (*number*)

Calculates a checksum with the provided algorithm.

class `localflavor.pl.forms.PLNationalIDCardNumberField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates as Polish National ID Card Number.

Checks the following rules:

- the length consist of 3 letter and 6 digits
- has a valid checksum

The algorithm is documented at http://en.wikipedia.org/wiki/Polish_identity_card.

has_valid_checksum (*number*)

Calculates a checksum with the provided algorithm.

class `localflavor.pl.forms.PLPESELField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates as Polish Identification Number (PESEL).

Checks the following rules:

- the length consist of 11 digits
- has a valid checksum

The algorithm is documented at <http://en.wikipedia.org/wiki/PESEL>.

has_valid_checksum (*number*)

Calculates a checksum with the provided algorithm.

class `localflavor.pl.forms.PLPostalCodeField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates as Polish postal code. Valid code is XX-XXX where X is digit.

class `localflavor.pl.forms.PLProvinceSelect` (*attrs=None*)

A select widget with list of Polish administrative provinces as choices.

```
class localflavor.pl.forms.PLREGONField(max_length=None, min_length=None, *args,
                                         **kwargs)
```

A form field that validates its input is a REGON number.

Valid regon number consists of 9 or 14 digits. See http://www.stat.gov.pl/bip/regon_ENG_HTML.htm for more information.

```
has_valid_checksum(number)
```

Calculates a checksum with the provided algorithm.

7.34.2 Data

```
localflavor.pl.pl_administrativeunits.ADMINISTRATIVE_UNIT_CHOICES = (('wroclaw', 'Wrocław'), ('jelenia', 'Jelenia Góra'))
```

•Regions correspond to the Portuguese ‘distritos’ and ‘regiões autónomas’ as per ISO3166:2-PT.

```
class localflavor.pt.forms.PTSocialSecurityNumberField(required=True, width=100, get=None, label=None, initial=None, help_text=None, error_messages=None, show_hidden_initial=False, validators=[], localize=False)
```

A field which validates Portuguese Social Security numbers (locally NISS - ‘Número de Identificação na Segurança Social’).

- Social Security numbers must be in the format XXXXXXXXXXXXX (where X is either 1 or 2 and Y is any other digit).

```
class localflavor.pt.forms.PTZipCodeField(max_length=None,    min_length=None,    *args,
                                           **kwargs)
```

A field which validates Portuguese zip codes.

NOTE - Zip codes have the format XYYY-YYY (where X is a digit between 1 and 9 and Y is any other digit).

7.35.2 Data

`localflavor.pt.pt_regions.REGION_CHOICES = (('01', 'Aveiro'), ('02', 'Beja'), ('03', 'Braga'), ('04', 'Bragança'), ('05', 'Brisbane'))`
A tuple representing Portuguese regions (as per ISO3166:2-PT).

7.36 Paraguay (py)

7.36.1 Forms

PY-specific Form helpers.

class localflavor.py_.forms.**PyDepartmentSelect** (*attrs=None*)
A Select widget with a list of Paraguayan departments as choices.

```
class localflavor.py_.forms.PyNumberedDepartmentSelect (attrs=None)
    A Select widget with a roman numbered list of Paraguayan departments as choices.
```

