NAME

      Sort::Key::Maker − multi−key sorter creator

SYNOPSIS

```
# create a function that sorts strings by length:
use Sort::Key::Maker sort_by_length => sub { length $_},  qw(integer);

# create a multi-key sort function;
# first key is integer sorted in descending order,
# second key is a string in default (ascending) order:
use Sort::Key::Maker ri_s_keysort => qw(-integer string);

# some sample data...
my @foo = qw(foo bar t too tood mama);

# and now, use the sorter functions previously made:

# get the values on @foo sorted by length:
my @sorted = sort_by_length @foo;

# sort @foo inplace by its length and then by its value:
ri_s_keysort_inplace { length $_, $_ } @foo;
```

DESCRIPTION

      Sort::Key::Maker is a pragmatic module that provides an easy to use interface to Sort::Key multi-key sorting functionality.

      It creates multi-key sorting functions on the fly for any key type combination and exports them to the caller package.

      The key types natively accepted are:

```
string, str, locale, loc, integer, int,
unsigned_integer, uint, number, num
```

      and support for other types can be added via Sort::Key::Register (or also via **Sort::Key::register_type()**).

     USAGE

      use Sort::Key::Maker foo_sort => `@keys`;

          exports two subroutines to the caller package: `foo_sort  (&@)` and `foo_sort_inplace` `(&\@)`.

          Those two subroutines require a sub reference as their first argument and then respectively, the list to be sorted or an array.

          For instance:

```
use Sort::Key::Maker bar_sort => qw(int int str);

@bar=qw(doo tomo 45s tio);
@sorted = bar_sort { unpack "CCs", $_ } @bar;
# or sorting @bar inplace
bar_sort_inplace { unpack "CCs", $_ } @bar;
```

      use Sort::Key::Maker foo_sort => \&genmultikey, `@keys`;

          when the first argument after the sorter name is a reference to a subroutine it is used as the multi-key extraction function. The generated sorter functions doesn't require neither accept one, i.e.:

```
use Sort::Key::Maker sort_by_length => sub { length $_ }, 'int';
my @sorted = sort_by_length qw(foo goo h mama picasso);
```

SEE ALSO

Sort::Key, Sort::Key::Register.

Sort::Maker also available from CPAN provides similar functionality.

AUTHOR

Salvador Fandiño, <sfandino@yahoo.com>

COPYRIGHT AND LICENSE

Copyright (C) 2005, 2014 by Salvador Fandiño

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself, either Perl version 5.8.4 or, at your option, any later version of Perl 5 you may have available.