


Модуль locale в Python, региональные настройки



sberbank.ru РЕКЛАМА

Больше информации на сайте рекламодателя

Подробнее

[Стандартная библиотека Python3.](#) / Модуль locale в Python, региональные настройки

Региональные настройки и стандарты в приложении

[Модуль locale](#) открывает доступ к базе данных локали POSIX и ее функциональности. Механизм локализации POSIX позволяет решать определенные региональные настройки и стандарты в приложении, не требуя, чтобы программист знал особенности страны, где выполняется программное обеспечение.

Модуль locale реализован поверх модуля `_locale`, который, в свою очередь, использует реализацию локали ANSI языка C, если она доступна.

Стандарт языка C определяет локаль как свойство всей программы, которое может быть относительно дорогим для изменения. Частые изменения локали могут вызвать дамп ядра. Это делает локаль несколько болезненной для правильного использования.

Когда программа запускается то подгружается языковой стандарт C, независимо от предпочтительного языкового стандарта пользователя. Есть одно исключение: при запуске [категория locale.LC_CTYPE](#) устанавливает текущую кодировку локали в соответствии с предпочтительной кодировкой локали пользователя. Программа должна явно вызвать [locale.setlocale\(locale.LC_ALL, ''\)](#), чтобы пользовательские настройки локали были выбраны для [других категорий](#).

Плохая идея вызывать `locale.setlocale()` в некоторой библиотечной подпрограмме, поскольку в качестве побочного эффекта это влияет на всю программу. Сохранять и восстанавливать состояние локали так же плохо: это дорого и влияет на другие потоки, которые запускаются до восстановления настроек.

Если при создании модуля общего пользования нужна независимая от локали версия операции, на которую влияет локаль, такие как определенные форматы, используемые с [time.strftime\(\)](#), то нужно будет найти способ сделать это без использования стандартной библиотечной процедуры. Только в крайнем случае нужно документировать, что созданный модуль общего пользования не совместим с настройками не-C локали.

Единственный способ выполнить числовые операции в соответствии с локалью - это использовать специальные функции, определенные этим модулем: [locale.atof\(\)](#), [locale.atoi\(\)](#), [locale.format\(\)](#), [locale.str\(\)](#).

Невозможно выполнить преобразование регистра и классификацию символов в зависимости от локали. Для [текстовых строк](#) (Unicode) они выполняются только в соответствии со значением символа, в то время как для [байтовых строк](#) преобразования и классификации выполняются в соответствии со значением ASCII байта и байтов, чей старший бит установлен (т. е. байты не ASCII), никогда не конвертируются и не считаются частью класса символов, такого как буква или пробел.

Содержание раздела:

- [КРАТКИЙ ОБЗОР МАТЕРИАЛА.](#)
- [Функция setlocale\(\) модуля locale](#)
- [Функция localeconv\(\) модуля locale](#)
- Вверх

[getdefaultlocale\(\) модуля locale](#)
- [Функция getlocale\(\) модуля locale](#)

- [Функция getpreferredencoding\(\) модуля locale](#)
- [Функция getencoding\(\) модуля locale](#)
- [Категории локали модуля locale](#)
- [Другие функции модуля locale](#)

ХОЧУ ПОМОЧЬ
ПРОЕКТУ

РЕКЛАМА



12+. Сроки проведения с 18.09.23 по 30.09.23. Товары и условия их распродажи определяются продавцами таких товаров и отмечены значком «Распродажа». Представлены собирательные образы товаров.