7.36.2 Data

```
localflavor.py_.py_department.DEPARTMENT_CHOICES = (('AG', 'Alto Paraguay'), ('AA', 'Alto Paraná'), ('AM', 'Amambay'), ('AS', 'Asunción'), ('BO', 'Boquerón'), ('BR', 'Breña Roja'), ('BU', 'Buzos'), ('CA', 'Caaguazú'), ('CB', 'Cabo Corrientes'), ('CC', 'Caazapa'), ('CD', 'Cardenal Amelino'), ('CE', 'Central'), ('CF', 'Concepción'), ('CH', 'Chiriquí'), ('CI', 'Ciudad del Este'), ('CM', 'Comandante Antonio G. Frutos'), ('CN', 'Cordillera'), ('CO', 'Colorado'), ('CP', 'Cruces del Sur'), ('CR', 'Curumanguat'), ('CS', 'Capiatzen'), ('CU', 'Cuata'), ('CY', 'Cyro Negro'), ('DA', 'Deseado'), ('DB', 'Deseado B'), ('DC', 'Deseado C'), ('DE', 'Deseado D'), ('DF', 'Deseado F'), ('DG', 'Deseado G'), ('DH', 'Deseado H'), ('DI', 'Deseado I'), ('DJ', 'Deseado J'), ('DK', 'Deseado K'), ('DL', 'Deseado L'), ('DM', 'Deseado M'), ('DN', 'Deseado N'), ('DO', 'Deseado O'), ('DP', 'Deseado P'), ('DQ', 'Deseado Q'), ('DR', 'Deseado R'), ('DS', 'Deseado S'), ('DT', 'Deseado T'), ('DU', 'Deseado U'), ('DV', 'Deseado V'), ('DW', 'Deseado W'), ('DX', 'Deseado X'), ('DY', 'Deseado Y'), ('DZ', 'Deseado Z'), ('EA', 'Eusebio Ayala'), ('EB', 'El Barrio'), ('EC', 'El Estero'), ('ED', 'El Estero de los Hornos'), ('EE', 'El Estero de los Hornos'), ('EF', 'El Estero de los Hornos'), ('EG', 'El Estero de los Hornos'), ('EH', 'El Estero de los Hornos'), ('EI', 'El Estero de los Hornos'), ('EJ', 'El Estero de los Hornos'), ('EK', 'El Estero de los Hornos'), ('EL', 'El Estero de los Hornos'), ('EM', 'El Estero de los Hornos'), ('EN', 'El Estero de los Hornos'), ('EO', 'El Estero de los Hornos'), ('EP', 'El Estero de los Hornos'), ('EQ', 'El Estero de los Hornos'), ('ER', 'El Estero de los Hornos'), ('ES', 'El Estero de los Hornos'), ('ET', 'El Estero de los Hornos'), ('EU', 'El Estero de los Hornos'), ('EV', 'El Estero de los Hornos'), ('EW', 'El Estero de los Hornos'), ('EX', 'El Estero de los Hornos'), ('EY', 'El Estero de los Hornos'), ('EZ', 'El Estero de los Hornos'), ('FA', 'Fátima'), ('FB', 'Fátima'), ('FC', 'Fátima'), ('FD', 'Fátima'), ('FE', 'Fátima'), ('FF', 'Fátima'), ('FG', 'Fátima'), ('FH', 'Fátima'), ('FI', 'Fátima'), ('FJ', 'Fátima'), ('FK', 'Fátima'), ('FL', 'Fátima'), ('FM', 'Fátima'), ('FN', 'Fátima'), ('FO', 'Fátima'), ('FP', 'Fátima'), ('FQ', 'Fátima'), ('FR', 'Fátima'), ('FS', 'Fátima'), ('FT', 'Fátima'), ('FU', 'Fátima'), ('FV', 'Fátima'), ('FW', 'Fátima'), ('FX', 'Fátima'), ('FY', 'Fátima'), ('FZ', 'Fátima'), ('GA', 'Gallineros'), ('GB', 'Gallineros'), ('GC', 'Gallineros'), ('GD', 'Gallineros'), ('GE', 'Gallineros'), ('GF', 'Gallineros'), ('GG', 'Gallineros'), ('GH', 'Gallineros'), ('GI', 'Gallineros'), ('GJ', 'Gallineros'), ('GK', 'Gallineros'), ('GL', 'Gallineros'), ('GM', 'Gallineros'), ('GN', 'Gallineros'), ('GO', 'Gallineros'), ('GP', 'Gallineros'), ('GQ', 'Gallineros'), ('GR', 'Gallineros'), ('GS', 'Gallineros'), ('GT', 'Gallineros'), ('GU', 'Gallineros'), ('GV', 'Gallineros'), ('GW', 'Gallineros'), ('GX', 'Gallineros'), ('GY', 'Gallineros'), ('GZ', 'Gallineros'), ('HA', 'Hacienda'), ('HB', 'Hacienda'), ('HC', 'Hacienda'), ('HD', 'Hacienda'), ('HE', 'Hacienda'), ('HF', 'Hacienda'), ('HG', 'Hacienda'), ('HH', 'Hacienda'), ('HI', 'Hacienda'), ('HJ', 'Hacienda'), ('HK', 'Hacienda'), ('HL', 'Hacienda'), ('HM', 'Hacienda'), ('HN', 'Hacienda'), ('HO', 'Hacienda'), ('HP', 'Hacienda'), ('HQ', 'Hacienda'), ('HR', 'Hacienda'), ('HS', 'Hacienda'), ('HT', 'Hacienda'), ('HU', 'Hacienda'), ('HV', 'Hacienda'), ('HW', 'Hacienda'), ('HX', 'Hacienda'), ('HY', 'Hacienda'), ('HZ', 'Hacienda'), ('IA', 'Itaipua'), ('IB', 'Itaipua'), ('IC', 'Itaipua'), ('ID', 'Itaipua'), ('IE', 'Itaipua'), ('IF', 'Itaipua'), ('IG', 'Itaipua'), ('IH', 'Itaipua'), ('II', 'Itaipua'), ('IJ', 'Itaipua'), ('IK', 'Itaipua'), ('IL', 'Itaipua'), ('IM', 'Itaipua'), ('IN', 'Itaipua'), ('IO', 'Itaipua'), ('IP', 'Itaipua'), ('IQ', 'Itaipua'), ('IR', 'Itaipua'), ('IS', 'Itaipua'), ('IT', 'Itaipua'), ('IU', 'Itaipua'), ('IV', 'Itaipua'), ('IW', 'Itaipua'), ('IX', 'Itaipua'), ('IY', 'Itaipua'), ('IZ', 'Itaipua'), ('JA', 'Jesús María'), ('JB', 'Jesús María'), ('JC', 'Jesús María'), ('JD', 'Jesús María'), ('JE', 'Jesús María'), ('JF', 'Jesús María'), ('JG', 'Jesús María'), ('JH', 'Jesús María'), ('JI', 'Jesús María'), ('JJ', 'Jesús María'), ('JK', 'Jesús María'), ('JL', 'Jesús María'), ('JM', 'Jesús María'), ('JN', 'Jesús María'), ('JO', 'Jesús María'), ('JP', 'Jesús María'), ('JQ', 'Jesús María'), ('JR', 'Jesús María'), ('JS', 'Jesús María'), ('JT', 'Jesús María'), ('JU', 'Jesús María'), ('JV', 'Jesús María'), ('JW', 'Jesús María'), ('JX', 'Jesús María'), ('JY', 'Jesús María'), ('JZ', 'Jesús María'), ('KA', 'Karai'), ('KB', 'Karai'), ('KC', 'Karai'), ('KD', 'Karai'), ('KE', 'Karai'), ('KF', 'Karai'), ('KG', 'Karai'), ('KH', 'Karai'), ('KI', 'Karai'), ('KJ', 'Karai'), ('KK', 'Karai'), ('KL', 'Karai'), ('KM', 'Karai'), ('KN', 'Karai'), ('KO', 'Karai'), ('KP', 'Karai'), ('KQ', 'Karai'), ('KR', 'Karai'), ('KS', 'Karai'), ('KT', 'Karai'), ('KU', 'Karai'), ('KV', 'Karai'), ('KW', 'Karai'), ('KX', 'Karai'), ('KY', 'Karai'), ('KZ', 'Karai'), ('LA', 'La Alameda'), ('LB', 'La Alameda'), ('LC', 'La Alameda'), ('LD', 'La Alameda'), ('LE', 'La Alameda'), ('LF', 'La Alameda'), ('LG', 'La Alameda'), ('LH', 'La Alameda'), ('LI', 'La