

Web scraping:

- First
- create youtube_statistics.py
- In this file extract data using YTstats class and generate a json file with all the data extracted.
- create main.py
- In main import youtube_statistics.py
- Add API key and channel id .
- using the first file data corresponding to the key given will be retrieved and saved to json file.
- end

Function lambda:

-تسمى ب anonymous function or lambda expression

-دالة من دون اسم

- تعرف باسمها lambda وليس def

- ترجع البيانات من دالة أخرى

- تستخدم لدوال بسيطة وليس معقدة

-عبارة عن سطر برمجي واحد وليس block

- syntax of lambda :

١ - الكلمة الرئيسية lambda

٢ - arguments غير محاطة بأقواس

٣ - expression (الجسم الوظيفي)

٤ - ex.

Lambda arguments : expression

sum = Lambda x : 3+x

print(sum(5))

outputs =8

خوارزمية البحث الثنائي Binary search

الهدف : البحث عن قيمة المفتاح key داخل المصفوفة X.

الفكرة : يبدأ البحث من العنصر الذي يقع في وسط المصفوفة، و في كل مرة نقارنه مع المفتاح Key، إذا كانت القيمتان متساويتان، فهذا يدل على أنه تم إيجاد قيمة المفتاح في المصفوفة، أما إذا كانت القيمتان مختلفتان ستقوم الخوارزمية بإجراء فحص جديد، إذا كانت قيمة المفتاح أصغر من قيمة العنصر الأوسط سيتم البحث في الجزء الأيسر من المصفوفة و في الحالة المعاكسة سيتم البحث في الجزء الأيمن من المصفوفة، و هكذا ، حتى نحصل على مصفوفة تتكون من خانة واحدة قيمتها مساوية للمفتاح أو مختلفة عنه.

النتيجة : إذا كانت X تحتوي على Key فسنحصل على رقم الخانة التي يوجد بها الأخير و إلا فالقيمة المُعادة ستكون - ١.

الإيجابيات: الـ Binary search تُنصف (تقلص للنصف) عدد عناصر المصفوفة في كل تكرار، لذا تستغرق عملية البحث وقت قليل جداً. تنتمي هذه الخوارزمية إلى عائلة فرق تسد.

السلبيات : خوارزمية البحث الثنائي أكثر تعقيداً من خوارزمية البحث الخطي التقليدية، كما أن الأولى تشترط الترتيب عند البحث.

SOLID Principle

ما هو ال SOLID :

هي كلمة مختصرة لخمس مبادئ اساسية بتطبيق هذه المبادئ تجعل الكود قابل للتوسع والتحديث والاختبار

حرف ال S - مبدأ المسؤولية الواحدة Single Responsibility Principle

حرف ال O - مبدأ الفتح والاعلاق Open-Close Principle

حرف ال L - مبدأ الاستبدال Liskov Substitution Principle

حرف ال I - مبدأ فصل الواجهات Interface segregation Principle

حرف ال D - مبدأ انعكاس التبعية Dependency Inversion Principle

الفرق بين ال Design Principle & Design Pattern

أولا Design Principle :

هي مجموعة مبادئ وقواعد توضح لنا شكل الكود علشان الكود في النهاية يكون

Testable, Extensible, Maintainable

هذه المبادئ بتجنبنا من العادة السيئة للتركيز على الوصول للحل دون الأخذ في اعتبار جودة الحل

ثانيا Design Pattern :

هي حلول عملية وفعلية تم بناؤها علشان تحل مشكله برمجية معينة

الغرض الأساسي منها تقديم حلول برمجية لمشاكل شائعة

- Coupling الارتباط

لما يكون في class يستعمل object من class آخر بنقول أنهم مترابطين
ممكن الارتباط

يكون ضعيفا : Loose Coupling

يكون قويا : Tight Coupling

كلما كان الارتباط بين classes وبعضها قوي كلما كان الكود صعب الصيانة
والتعديل

ولو الارتباط ضعيف سيكون أفضل لأننا بنقل الاعتمادية

- المبدأ الأول Single Responsibility

معناه أن كل Function أو Class أو Model له مسؤولية واحدة فقط يقوم بها وله سبب واحد فقط في التغيير

مثال Class واحد يقوم بقراءة البيانات وحفظ البيانات في قاعدة البيانات والتحقق من سلامة البيانات يتم تقسيمه إلى Class يقوم بقراءة البيانات و Class يقوم بحفظ البيانات في قاعدة البيانات و Class يتحقق من سلامة البيانات .

- المبدأ الثاني Open-Close Principle

معناه أنه يسمح بالإضافة وغير مسموح بالتعديل

مسموح بالتوسع والاضافه <Open Extended

غير مسموح بالتعديل علي الكود الحالي > Modified Close

- المبدأ الثالث Liskov Substitution Principle

عندنا Class للأب Class للأبن نستطيع نبادل الأدوار بينهم من غير مشاكل في البرنامج معناها لو أن الأب غير موجود ممكن الأبن يقوم بنفس الدور.

- المبدأ الرابع Interface Segregation Principle

Class اضع فيه دوال Non Implemented \

مثال: انا لما يكون عندي Interface لما اي Class مثلا يورث منه لازم يطبق كل الدوال الذي داخله

وإذا ال Class لا يريد تطبيق كل الدوال الذي فيه في الحالة هذه نستخدم مبدأ ISP وهي بنقوم بوضع الدوال الذي محتاجها ال Class في Interface لوحدة

- المبدأ الخامس Dependency Inversion Principle

معناها ان ال High Level Module لا يعتمد على ال Low Level Module لازم يعتمدوا الاثنين على ال Abstract

لا ينفع ال Abstraction يعتمد على تفاصيل ال Concrete و العكس صحيح

ما الفرق بين ال Low Level Module و ال High Level Module

High Level Module - هو ال Module الذي بيعتمد على Module آخر

Low Level Module - هو ال Module الذي لا بيعتمد على Module آخر أو بيعتمد عليه لكن في أضيق الحدود