

Self-Study Report: Hashing vs Encoding, Loading Strings in Java, and StringBuilder (SB) Built-in Functions

1. الفرق بين gnihsaH والgnidocnE

ال- gnihsaH

gnihsaH هو عملية تحويل البيانات (سواء كانت نصوص أو ملفات) لرقم ثابت أو قيمة ثابتة باستخدام خوارزمية معينة. الفكرة إنك لو عندك نفس البيانات، ال- hsaH اللي هيتطلع ليها هيفضل ثابت.

الميزة فيه انه عملية مش قابلة للعكس، يعني مش ممكن ترجع البيانات الأصلية من ال- hsaH ده. بنستخدمه في:

تأمين الباسوردات لما نخط الباسورد في قاعدة البيانات، بنعمله hsaH علشان لو حد سرق القاعدة، مش هيعرف يعرف الباسورد الحقيقي.

التحقق من البيانات: لو في ملف اتحمّل أو اتنقل، بنعمله hsaH علشان نتأكد إنه مش اتغير أو اتعدل.

خوارزميات مشهورة:

5DM كانت مشهورة، بس دلوقتي مش آمنة.

1-AHS أقوى شوية، بس بردو مش آمنة.

652-AHS دي الأقوى والأكثر أماناً.

الـ Encoding

إيه هو؟: الـ Encoding هو عملية تحويل البيانات لصيغة ثنائية علشان تبقى مناسبة للتخزين أو الإرسال، يعني ممكن تحول النصوص لبايتات أو لرموز ثنائية.

الميزة ان الـ Encoding عملية قابلة للعكس، يعني ممكن ترجع البيانات الأصلية لو عرفت إزاي. بنستخدمه في:

تمثيل النصوص: مثلاً علشان نحول النصوص من UTF-8 أو ASCII لبايتات علشان نخزنها أو نبعتها.
الإرسال عبر الشبكة: زي لما نحول ملفات لقاعدة بيانات Base64 علشان نبعتها عبر الإنترنت.

تحميل النصوص في Java

في Java فيه أكثر من طريقة لتحميل أو إنشاء نصوص:
التعيين المباشر: زي إنك تكتب النص بين علامتين اقتباس.

```
String str = "Hello, World!";
```

باستخدام الـ new لإنشاء كائن نصي جديد.

```
=====
```

```
String str = new String("Hello, World!");
```

الدمج بين النصوص: ممكن تستخدم الـ + أو الـ concat() علشان تجمع نصوص.

```
=====
```

```
String part1 = "Hello, ";
```

```
String part2 = "World!";
```

```
String result = part1 + part2;
```

قراءة من المستخدم: لو عايز تقرأ نص من المستخدم.

```
=====
```

```
Scanner scanner = new Scanner(System.in);
```

```
String input = scanner.nextLine();
```

قراءة من ملف: لو عايز تقرأ نص من ملف.

3. دوال الـ StringBuilder في C#

الـ **StringBuilder** هو كائن في **C#** يستخدم علشان تعدل النصوص بشكل أسرع، لأن النصوص في **C#** لما تعدل عليها، بينشأ كائن جديد في الذاكرة، لكن الـ **StringBuilder** بيغير النص بدون ما ينشئ كائنات جديدة.

أهم الدوال:

لإضافة نص في آخر النص الحالي: **Append()**

```
StringBuilder sb = new StringBuilder();  
sb.Append("Hello");  
sb.Append(" World");
```

علشان تحط نص في مكان معين: **Insert()**

```
sb.Insert(6, "C# ");
```

علشان تشيل جزء من النص: **Remove()**

```
sb.Remove(6, 3); // "C# "
```

علشان تغير جزء من النص: **Replace()**

```
sb.Replace("World", "C#");
```

لو عايز تمسح كل المحتوى: **Clear()**

```
sb.Clear();
```