Alameda'), ('LJ', 'La Alameda'), ('LK', 'La Alameda'), ('LL', 'La Alameda'), ('LM', 'La Alameda'), ('LN', 'La Alameda'), ('LO', 'La Alameda'), ('LP', 'La Alameda'), ('LQ', 'La Alameda'), ('LR', 'La Alameda'), ('LS', 'La Alameda'), ('LT', 'La Alameda'), ('LU', 'La Alameda'), ('LV', 'La Alameda'), ('LW', 'La Alameda'), ('LX', 'La Alameda'), ('LY', 'La Alameda'), ('LZ', 'La Alameda'), ('MA', 'Mariano Roque Sáenz Peña'), ('MB', 'Mariano Roque Sáenz Peña'), ('MC', 'Mariano Roque Sáenz Peña'), ('MD', 'Mariano Roque Sáenz Peña'), ('ME', 'Mariano Roque Sáenz Peña'), ('MF', 'Mariano Roque Sáenz Peña'), ('MG', 'Mariano Roque Sáenz Peña'), ('MH', 'Mariano Roque Sáenz Peña'), ('MI', 'Mariano Roque Sáenz Peña'), ('MJ', 'Mariano Roque Sáenz Peña'), ('MK', 'Mariano Roque Sáenz Peña'), ('ML', 'Mariano Roque Sáenz Peña'), ('MM', 'Mariano Roque Sáenz Peña'), ('MN', 'Mariano Roque Sáenz Peña'), ('MO', 'Mariano Roque Sáenz Peña'), ('MP', 'Mariano Roque Sáenz Peña'), ('MQ', 'Mariano Roque Sáenz Peña'), ('MR', 'Mariano Roque Sáenz Peña'), ('MS', 'Mariano Roque Sáenz Peña'), ('MT', 'Mariano Roque Sáenz Peña'), ('MU', 'Mariano Roque Sáenz Peña'), ('MV', 'Mariano Roque Sáenz Peña'), ('MW', 'Mariano Roque Sáenz Peña'), ('MX', 'Mariano Roque Sáenz Peña'), ('MY', 'Mariano Roque Sáenz Peña'), ('MZ', 'Mariano Roque Sáenz Peña'), ('NA', 'Naranja'), ('NB', 'Naranja'), ('NC', 'Naranja'), ('ND', 'Naranja'), ('NE', 'Naranja'), ('NF', 'Naranja'), ('NG', 'Naranja'), ('NH', 'Naranja'), ('NI', 'Naranja'), ('NJ', 'Naranja'), ('NK', 'Naranja'), ('NL', 'Naranja'), ('NM', 'Naranja'), ('NN', 'Naranja'), ('NO', 'Naranja'), ('NP', 'Naranja'), ('NQ', 'Naranja'), ('NR', 'Naranja'), ('NS', 'Naranja'), ('NT', 'Naranja'), ('NU', 'Naranja'), ('NV', 'Naranja'), ('NW', 'Naranja'), ('NX', 'Naranja'), ('NY', 'Naranja'), ('NZ', 'Naranja'), ('OA', 'Omar'), ('OB', 'Omar'), ('OC', 'Omar'), ('OD', 'Omar'), ('OE', 'Omar'), ('OF', 'Omar'), ('OG', 'Omar'), ('OH', 'Omar'), ('OI', 'Omar'), ('OJ', 'Omar'), ('OK', 'Omar'), ('OL', 'Omar'), ('OM', 'Omar'), ('ON', 'Omar'), ('OO', 'Omar'), ('OP', 'Omar'), ('OQ', 'Omar'), ('OR', 'Omar'), ('OS', 'Omar'), ('OT', 'Omar'), ('OU', 'Omar'), ('OV', 'Omar'), ('OW', 'Omar'), ('OX', 'Omar'), ('OY', 'Omar'), ('OZ', 'Omar'), ('PA', 'Palmira'), ('PB', 'Palmira'), ('PC', 'Palmira'), ('PD', 'Palmira'), ('PE', 'Palmira'), ('PF', 'Palmira'), ('PG', 'Palmira'), ('PH', 'Palmira'), ('PI', 'Palmira'), ('PJ', 'Palmira'), ('PK', 'Palmira'), ('PL', 'Palmira'), ('PM', 'Palmira'), ('PN', 'Palmira'), ('PO', 'Palmira'), ('PP', 'Palmira'), ('PQ', 'Palmira'), ('PR', 'Palmira'), ('PS', 'Palmira'), ('PT', 'Palmira'), ('PU', 'Palmira'), ('PV', 'Palmira'), ('PW', 'Palmira'), ('PX', 'Palmira'), ('PY', 'Palmira'), ('PZ', 'Palmira'), ('QA', 'Quevedo'), ('QB', 'Quevedo'), ('QC', 'Quevedo'), ('QD', 'Quevedo'), ('QE', 'Quevedo'), ('QF', 'Quevedo'), ('QG', 'Quevedo'), ('QH', 'Quevedo'), ('QI', 'Quevedo'), ('QJ', 'Quevedo'), ('QK', 'Quevedo'), ('QL', 'Quevedo'), ('QM', 'Quevedo'), ('QN', 'Quevedo'), ('QO', 'Quevedo'), ('QP', 'Quevedo'), ('QQ', 'Quevedo'), ('QR', 'Quevedo'), ('QS', 'Quevedo'), ('QT', 'Quevedo'), ('QU', 'Quevedo'), ('QV', 'Quevedo'), ('QW', 'Quevedo'), ('QX', 'Quevedo'), ('QY', 'Quevedo'), ('QZ', 'Quevedo'), ('RA', 'Río de la Plata'), ('RB', 'Río de la Plata'), ('RC', 'Río de la Plata'), ('RD', 'Río de la Plata'), ('RE', 'Río de la Plata'), ('RF', 'Río de la Plata'), ('RG', 'Río de la Plata'), ('RH', 'Río de la Plata'), ('RI', 'Río de la Plata'), ('RJ', 'Río de la Plata'), ('RK', 'Río de la Plata'), ('RL', 'Río de la Plata'), ('RM', 'Río de la Plata'), ('RN', 'Río de la Plata'), ('RO', 'Río de la Plata'), ('RP', 'Río de la Plata'), ('RQ', 'Río de la Plata'), ('RR', 'Río de la Plata'), ('RS', 'Río de la Plata'), ('RT', 'Río de la Plata'), ('RU', 'Río de la Plata'), ('RV', 'Río de la Plata'), ('RW', 'Río de la Plata'), ('RX', 'Río de la Plata'), ('RY', 'Río de la Plata'), ('RZ', 'Río de la Plata'), ('SA', 'San Andrés'), ('SB', 'San Andrés'), ('SC', 'San Andrés'), ('SD', 'San Andrés'), ('SE', 'San Andrés'), ('SF', 'San Andrés'), ('SG', 'San Andrés'), ('SH', 'San Andrés'), ('SI', 'San Andrés'), ('SJ', 'San Andrés'), ('SK', 'San Andrés'), ('SL', 'San Andrés'), ('SM', 'San Andrés'), ('SN', 'San Andrés'), ('SO', 'San Andrés'), ('SP', 'San Andrés'), ('SQ', 'San Andrés'), ('SR', 'San Andrés'), ('SS', 'San Andrés'), ('ST', 'San Andrés'), ('SU', 'San Andrés'), ('SV', 'San Andrés'), ('SW', 'San Andrés'), ('SX', 'San Andrés'), ('SY', 'San Andrés'), ('SZ', 'San Andrés'), ('TA', 'Tapiá'), ('TB', 'Tapiá'), ('TC', 'Tapiá'), ('TD', 'Tapiá'), ('TE', 'Tapiá'), ('TF', 'Tapiá'), ('TG', 'Tapiá'), ('TH', 'Tapiá'), ('TI', 'Tapiá'), ('TJ', 'Tapiá'), ('TK', 'Tapiá'), ('TL', 'Tapiá'), ('TM', 'Tapiá'), ('TN', 'Tapiá'), ('TO', 'Tapiá'), ('TP', 'Tapiá'), ('TQ', 'Tapiá'), ('TR', 'Tapiá'), ('TS', 'Tapiá'), ('TT', 'Tapiá'), ('TU', 'Tapiá'), ('TV', 'Tapiá'), ('TW', 'Tapiá'), ('TX', 'Tapiá'), ('TY', 'Tapiá'), ('TZ', 'Tapiá'), ('UA', 'Uyapiti'), ('UB', 'Uyapiti'), ('UC', 'Uyapiti'), ('UD', 'Uyapiti'), ('UE', 'Uyapiti'), ('UF', 'Uyapiti'), ('UG', 'Uyapiti'), ('UH', 'Uyapiti'), ('UI', 'Uyapiti
```

```
localflavor.py_.py_department.DEPARTMENT_ROMAN_CHOICES = (('CN', 'I Concepción'), ('SP', 'II San Pedro'), ('  
list of departments sorted by its roman number
```

