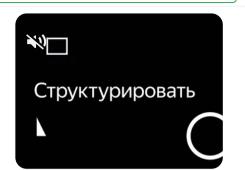
Сообщить об ошибке.

ХОЧУ ПОМОЧЬ ПРОЕКТУ

Jinja2 - движок шаблонов для Python



n practicum.yandex.ru

РЕКЛАМА • 16+

Курс «Специалист по Data Science». Начните с нуля.

Обучаем специалистов в Data Science с нуля. 20 часов практики – бесплатно.

Узнать больше

для своего бизнеса на hh.ru ком шаб Пре Найти Реклама. ООО «Хэдхантер».

Подробнее на сайте hh.ru

<u>3.</u> / Jinja2 - движок шаблонов для Python

ий и удобный язык шаблонов для Python, созданный по образцу шаблонов Django. Он быстр, т.к. проко используется и безопасен благодаря дополнительной среде выполнения изолированных

<u>nja2</u>:

ранирования HTML для предотвращения XSS.

ддержка макросов.

оптимального кода Python (можно отключить при отладке).

исключений точно указывают на неправильную строку в шаблоне.

много встроенный фильтров.

етодов стандартных типов Python в шаблонах.

й Python в шаблонах.

а2 в виртуальное окружение:

```
# создаем виртуальное окружение, если нет
$ python3 -m venv .venv --prompt VirtualEnv
# активируем виртуальное окружение
$ source .venv/bin/activate
# ставим модуль Jinja2
(VirtualEnv):~$ python -m pip install -U Jinja2
```

Инициализация движка шаблонов Jinja2 в Python.

Модуль Jinja использует центральный объект, называемый шаблоном <u>jinja2.Environment()</u>. Экземпляры этого класса используются для хранения конфигурации и глобальных объектов, а также для <u>загрузки шаблонов</u> из файловой системы или других мест. Даже если создавать шаблоны из строк с помощью конструктора класса <u>jinja2.Template()</u>, среда Environment создается автоматически, только она будет совместно используемая.

Большинство приложений создают один объект Environment при инициализации приложения и используют его для загрузки шаблонов. Однако в некоторых случаях полезно иметь несколько сред рядом, если используются разные конфигурации.

Самый простой способ настроить Jinja для загрузки шаблонов для приложения выглядит примерно так:

```
from jinja2 import Environment, PackageLoader, select_autoescape
env = Environment(
    loader=PackageLoader('yourapplication', 'templates'),
    autoescape=select_autoescape(['html', 'xml'])
)
```

Это создаст шаблонную среду с настройками по умолчанию и загрузчиком loader, который будет искать шаблоны в папке шаблонов templates внутри пакета python yourapplication. <u>Доступны разные загрузчики</u>, также можно написать свой собственный, если необходимо загружать шаблоны из базы данных или других ресурсов. Код примера, так же, определяет автоматическое экранирование файлов с расширениями .html и .xml.

Чтобы загрузить шаблон из среды env, которая определены в примере выше, нужно просто вызвать метод env.get_template(), который затем возвращает загруженный шаблон <u>Template</u>:

```
template = env.get_template('mytemplate.html')
```

Чт Вверх разить его с некоторыми переменными, просто вызовите метод Template.render():

Вэ

Сам

рек

```
print(template.render(the='variables', go='here'))
```

Использование <u>загрузчика шаблонов</u> вместо передачи обычных <u>строк</u> в Template или Environment.from_string() имеет несколько пре<u>имуществ. Помимо того, что</u> загрузчик намного проще в использовании, он также позволяет <u>наследование шаблонов</u>.

```
При hh

В 6
Рек Ищите
при работников
Для своего
при бизнеса
фил на hh.ru
точ
выр
```

матическое экранирование будет включено по умолчанию из соображений безопасности.

<u>втоматическое экранирование</u>, а не полагаться на значение по умолчанию.

в шаблонах.

<u>1Ю</u>.

вания Python. Допустимые идентификаторы могут быть любой комбинацией символов Юникода,

остранствах имен и имеют слегка измененный синтаксис идентификатора. Фильтры могут содержать и. Например, вполне допустимо добавить метод в фильтр и вызвать его как .unicode. Регулярное рикаторов фильтров: [a-zA-Z_][a-zA-Z0-9_]*(\.[a-zA-Z_][a-zA-Z0-9_]*)*.

ние движка шаблонов Jinja2.

введение в Python API для шаблонов Jinja.

лаблон и отрендерить его - использовать класс <u>jinja2.Template()</u>. Такой способ работы не эгружаются не из строк, а из файловой системы или другого источника данных:

```
te
o {{ name }}!')
n Doe')

# Подробнее на сайте hh.ru

>>> content = {'a': 5, 'b': 2}
>>> tpl = 'Сумма чисел {{ a }} и {{ b }} равна {{ a + b }}'
>>> Template(tpl).render(content)

# 'Сумма чисел 5 и 2 равна 7'
```

Пример разбора шаблона с циклом.

