

ХОЧУ ПОМОЧЬ
ПРОЕКТУ

Модуль iccream в Python



study.skysmart.ru

РЕКЛАМА

Учим детей программированию на основе реальных заданий

4,8 ★ Рейтинг организации ⓘ

Узнать больше

gb.ru

Курс: искусственный интеллект с 0! Осенняя распродажа!

5,0 ★ Рейтинг организации ⓘ

Создайте и обучите свою первую нейросеть с нуля. Погрузитесь в мир Data Science!

Скидка 63%

Год английского языка за 0 р.

Узнать больше

Модуль iccream в Python

При отладке проекта в Python

при отладке проекта или создании его прототипа, вряд ли кто сразу бросается настраивать [модуль](#) инспектировании переменных. 85% разработчиков используют для этих целей [функцию print\(\)](#).

используя сторонний [модуль iccream](#), или сокращенно ic. Этот модуль похож на функцию специальной настройки, подобной модулю logging.

работает в терминале Python и в случае попытки выполнить iccream.ic() будет ругаться, что в исходном коде для анализа. Не запускайте примеры в терминале, они будут работать, представленных ниже примеров.

iccream:

переменных, так и их значения;
();

- легко отключить вывод отладочной информации;
- может включать контекст программы: имя файла, родительскую функцию и номер строки;
- модуль хорошо протестирован и поддерживает Python2, Python3, PyPy2 и PyPy3;

Установка модуля iccream в виртуальное окружение:

```
# создаем виртуальное окружение, если нет
$ python3 -m venv .venv --prompt VirtualEnv
# активируем виртуальное окружение
$ source .venv/bin/activate
# ставим модуль iccream
(VirtualEnv):~$ python -m pip install -U iccream
```

Содержание.

- [Инспектирование/проверка переменных](#).
- [Инспектирование порядка выполнения кода программы](#).
- [Другие варианты использования модуля iccream](#).
- [Включение/выключение вывода модуля iccream](#).
- [Доступ к модулю iccream из импортируемых файлов](#).
- [Настройка модуля iccream](#).

Инспектирование/проверка переменных.

При отладке программы иногда необходимо вывести значения некоторых переменных или выражений, при этом функцию print() используют следующим образом:

```
print(foo('123'))
# или
print("foo('123')", foo('123'))
```

В данной ситуации очень поможет функцию ic() модуля iccream. С помощью аргументов, ic() проверяет себя и выводит как свои собственные аргументы, так и значения этих аргументов.


```
from iccream import ic
def foo(1):
```

Вверх


```
return i + 333
```

```
ic(foo(123))  
# ic| foo(123): 456
```



РЕКЛАМА · 16+

 **GeekBrains**

Узнайте все о разработке
искусственного интеллекта!

 gb.ru

**Курс: искусственный
интеллект с 0!
Осенняя
распродажа!**

5,0  Рейтинг организации 

Создайте и обучите свою
первую нейросеть с нуля.
Погрузитесь в мир Data Science!

Скидка 63% >

Год английского языка за 0 р. >

Узнать больше

```
if expression:  
    ic()  
    second()  
else:  
    ic()  
    third()  
  
# ic| test.py:4 in foo()  
# ic| test.py:11 in foo()
```

Другие варианты использования модуля iccream.

Функция `ic()` возвращает переданные ей аргументы, следовательно ее можно легко вставить в уже существующий код.

```
from iccream import ic  
a = 6  
  
def half(i):  
    return i / 2  
  
b = half(ic(a))  
# ic| a: 6  
ic(b)  
# ic| b: 3
```

Команда `ic.format(*args)` похож на `ic()`, но вывод возвращается в виде строки, а не записывается в `stderr`.

```
from iccream import ic  
  
s = 'sup'  
out = ic.format(s)  
  
print(out)  
# ic| s: 'sup'
```

Вверх


https://docs-python.ru/packages/modul-iccream-python/

2/4


Включение/выключение вывода модуля iccream.