7.37 Romania (ro)

7.37.1 Forms

Romanian specific form helpers.

class `localflavor.ro.forms.ROCIFField` (*max_length=10, min_length=2, *args, **kwargs*)
 A Romanian fiscal identity code (CIF) field

For CIF validation algorithm see <http://www.validari.ro/cui.html>

clean (*value*)
 CIF validation

class `localflavor.ro.forms.ROCNPFfield` (*max_length=13, min_length=13, *args, **kwargs*)
 A Romanian personal identity code (CNP) field

For CNP validation algorithm see <http://www.validari.ro/cnp.html>

clean (*value*)
 CNP validations

class `localflavor.ro.forms.ROCountyField` (*required=True, widget=None, label=None, initial=None, help_text=None, error_messages=None, show_hidden_initial=False, validators=[], localize=False*)

A form field that validates its input is a Romanian county name or abbreviation. It normalizes the input to the standard vehicle registration abbreviation for the given county

WARNING: This field will only accept names written with diacritics; consider using `ROCountySelect` if this behavior is unacceptable for you

Example:

Argeş => valid Arges => invalid

class `localflavor.ro.forms.ROCountySelect` (*attrs=None*)
 A Select widget that uses a list of Romanian counties (judete) as its choices.

class `localflavor.ro.forms.ROIBANField` (*max_length=40, min_length=24, *args, **kwargs*)
 Romanian International Bank Account Number (IBAN) field

For Romanian IBAN validation algorithm see <http://validari.ro/iban.html>

clean (*value*)
 Strips - and spaces, performs country code and checksum validation

class `localflavor.ro.forms.ROPhoneNumberField` (*max_length=20, min_length=10, *args, **kwargs*)
 Romanian phone number field

clean (*value*)
 Strips -, (,) and spaces. Checks the final length.

class `localflavor.ro.forms.ROPostalCodeField` (*max_length=6, min_length=6, *args, **kwargs*)
 Romanian postal code field.

7.37.2 Data

`localflavor.ro.ro_counties.COUNTIES_CHOICES` = (('AB', 'Alba'), ('AR', 'Arad'), ('AG', 'Argeş'), ('BC', 'Bacău'),
 A list of Romanian counties as *choices* in a formfield.

