

# المشروع في Python حزم تحليل

ويقوم التكرار، ويزيل java. ملفات في المستخدمة حزم جميع ويجد تكراري، بشكل مجلدًا يحلل الذي Python نص هنا  
قد التي المشاريع بمعالجة ويقوم فقط java. ملفات على النص يركز المشروع. في المستخدمة الفرعية الحزم جميع بقائمة  
الفرعية. المجلدات جميع عبور خلال من وحدات على تحتوي

النص Python:

```
import os
import sys

def find_java_files(root_dir):
    """
    Find all Java files in the directory tree rooted at root_dir.

    Args:
        root_dir (str): The root directory to search.

    Yields:
        str: The path to each Java file.

    """
    for dirpath, dirnames, filenames in os.walk(root_dir):
        for filename in filenames:
            if filename.endswith('.java'):
                yield os.path.join(dirpath, filename)

def extract_package(import_statement):
    """
    Extract the package name from an import statement.

    Args:
        import_statement (str): The import statement to parse.

    Returns:
        str: The package name.

    """
    # 'import' 'static'
    parts = import_statement.split()
```

```

if parts[0] == 'import':
    parts = parts[1:]
if parts[0] == 'static':
    parts = parts[1:]

#           ';'
import_path = ' '.join(parts).strip(';').strip()

#           '.'
identifiers = import_path.split('.')

#           '*'
for i, ident in enumerate(identifiers):
    if ident == '*' or (ident and ident[0].isupper()):
        package_parts = identifiers[:i]
        break
else:
    package_parts = []

package = '.'.join(package_parts)
return package

if __name__ == '__main__':
    #
    if len(sys.argv) < 2:
        print(" python script.py <root_directory>")
        sys.exit(1)

    root_dir = sys.argv[1]
    packages = set()

    #           .java
    for java_file in find_java_files(root_dir):
        try:
            with open(java_file, 'r', encoding='utf-8') as f:
                for line in f:
                    line = line.strip()
                    if line.startswith('import'):
                        package = extract_package(line)
                        if package: #

```

```

        packages.add(package)

    except Exception as e:

        print(f"      :      }java_file}: {e}")

        continue

#

for package in sorted(packages):

    print(package)

```

النص: استخدام لفي في

1. analyze\_java\_packages.py مثل ملف، في النص احفظ.
2. :لل مشروع الجذر المجلد مسار تقديم الأوامر، السطر من النص بتشغيل ق م

```
python analyze_java_packages.py /path/to/your/java/project
```

3. .java . ملفات في استيراده تم التي الفريدة الحزم أسماء من مرتبة قائمة بإخراج النص سيقوم.

النص: يفعله ما

□ □ .java : ملفات يحد

- الفرعية. ومجلداته للمجلد التكراري للعبور os.walk() يستخدم
- .java . بـ تنتهى التي الملفات جميع يحدد

□ □ الحزم: أسماء يستخرج

- سطر. بعد سطرًا .java . ملف لكل يحلل
- import. بـ تبدأ التي السطور يحدد
- مع الاستيراد، بيانات من الحزم أسماء يستخرج
  - import java.util.List;□. مثل العادية الاستيرادات
  - import java.util.\*;□. مثل البرية الاستيرادات
  - import static java.util.Collections.sort;□. مثل الثابتة الاستيرادات
- □ □ □ :تقاليد على يعتمد استدلال يستخدم
  - java.util.□ مثل صغيرة بأحرف عادة الحزم أسماء
  - List, Collections□. مثل كبيرة بأحرف عادة تبدأ الفئات أسماء
  - \* . بـ تنتهى البرية الاستيرادات

□ □ التكرار: يزيل

- تلقائيًا. التكرار لإزالة مجموعة في الحزم أسماء جميع يجمع

□ □ النتائج: يخرج

- جدي. سطر في حزمة اسم لكل ويطبوع مرتبة قائمة إلى المجموعة يحول

## والتقنيات: الافتراضات

### الاستيراد: بيانات

الأسلوب. أو التعليلات في مدفونة تكون ولا من فصله سطور في الاستيراد بيانات أن يفترض

البرية. الاستيرادات ذلك في بما الثابتة، والاستيرادات العادية الاستيرادات من كل يعالج

### التسمية: تقاليد

كبيرة. بأحرف يبدأ الفئة واسم صغيرة، بأحرف الحزم أسماء تكون الذي التقليل على يعتد

الممارسة. في نادر هذا ولكن كبيرة، بأحرف الحزم أسماء مثل القياسية غير التسمية صحيح بشكل يعالج لا قد

### الملف: تشفير

الأم. لزم إذا encoding بتعديل ق م 8. في مشفرة java. ملفات أن يفترض

### الأخطاء: معالجة

الاستمرار. وتحذير طباعة للقراءة، القابلة غير للملفات الأساسية الأخطاء معالجة يتضمن

## المثال:

التالي: java. الملفات على يحتوي 1. مشروعك كان إذا

### 1.:

```
import java.util.List;
import java.util.*;
import java.lang.String;
```

### 2.:

```
import java.util.Collections;
import static java.util.Collections.sort;
import com.example.myapp.*;
```

سخرج: النص تشغيل عند

```
com.example.myapp
java.lang
java.util
```

أبجدياً. مرتبة المشروع، في المستوردة الفريدة الحزم جميع يدرج هذا

## الملاحظات:

- يستخدمها التي والداخلية الخارجية الحزم تمثّل والتي الاستيراد، بيانات عبر المستوردة الحزم على النص يركّز □ المشروع.
- التي الحزم تعرّف هذه لأن، □ `package com.mycompany.myapp;` مثّل نفسها `java`. ملفات في الحزم بيانات يشمل ل □ تستخدمها. التي الحزم وليس الملفات، إليها تنتمي
- جميع معالجة لضمان الفرعية المجمّلات لجميع تكراري تحليّل بعمل النص يقوم وحدات، على يحتوي قد المشروع لأن نظرًا □ `java`. ملفات