Вывод функции ic() можно полностью отключить, а позже снова включить с помощью ic.disable() и ic.enable() соответственно. При отключенном выводе, функция ic(), конечно же, продолжит возвращать переданные ей аргументы.



РЕКЛАМА · 16+

 **GeekBrains**

Узнайте все о разработке искусственного интеллекта!

 gb.ru

Курс: искусственный интеллект с 0! Осенняя распродажа!

5,0  Рейтинг организации 

Создадите и обучите свою первую нейросеть с нуля. Погрузитесь в мир Data Science!

Скидка 63% >

Год английского языка за 0 р. >

Узнать больше

Исключение iccream из импортируемых файлов.

В каждом файле программы без необходимости импорта его в каждый файл, то нужно вызвать ic.disable(). Например:

```
from iccream import ic
ic.disable()
```

Если iccream импортируется в a.py, просто будем вызывать функцию ic():

```
# импортируемый файл `b.py`
def foo():
    x = 3
    ic(x)
```

Функция install() добавляет ic() во встроенный [модуль builtins](#), который используется всеми файлами, импортированными интерпретатором. Точно так же ic() позже можно удалить функцией uninstall().

Функция ic() также может быть импортирована способом, который не завершается сбоем, если модуль iccream не установлен, например, на боевых серверах. С этой целью можно использовать следующий фрагмент импорта:

```
try:
    from iccream import ic
except ImportError:
    # откат, если модуль `iccream` не установлен.
    ic = lambda *a: None if not a else (a[0] if len(a) == 1 else a)
```

Настройка модуля iccream.

Для настройки вывода iccream можно использовать ic.configureOutput(). Вот ее синтаксис:

```
ic.configureOutput(prefix='ic |',
                  outputFunction=sys.stderr,
                  argToStringFunction=pprint.pformat,
                  includeContext=False)
```

Она поможет изменить следующие настройки по умолчанию:

- префикс вывода (по умолчанию ic |),
- функцию, отвечающую за вывод (по умолчанию sys.stderr),
- способ сериализации аргументов в строки,
- включить контекст вызова ic() (имя файла, номер строки и родительскую функцию) в ее вывод.

Аргумент prefix.

Аргумент prefix отвечает за изменение префикса, при выводе отладочной информации.

Вверх

```
from icecream import ic
ic.configureOutput(prefix='hello => ')
ic('world')
# hello => 'world'
```



gb.ru

Курс: искусственный интеллект с 0!
Осенняя распродажа!

5,0 ★ Рейтинг организации ⓘ

Создадите и обучите свою первую нейросеть с нуля. Погрузитесь в мир Data Science!

Скидка 63% >

Год английского языка за 0 р. >

Узнать больше

может быть функцией.

|'

.timestamp)

он.

быть функцией, в которую будет передан вывод ic(). По умолчанию используется [sys.stderr](#).

on=warn)

Аргумент argToStringFunction.

Аргумент `outputFunction`, должен быть функцией, которая будет вызвана со значениями аргументов, передаваемых `ic()` и которые должны быть сериализованы в отображаемые строки. По умолчанию используется [функция `pprint.pformat\(\)`](#), но ее можно изменить, например, для обработки нестандартных типов данных индивидуальным образом.

```
from icecream import ic

def toString(obj):
    if isinstance(obj, str):
        return f'[{!string "{obj}" with length {len(obj)}!]'
    return repr(obj)

ic.configureOutput(argToStringFunction=toString)
ic(7, 'hello')
# ic| 7, [{!string "hello" with length 5!}]
```

Аргумент includeContext.

Если аргумент `IncludeContext=True`, то в вывод добавляется: имя файла, родительская функция и номер строки, в которой вызвана `ic()`.

```
from icecream import ic
ic.configureOutput(includeContext=True)

def foo():
    ic('str')

foo()
# ic| test.py:5 in foo()- 'str'
```