7.38 Russia (ru)

7.38.1 Forms

Russian-specific forms helpers

```
class localflavor.ru.forms.RUAlienPassportNumberField(max_length=None,
                                                    min_length=None,      *args,
                                                    **kwargs)
```

Russian alien's passport number format: XX XXXXXXXX where X - any digit.

```
class localflavor.ru.forms.RUCountySelect(attrs=None)
```

A Select widget that uses a list of Russian Counties as its choices.

```
class localflavor.ru.forms.RUPassportNumberField(max_length=None,  min_length=None,
                                                    *args, **kwargs)
```

Russian internal passport number format: XXXX XXXXXX where X - any digit.

```
class localflavor.ru.forms.RUPostalCodeField(max_length=None, min_length=None, *args,
                                                    **kwargs)
```

Russian Postal code field. Format: XXXXXX, where X is any digit, and first digit is not zero.

```
class localflavor.ru.forms.RURegionSelect(attrs=None)
```

A Select widget that uses a list of Russian Regions as its choices.

7.38.2 Data

```
localflavor.ru.ru_regions.RU_COUNTY_CHOICES = (('Central Federal County', 'Central Federal County'), ('South I  
http://ru.wikipedia.org/wiki/__
```

```
localflavor.ru.ru_regions.RU_REGIONS_CHOICES = (('77', 'Moskva'), ('78', 'Saint-Peterburg'), ('50', 'Moskovskay  
http://ru.wikipedia.org/wiki/__
```

7.39 Sweden (se)

7.39.1 Forms

Swedish specific Form helpers

```
class localflavor.se.forms.SECountySelect(attrs=None)
```

A Select form widget that uses a list of the Swedish counties (län) as its choices.

The cleaned value is the official county code – see http://en.wikipedia.org/wiki/Counties_of_Sweden for a list.

```
class localflavor.se.forms.SEOrganisationNumberField(max_length=None,
                                                    min_length=None,      *args,
                                                    **kwargs)
```

A form field that validates input as a Swedish organisation number (organisationsnummer).

It accepts the same input as SEPersonalIdentityField (for sole proprietorships (enskild firma). However, co-ordination numbers are not accepted.

It also accepts ordinary Swedish organisation numbers with the format NNNNNNNNNN.

The return value will be YYYYMMDDXXXXX for sole proprietors, and NNNNNNNNNNN for other organisations.

class `localflavor.se.forms.SEPersonalIdentityNumberField` (*coordination_number=True, *args, **kwargs*)

A form field that validates input as a Swedish personal identity number (personnummer).

The correct formats are YYYYMMDD-XXXX, YYYYMMDDXXXX, YYMMDD-XXXX, YYMMDDXXXX and YYMMDD+XXXX.

A + indicates that the person is older than 100 years, which will be taken into consideration when the date is validated.

The checksum will be calculated and checked. The birth date is checked to be a valid date.

By default, co-ordination numbers (samordningsnummer) will be accepted. To only allow real personal identity numbers, pass the keyword argument `coordination_number=False` to the constructor.

The cleaned value will always have the format YYYYMMDDXXXX.

class `localflavor.se.forms.SEPostalCodeField` (**args, **kwargs*)

A form field that validates input as a Swedish postal code (postnummer). Valid codes consist of five digits (XXXXX). The number can optionally be formatted with a space after the third digit (XXX XX).

The cleaned value will never contain the space.

7.39.2 Utilities

`localflavor.se.utils.id_number_checksum` (*gd*)

Calculates a Swedish ID number checksum, using the “Luhn”-algorithm

`localflavor.se.utils.validate_id_birthdate` (*gd, fix_coordination_number_day=True*)

Validates the birth_date and returns the datetime.date object for the birth_date.

If the date is an invalid birth day, a ValueError will be raised.

7.39.3 Data

`localflavor.se.se_counties.COUNTY_CHOICES` = (('AB', 'Stockholm'), ('AC', 'Västerbotten'), ('BD', 'Norrbotten'),

An alphabetical list of Swedish counties, sorted by codes. http://en.wikipedia.org/wiki/Counties_of_Sweden

7.40 Slovenia (si)

7.40.1 Forms

Slovenian specific form helpers.

class `localflavor.si.forms.SIEMSOField` (*max_length=None, min_length=None, *args, **kwargs*)

A form for validating Slovenian personal identification number.

Additionally stores gender, nationality and birthday to self.info dictionary.

class `localflavor.si.forms.SIPhoneNumberField` (*max_length=None, min_length=None, *args, **kwargs*)

Slovenian phone number field.

Phone number must contain at least local area code. Country code can be present.

Examples:

- +38640XXXXXX
- 0038640XXXXXX
- 040XXXXXX
- 01XXXXXX
- 0590XXXXX

```
class localflavor.si.forms.SIPostalCodeField(*args, **kwargs)
    Slovenian post codes field.
```

class `localflavor.si.forms.SIPostalCodeSelect` (*attrs=None*)
A Select widget that uses Slovenian postal codes as its choices.

```
class localflavor.si.forms.SITaxNumberField(max_length=None, min_length=None, *args,  
                                             **kwargs)
```

Slovenian tax number field.

Valid input is SIXXXXXXXXX or XXXXXXXXX where X is a number.

