

SHA-256 in Action – Cryptographic Hashing  
  
**Objective/Aim:**  
  
 To study the working of the SHA-256 cryptographic hashing algorithm and demonstrate how it generates



fixed-size hash outputs for varying inputs, ensuring data integrity and security.

**Apparatus/Software Used:**

* Laptop – to run programs
* Python 3.x / Online IDE / Remix console – for coding and demonstrations.
* Hashing libraries (hashlib in Python).
* Internet – for reference material and documentation.

**Theory/Concept:**

**What is a Hash Algorithm?**

Hash Function: A one-way mathematical function that converts input data of any size into a fixed-size output.

**SHA-256 (Secure Hash Algorithm 256-bit):**

* Produces a 256-bit (64 hex characters) hash.
* Deterministic (same input → same output).
* Irreversible (cannot derive input from output).

**Properties of SHA-256**:

* Deterministic – Same input always gives same hash.
* Avalanche Effect – Small input change drastically changes hash output.
* Collision Resistant – Very unlikely that two inputs produce the same hash.



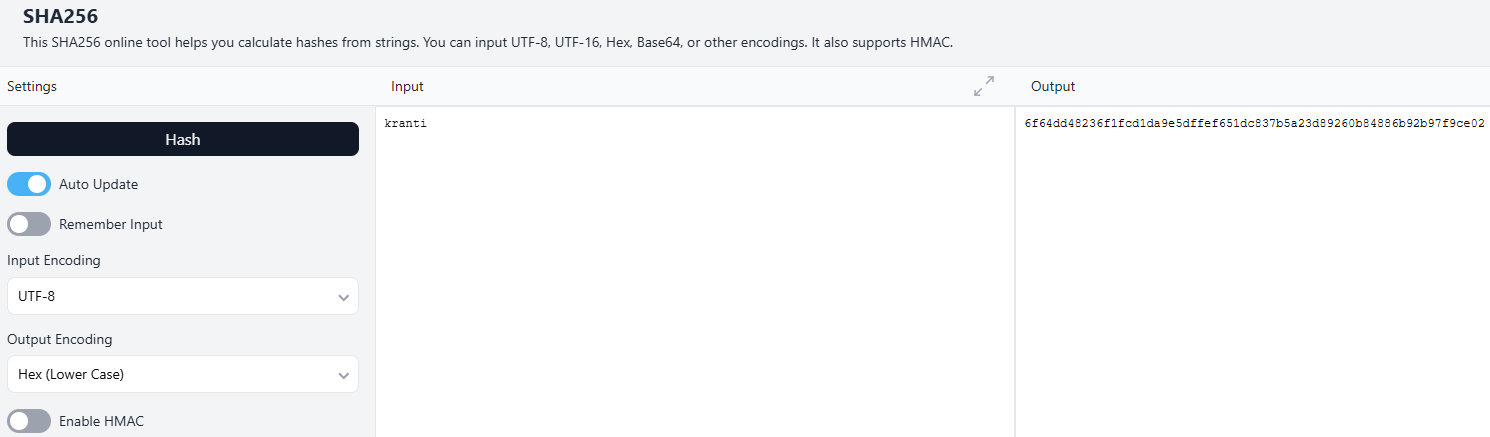
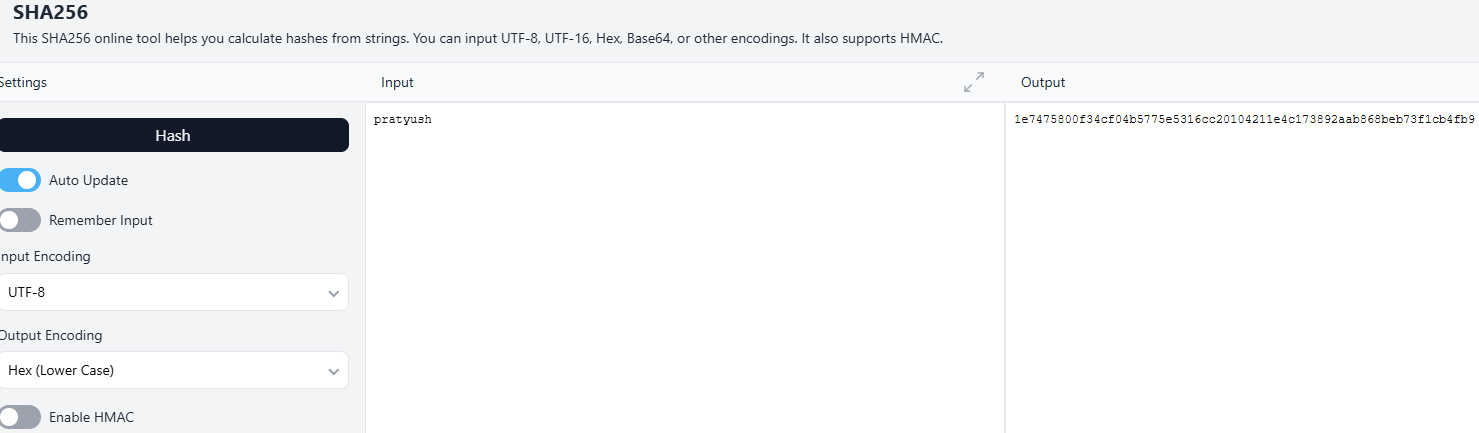
**Procedure:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Step 1:** Go to online SHA-256 online tool <https://emn178.github.io/online-tools/sha256.html>.

**Step 2:** give a string as an input

**Step 3:** get the hash as an output if a single alphabet changes then the hash also changes



**Observation:**

