

Modules & Packages

الوحدات و الرُزم Modules & Packages

البرمجة المعتمدة على الوحدات Modular Programming

طريقة البرمجة المعتمدة على الوحدات تشير الى الفكرة التي تقول بأن تجزيء و تقسيم المشروع الى أجزاء (وحدات) اصغر ، تسهل التحكم بالمشروع ككل . بالتالي يمكن ترتيب المشروع الى اجزاء اصغر ، مرتبة حسب المهمة .

مزايا البرمجة المعتمدة على الوحدات Modular Programming Advantages

- **السهولة**
التركيز على وحدة من البرنامج تجعل البرمجة أسهل .
- **قابلية الصيانة و المحافظة على الكود**
تقسيم المشروع الى وحدات و اجزاء اصغر تسمح لك بالتعديل على الكود من دون التأثير على باقي الأجزاء . بالتالي تسمح ايضا للفريق بالتعاون.
- **اعادة الإستخدام**
بسبب ان المشروع مقسم الى وحدات اصغر ، بالتالي بالإمكان تكرار استخدام نفس الوحدة في أماكن عدة من المشروع .
- **تعيين المدى**
تقسيم الكود في مشروعك الى وحدات و رُزم يساعدك على تقادي اصطدام المسميات .

الوحدة في البايثون

أي ملف بايثون يُعتبر وحدة Module ، بالتالي بالإمكان استيراده للإستخدام في مختلف أجزاء مشروعك .

لإستيراد وحدة Module

نستخدم الكلمة المفتاحية import متبوعة بإسم الوحدة (الموديول)

```
import module_name
```

إستيراد وحدة مع اسم بديل alias

يفيدنا هذا في اختصار بعض المسارات الطويل للوحدات . او استخدام كلمة مختصرة للتعبير عن ملف معين .

```
import orders as ord
```

لإستخدام متغير أو دالة أو عنصر من الوحدة المستوردة ، نذكر اسم الوحدة بعد الإستيراد من ثم نضيف نقطة و بعدها نستدعي المطلوب

```
import orders as ord
print(ord.some_list)
```

إستيراد عناصر ، متغيرات ، او دوال على حدى من وحدة معينة
بعد استيراد العناصر على انفراد ، يمكن استخدامها مباشرة كما لو تم تعريفها في نفس الملف

```
from users import users_list, create_user
```

بالإمكان استيراد جميع العناصر من الوحدة بإستخدام علامة النجمة *

```
from users import *
```

الرزمة في البايثون Package

الرُزم عبارة عن المجلدات داخل مشروع البايثون . فأى مجلد داخل مشروع البايثون سيتم اعتباره ك رُزمة package . هذا يسمح لنا ان نرتب مشروعنا الى وحدات و رُزم . فيتم تقسيم الوحدات ذات الصلة في رُزم (مجلدات) خاصة بها . مما يجعل المشروع مُرتب بشكل هرمي ، و الوصول الى اي جزء من المشروع أسهل و أسرع .

ايضا نتيج لنا لغة البايثون أن نظيف ملف تهيئة ، بحيث يتم تنفيذ بعض العمليات لتهيئة اي وظائف مطلوبة من هذه الرُزمة . هذا الخاصية اختيارية و يتم استخدامها على حسب احتياج المشروع.

استيراد الرزم packages import

بعد انشاء رزمة جديدة (مجلد جديد) في مشروعنا و بداخله بعض الوحدات (الملفات) ، يمكننا استيراد الرزمة كاملة في مختلف أجزاء المشروع .

لإستيراد الوحدات من داخل رزمة معينة ، نستخدم from ... import

```
#to import indivisual modules from the package
from delivery import create, info
```

تهيئة الرزم Package Initialization

بالإمكان تهيئة الرزمة و تنفيذ بعض الوظائف او العمليات المطلوبة من خلال اضافة ملف داخل المجلد (الرزمة) ، يجب أن يكون اسم الملف هذا `__init__.py` . سيتم استدعاء و تنفيذ ما بداخل هذا الملف عند عمل استيراد للرزمة . هذا الملف اختياري . و يتم استخدامه عادة لتهيئة بعض البيانات أو بعض الروابط التي تحتاجها الرزمة .

مكان البحث عن مسار الرزم و الوحدات Module Search Path

عندما يقوم المترجم python interpreter بتنفيذ امر الإستيراد سيقوم بالبحث عن الملفات و المجلدات داخل المسارات التالية كبداية :

- مسار المجلد الذي يتم تنفيذ البرنامج بداخله .
- مسار المجلدات المضمنة في متغير البيئة PYTHONPATH (في حال كان معين، حيث انه يعتمد على النظام)
- مسار مجموعة من المجلدات التي قد يتم تعيينها وقت تنصيب البايثون .

للإطلاع على قائمة بالمسارات التي سيتم البحث بداخلها عن الوحدات و الرزمات ، نفذ التالي :

```
import sys
print(sys.path)
```

وحدات البايثون المضمنة الاساسية

للإطلاع على الوحدات المضمنة مع البايثون ،يمكنك زيارة Python Module Index ، حيث سيتم استعراض جميع الوحدات مع التوثيق و المعلومات الوافية عن كل منها و وظائفها .

قم بزيارة الرابط التالي : <https://docs.python.org/3/py-modindex.html>