7.40.2 Data

```
localflavor.si.si_postalcodes.SI_POSTALCODES = ((1000, 'Ljubljana'), (1215, 'Medvode'), (1216, 'Smlednik'), (12
```

7.42 Turkey (tr)

7.42.1 Forms

```
class localflavor.tr.forms.TRIdentificationNumberField(required=True,          wid-
                                                         get=None, label=None, ini-
                                                         tial=None, help_text=None,
                                                         error_messages=None,
                                                         show_hidden_initial=False,
                                                         validators=[], localize=False)
```

A Turkey Identification Number number. See: http://tr.wikipedia.org/wiki/T%C3%BCrkiye_Cumhuriyeti_Kimlik_Numaras%C4%9F

Checks the following rules to determine whether the number is valid:

- The number is 11-digits.
- First digit is not 0.
- Conforms to the following two formula: $(\text{sum}(\text{1st, 3rd, 5th, 7th, 9th}) * 7 - \text{sum}(\text{2nd, 4th, 6th, 8th})) \% 10 = 10\text{th digit}$ $\text{sum}(\text{1st to 10th}) \% 10 = 11\text{th digit}$

```
class localflavor.tr.forms.TRPhoneNumberField(max_length=None, min_length=None, *args,
                                                         **kwargs)
```

A form field that validates input as a Turkish phone number. The correct format is 0xxx xxx xxxx. +90xxx xxx xxxx and inputs without spaces also validates. The result is normalized to xxx xxx xxxx format.

```
class localflavor.tr.forms.TRPostalCodeField(max_length=5, min_length=5, *args,
                                                         **kwargs)
```

A form field that validates input as a Turkish zip code. Valid codes consist of five digits.

```
class localflavor.tr.forms.TRProvinceSelect(attrs=None)
```

A Select widget that uses a list of provinces in Turkey as its choices.

7.42.2 Data

```
localflavor.tr.tr_provinces.PROVINCE_CHOICES = (('01', 'Adana'), ('02', 'Adiyaman'), ('03', 'Afyonkarahisar'), (
```

A list of Turkish provinces

7.43 United States of America (us)

7.43.1 Forms

USA-specific Form helpers

```
class localflavor.us.forms.USPSSelect(attrs=None)
```

A Select widget that uses a list of US Postal Service codes as its choices.

```
class localflavor.us.forms.USPhoneNumberField(max_length=None, min_length=None, *args,
                                                         **kwargs)
```

A form field that validates input as a U.S. phone number.

```
class localflavor.us.forms.USSocialSecurityNumberField(required=True, widget=None, label=None, initial=None, help_text=None, error_messages=None, show_hidden_initial=False, validators=[], localize=False)
```

A United States Social Security number.

Checks the following rules to determine whether the number is valid:

- Conforms to the XXX-XX-XXXX format.
- No group consists entirely of zeroes.
- The leading group is not “666” (block “666” will never be allocated).
- The number is not in the promotional block 987-65-4320 through 987-65-4329, which are permanently invalid.
- The number is not one known to be invalid due to otherwise widespread promotional use or distribution (e.g., the Woolworth’s number or the 1962 promotional number).

```
class localflavor.us.forms.USStateField(required=True, widget=None, label=None, initial=None, help_text=None, error_messages=None, show_hidden_initial=False, validators=[], localize=False)
```

A form field that validates its input is a U.S. state name or abbreviation. It normalizes the input to the standard two-letter postal service abbreviation for the given state.

```
class localflavor.us.forms.USStateSelect(attrs=None)
```

A Select widget that uses a list of U.S. states/territories as its choices.

```
class localflavor.us.forms.USZipCodeField(max_length=None, min_length=None, *args, **kwargs)
```

” A form field that validates input as a U.S. ZIP code. Valid formats are XXXXX or XXXXX-XXXX.

7.43.2 Models

```
class localflavor.us.models.PhoneNumberField(*args, **kwargs)
```

A `CharField` that checks that the value is a valid U.S.A.-style phone number (in the format XXX-XXX-XXXX).

```
class localflavor.us.models.USPostalCodeField(*args, **kwargs)
```

” A model field that forms represent as a `forms.USPSSelect` field and stores the two-letter U.S Postal Service abbreviation in the database.

```
class localflavor.us.models.USStateField(*args, **kwargs)
```

A model field that forms represent as a `forms.USStateField` field and stores the two-letter U.S. state abbreviation in the database.

7.43.3 Data

```
localflavor.us.us_states.CONTIGUOUS_STATES = (('AL', 'Alabama'), ('AZ', 'Arizona'), ('AR', 'Arkansas'), ('CA', 'California'), ('CO', 'Colorado'), ('CT', 'Connecticut'), ('DE', 'Delaware'), ('FL', 'Florida'), ('GA', 'Georgia'), ('HI', 'Hawaii'), ('IL', 'Illinois'), ('IN', 'Indiana'), ('IOWA', 'Iowa'), ('KS', 'Kansas'), ('KY', 'Kentucky'), ('LA', 'Louisiana'), ('MA', 'Massachusetts'), ('MD', 'Maryland'), ('ME', 'Maine'), ('MI', 'Michigan'), ('MN', 'Minnesota'), ('MO', 'Missouri'), ('NE', 'Nebraska'), ('NH', 'New Hampshire'), ('NJ', 'New Jersey'), ('NM', 'New Mexico'), ('NY', 'New York'), ('NC', 'North Carolina'), ('ND', 'North Dakota'), ('OH', 'Ohio'), ('OK', 'Oklahoma'), ('OR', 'Oregon'), ('PA', 'Pennsylvania'), ('RI', 'Rhode Island'), ('SC', 'South Carolina'), ('SD', 'South Dakota'), ('TN', 'Tennessee'), ('TX', 'Texas'), ('UT', 'Utah'), ('VA', 'Virginia'), ('VT', 'Vermont'), ('WA', 'Washington'), ('WI', 'Wisconsin'), ('WY', 'Wyoming'))
```