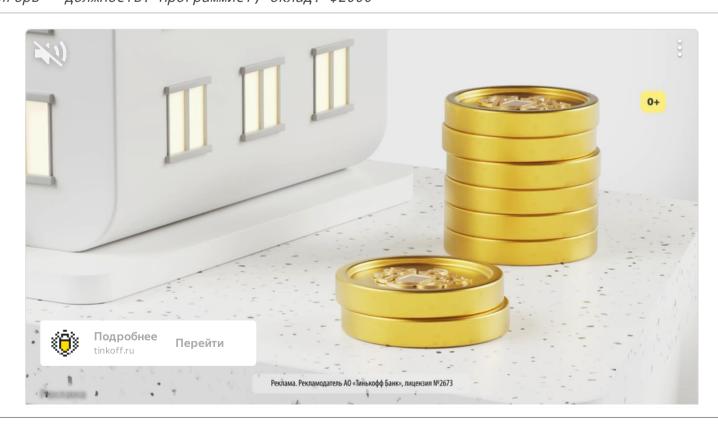
```
>>> import jinja2
# шаблон с циклом
>>> tpl = """{{ title }}
... {{ '-' * title|length }}
  . {% for n, user in enumerate(users, 1) %}
... {{ n }}. {{ user.name }} - должность: {{ user.status }}, оклад: ${{ user.salary }}
... {% endfor %}
# собираем данные для шаблона
>>> content = {}
>>> content['title'] = 'Итерация по пользователям'
>>> content['users'] = []
>>> content['users'].append({'name': 'Маша', 'status': 'Менеджер', 'salary': 1500})
>>> content['users'].append({'name': 'Света', 'status': 'Дизайнер', 'salary': 1000})
>>> content['users'].append({'name': 'Игорь', 'status': 'Программист', 'salary': 2000})
# В словаре передаем в шаблон функцию Python
>>> content['enumerate'] = enumerate
# Смотрим, что получилось
>>> print(jinja2.Template(tpl, trim_blocks=True).render(content))
# Итерация по пользователям
# -----
# 1. Маша - должность: Менеджер, оклад: $1500
# 2. Света - должность: Дизайнер, оклад: $1000
# 3. Игорь - должность: Программист, оклад: $2000
```

Загрузка шаблонов из файловой системы.

Сохраним шаблон из предыдущего примера в директорию ~/temp/main.txt.

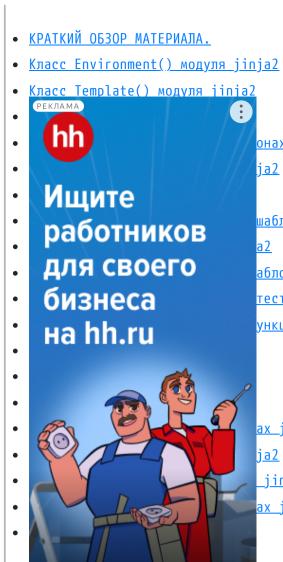
```
{# Вверх }} {# Добавим условие #}
```

```
12.09.2023, 15:00
                                                         Модуль jinja2 в Python, язык шаблонов
 {% if title %}
 {# Если существует переменная `title`, то будем ее подчеркивать #}
 {{ '-' * title|length }}
 {% endif %}
 {#
                               sers, 1) %}
                               лжность: {{ user.status }}, оклад: ${{ user.salary }}
 {%
     Ищите
                               ия работает с сохраненным шаблоном main.txt. Для понимания, что происходит, код снабжен
под
     работников
     для своего
     бизнеса
                                шаблонов из файловой системы
                               охраненный шаблон 'main.txt')
     на hh.ru
 10
                               der('temp')
 #
 #
                               er=loader, trim_blocks=True)
 en
 #
 COI
                               по пользователям'
 COI
 COI
                               e': 'Маша', 'status': 'Менеджер', 'salary': 1500})
 COI
                               e': 'Света', 'status': 'Дизайнер', 'salary': 1000})
 COI
                               e': 'Игорь', 'status': 'Программист', 'salary': 2000})
 COI
               Найти
                                функцию Python
                               ate
 COI
      Реклама. ООО «Хэдхантер»
      Подробнее на сайте hh.ru
 tpl = env.get_template('main.txt')
 # рендерим шаблон в переменную `result`
 result = tpl.render(content)
 # Сохраним получившийся текст
 with open('result.txt', 'w') as fp:
     fp.write(result)
 # Прочитаем записанный файл
 with open('result.txt', 'r') as fp:
     print(fp.read())
 # Итерация по пользователям
 # -----
 # 1. Маша - должность: Менеджер, оклад: $1500
 # 2. Света - должность: Дизайнер, оклад: $1000
 # 3. Игорь - должность: Программист, оклад: $2000
```



Содержание раздела:

Вверх



DOCS-Python.ru™, 2023 г.

Подробнее на сайте hh.ru

nja2
oнax Jinja
ja2

шаблонов модулем jinja2
a2
aблонов jinja2
тесты для jinja2
ункции модуля jinja2

ax_jinja2 ja2 _jinja2 ax_jinja2

(Внимание! При копировании материала ссылка на источник обязательна)

@docs_python_ru

Вверх