Microsoft Partner
Silver Learning

Python Essential

Генератори

ITVDN
IT VIDEO DEVELOPERS NETWORK

Після уроку обов'язково

Повторіть цей урок у відео форматі на ITVDN.com

Перевірте, як Ви засвоїли даний матеріал на TestProvider.com

Тема

Генератори

Генератори

Функція-генератор (generator function) – це функція, яка повертає спеціальний **ітератор генератора (generator iterator)** (також **об'єкт-генератор - generator object**). Вона характеризується наявністю ключового слова yield всередині функції.

- Термін **генератор (generator)**, залежно від контексту, може означати або функцію-генератор, або ітератор генератора (найчастіше, останнє).
- Методи __*iter*__ та __*next*__ у генераторів створюються автоматично.
- yield заморожує стан функції-генератора та повертає поточне значення. Після наступного виклику __*next*__() функція-генератор продовжує своє виконання з того місця, де вона була призупинена.
- Коли виконання функції-генератора завершується (за допомогою ключового слова return або досягнення кінця функції), виникає виняток StopIteration.

Вирази-генератори

Деякі прості генератори можуть бути записані у вигляді виразу. Вони виглядають як вираз, що містить деякі змінні, після якого одне або кілька ключових слів for, котрі задають, які значення повинні приймати дані змінні (синтаксис відповідає заголовку циклу for), і нуль або кілька умов, що фільтрують значення, які генеруються (синтаксис відповідає заголовку оператора if). Такі вирази називаються **виразами-генераторами (generator expressions)**.

```
function(x, y) for x in range(10) for y in range(5) if x != y
```

Підгенератори

У Python 3 є так звані **підгенератори (subgenerators)**. Якщо у функції-генераторі зустрічається пара ключових слів yield from, після яких слідує об'єкт-генератор, то даний генератор делегує доступ до підгенератора, поки він не завершиться (не закінчаться його значення), після чого продовжує своє виконання.

```
def generator():
    ...
    yield from subgenerator()
    ...
```

Yield-вирази

- Насправді yield є виразом. Воно може набувати значень, які відправляються в генератор. Якщо в генератор не відправляються значення, результат цього виразу дорівнює None.
- yield from також є виразом. Його результатом є те значення, яке підгенератор повертає у виключенні StopIteration (для цього значення повертається за допомогою ключового слова return).

```
def generator():
    ...
    data = yield
    ...
```

Методи генераторів

Метод	Опис
__next__()	Починає або продовжує виконання функції-генератора. Результат поточного yield-виразу дорівнюватиме None. Виконання потім продовжується до наступного yield-виразу, який видає значення туди, де був викликаний __next__. Якщо генератор завершується без видачі значення за допомогою yield, виникає виняток StopIteration. Метод зазвичай викликається неявно, тобто циклом for або вбудованою функцією next().
send(value)	Продовжує виконання та відправляє значення до функції-генератора. Аргумент value стає значенням поточного yield-виразу. Повертає видане значення. Якщо send() використовується для запуску генератора, то єдиним допустимим значенням є None, так як ще не було виконано жодного yield-виразу, якому можна привласнити це значення.
throw(type[, value[, traceback]])	Викидає виняток типу type у місці, де було призупинено генератор, і повертає наступне значення генератора (або викидає StopIteration).
close()	Викидає виняток GeneratorExit у місці, де було призупинено генератор. Якщо генератор повертає чергове значення, викидається виняток RuntimeError.

Співпрограми

- **Співпрограма** (англ. **coroutine**) – компонент програми, що узагальнює поняття підпрограми, який додатково підтримує безліч вхідних точок (а не одну, як підпрограма) та зупинку і продовження виконання із збереженням певного положення.
- Розширені можливості генераторів у Python (вирази yield та yield from, відправка значень у генератори) використовуються для реалізації співпрограм.
- Співпрограми корисні для реалізації асинхронних неблокуючих операцій і кооперативної багатозадачності в одному потоці без використання функцій зворотного виклику (callback-функцій) і написання асинхронного коду в синхронному стилі.
- Python 3.5 включає підтримку співпрограм на рівні мови. Для цього використовуються ключові слова async та await.

Дивіться наші уроки у відео форматі

ITVDN.com

Перегляньте цей урок у відео форматі на освітньому порталі ITVDN.com
для закріплення пройденого матеріалу.

Усі курси записані сертифікованими тренерами, які працюють у навчальному центрі CyberBionic Systematics

ITVDN
IT VIDEO DEVELOPERS NETWORK

Перевірка знань

TestProvider.com

TestProvider – це online сервіс перевірки знань з інформаційних технологій. З його допомогою Ви можете оцінити Ваш рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT-спеціаліста.

Після кожного уроку проходьте тестування для перевірки знань на TestProvider.com

Успішне проходження фінального тестування дозволить Вам отримати відповідний Сертифікат.

TestProvider

Q&A

Інформаційний відеосервіс для розробників програмного забезпечення