The 48 contiguous states, plus the District of Columbia.

```
localflavor.us.us_states.US_STATES = (('AL', 'Alabama'), ('AK', 'Alaska'), ('AZ', 'Arizona'), ('AR', 'Arkansas'), ('CA', 'California'), ('CO', 'Colorado'), ('CT', 'Connecticut'), ('DE', 'Delaware'), ('FL', 'Florida'), ('GA', 'Georgia'), ('HI', 'Hawaii'), ('IL', 'Illinois'), ('IN', 'Indiana'), ('IOWA', 'Iowa'), ('KS', 'Kansas'), ('KY', 'Kentucky'), ('LA', 'Louisiana'), ('MA', 'Massachusetts'), ('MD', 'Maryland'), ('ME', 'Maine'), ('MI', 'Michigan'), ('MN', 'Minnesota'), ('MO', 'Missouri'), ('NE', 'Nebraska'), ('NH', 'New Hampshire'), ('NJ', 'New Jersey'), ('NM', 'New Mexico'), ('NY', 'New York'), ('NC', 'North Carolina'), ('ND', 'North Dakota'), ('OH', 'Ohio'), ('OK', 'Oklahoma'), ('OR', 'Oregon'), ('PA', 'Pennsylvania'), ('RI', 'Rhode Island'), ('SC', 'South Carolina'), ('SD', 'South Dakota'), ('TN', 'Tennessee'), ('TX', 'Texas'), ('UT', 'Utah'), ('VA', 'Virginia'), ('VT', 'Vermont'), ('WA', 'Washington'), ('WI', 'Wisconsin'), ('WY', 'Wyoming'))
```

All 50 states, plus the District of Columbia.

`localflavor.us.us_states.US_TERRITORIES` = (('AS', 'American Samoa'), ('GU', 'Guam'), ('MP', 'Northern Marianas'), ('PW', 'Palau'))
 Non-state territories.

`localflavor.us.us_states.ARMED_FORCES_STATES` = (('AA', 'Armed Forces Americas'), ('AE', 'Armed Forces Europe'))
 Military postal “states”. Note that ‘AE’ actually encompasses Europe, Canada, Africa and the Middle East.

`localflavor.us.us_states.COFA_STATES` = (('FM', 'Federated States of Micronesia'), ('MH', 'Marshall Islands'), ('PW', 'Palau'))
 Non-US locations serviced by USPS (under Compact of Free Association).

`localflavor.us.us_states.OBSELETE_STATES` = (('CM', 'Commonwealth of the Northern Mariana Islands'), ('CZ', 'Czech Republic'))
 Obsolete abbreviations (no longer US territories/USPS service, or code changed).

`localflavor.us.us_states.STATE_CHOICES` = (('AL', 'Alabama'), ('AK', 'Alaska'), ('AS', 'American Samoa'), ('AZ', 'Arizona'), ('CA', 'California'), ('CO', 'Colorado'), ('CT', 'Connecticut'), ('DE', 'Delaware'), ('FL', 'Florida'), ('GA', 'Georgia'), ('HI', 'Hawaii'), ('IL', 'Illinois'), ('IN', 'Indiana'), ('IOWA', 'Iowa'), ('KS', 'Kansas'), ('KY', 'Kentucky'), ('LA', 'Louisiana'), ('MA', 'Massachusetts'), ('MD', 'Maryland'), ('ME', 'Maine'), ('MI', 'Michigan'), ('MN', 'Minnesota'), ('MO', 'Missouri'), ('NE', 'Nebraska'), ('NH', 'New Hampshire'), ('NJ', 'New Jersey'), ('NM', 'New Mexico'), ('NY', 'New York'), ('OH', 'Ohio'), ('OK', 'Oklahoma'), ('OR', 'Oregon'), ('PA', 'Pennsylvania'), ('RI', 'Rhode Island'), ('SC', 'South Carolina'), ('SD', 'South Dakota'), ('Tennessee', 'Tennessee'), ('TX', 'Texas'), ('UT', 'Utah'), ('VA', 'Virginia'), ('VT', 'Vermont'), ('WA', 'Washington'), ('WI', 'Wisconsin'), ('WY', 'Wyoming'))
 All US states and territories plus DC and military mail.

`localflavor.us.us_states.USPS_CHOICES` = (('AL', 'Alabama'), ('AK', 'Alaska'), ('AS', 'American Samoa'), ('AZ', 'Arizona'), ('CA', 'California'), ('CO', 'Colorado'), ('CT', 'Connecticut'), ('DE', 'Delaware'), ('FL', 'Florida'), ('GA', 'Georgia'), ('HI', 'Hawaii'), ('IL', 'Illinois'), ('IN', 'Indiana'), ('IOWA', 'Iowa'), ('KS', 'Kansas'), ('KY', 'Kentucky'), ('LA', 'Louisiana'), ('MA', 'Massachusetts'), ('MD', 'Maryland'), ('ME', 'Maine'), ('MI', 'Michigan'), ('MN', 'Minnesota'), ('MO', 'Missouri'), ('NE', 'Nebraska'), ('NH', 'New Hampshire'), ('NJ', 'New Jersey'), ('NM', 'New Mexico'), ('NY', 'New York'), ('OH', 'Ohio'), ('OK', 'Oklahoma'), ('OR', 'Oregon'), ('PA', 'Pennsylvania'), ('RI', 'Rhode Island'), ('SC', 'South Carolina'), ('SD', 'South Dakota'), ('Tennessee', 'Tennessee'), ('TX', 'Texas'), ('UT', 'Utah'), ('VA', 'Virginia'), ('VT', 'Vermont'), ('WA', 'Washington'), ('WI', 'Wisconsin'), ('WY', 'Wyoming'))
 All US Postal Service locations.

`localflavor.us.us_states.STATES_NORMALIZED` = {'gu': 'GU', 'nc': 'NC', 'co': 'CO', 'colorado': 'CO', 'neb': 'NE', 'nebraska': 'NE', 'ny': 'NY', 'new_york': 'NY', 'oh': 'OH', 'ok': 'OK', 'oklahoma': 'OK', 'or': 'OR', 'oregon': 'OR', 'pa': 'PA', 'pennsylvania': 'PA', 'ri': 'RI', 'rhode_island': 'RI', 'sc': 'SC', 'south_carolina': 'SC', 'sd': 'SD', 'south_dakota': 'SD', 'tn': 'TN', 'tennessee': 'TN', 'tx': 'TX', 'utah': 'UT', 'va': 'VA', 'vt': 'VT', 'vermont': 'VT', 'wa': 'WA', 'washington': 'WA', 'wi': 'WI', 'wisconsin': 'WI', 'wy': 'WY', 'wyoming': 'WY'}
 Normalized versions of state names

7.44 Uruguay (uy)

7.44.1 Forms

UY-specific form helpers.

class `localflavor.uy.forms.UYCIField` (*args, **kwargs)
 A field that validates Uruguayan ‘Cedula de identidad’ (CI) numbers.

clean (value)
 Validates format and validation digit.

The official format is [X.]XXX.XXX-X but usually dots and/or slash are omitted so, when validating, those characters are ignored if found in the correct place. The three typically used formats are supported: [X]XXXXXXXX, [X]XXXXXX-X and [X.]XXX.XXX-X.

class `localflavor.uy.forms.UYDepartmentSelect` (attrs=None)
 A Select widget that uses a list of Uruguayan departments as its choices.

7.44.2 Utilities

`localflavor.uy.util.get_validation_digit` (number)
 Calculates the validation digit for the given number.

7.44.3 Data

`localflavor.uy.uy_departments.DEPARTMENT_CHOICES` = (('G', 'Artigas'), ('A', 'Canelones'), ('E', 'Cerro Largo'), ('F', 'Florida'), ('M', 'Maldonado'), ('N', 'Montevideo'), ('P', 'Punta del Este'), ('R', 'Rivadavia'), ('S', 'San Jose del Rio'), ('T', 'Tandil'), ('U', 'Uruguay'))
 A list of Uruguayan departments as *choices* in a formfield.

7.45 South Africa (za)

7.45.1 Forms

South Africa-specific Form helpers

class `localflavor.za.forms.ZAIDField` (*max_length=None, min_length=None, *args, **kwargs*)
 A form field for South African ID numbers – the checksum is validated using the Luhn checksum, and uses a simplistic (read: not entirely accurate) check for the birthdate

class `localflavor.za.forms.ZAPostCodeField` (*max_length=None, min_length=None, *args, **kwargs*)
 A form field that validates input as a South African postcode. Valid postcodes must have four digits.

class `localflavor.za.forms.ZAProvinceSelect` (*attrs=None*)
 A Select widget that uses a list of South African Provinces as its choices.

7.45.2 Data

`localflavor.za.za_provinces.PROVINCE_CHOICES` = (('EC', 'Eastern Cape'), ('FS', 'Free State'), ('GP', 'Gauteng'), ...)
 A list of South African provinces as *choices* in a formfield.

7.46 Generic helpers

7.46.1 Forms

class `localflavor.generic.forms.DateField` (*input_formats=None, *args, **kwargs*)
 A date input field which uses non-US date input formats by default.

class `localflavor.generic.forms.DateTimeField` (*input_formats=None, *args, **kwargs*)
 A date and time input field which uses non-US date and time input formats by default.

class `localflavor.generic.forms.SplitDateTimeField` (*input_date_formats=None, input_time_formats=None, *args, **kwargs*)
 Split date and time input fields which use non-US date and time input formats by default.

PYTHON MODULE INDEX

I

- `localflavor`, 3
- `localflavor.ar.forms`, 15
- `localflavor.at.forms`, 15
- `localflavor.au.forms`, 16
- `localflavor.au.models`, 16
- `localflavor.be.forms`, 17
- `localflavor.br.forms`, 18
- `localflavor.ca.forms`, 18
- `localflavor.ch.forms`, 19
- `localflavor.cl.forms`, 20
- `localflavor.cn.forms`, 21
- `localflavor.co.forms`, 22
- `localflavor.cz.forms`, 22
- `localflavor.de.forms`, 23
- `localflavor.ec.forms`, 23
- `localflavor.es.forms`, 24
- `localflavor.fi.forms`, 25
- `localflavor.fr.forms`, 25
- `localflavor.gb.forms`, 26
- `localflavor.generic.forms`, 49
- `localflavor.hk.forms`, 27
- `localflavor.hr.forms`, 27
- `localflavor.id_.forms`, 29
- `localflavor.ie.forms`, 30
- `localflavor.il.forms`, 30
- `localflavor.in_.forms`, 31
- `localflavor.is_.forms`, 31
- `localflavor.it.forms`, 32
- `localflavor.it.util`, 32
- `localflavor.jp.forms`, 33
- `localflavor.kw.forms`, 33
- `localflavor.lt.forms`, 33
- `localflavor.mk.forms`, 34
- `localflavor.mk.models`, 35
- `localflavor.mx.forms`, 35
- `localflavor.mx.models`, 36
- `localflavor.nl.forms`, 37
- `localflavor.no.forms`, 38
- `localflavor.pe.forms`, 38
- `localflavor.pl.forms`, 39
- `localflavor.pt.forms`, 40
- `localflavor.py_.forms`, 41
- `localflavor.ro.forms`, 41
- `localflavor.ru.forms`, 43
- `localflavor.se.forms`, 43
- `localflavor.se.utils`, 44
- `localflavor.si.forms`, 44
- `localflavor.sk.forms`, 45
- `localflavor.tr.forms`, 46
- `localflavor.us.forms`, 46
- `localflavor.us.models`, 47
- `localflavor.uy.forms`, 48
- `localflavor.uy.util`, 48
- `localflavor.za.forms`, 49