

 SHA-256 in Action – Cryptographic Hashing  
  
**Objective/Aim:**

To explore the functionality of the SHA-256 hashing technique by applying it on sample inputs and

examining its fundamental properties.

**Apparatus/Software Used:**

* Computer or laptop
* Internet connection
* Online SHA-256 generator tool
* MS Word (for preparing the report)

**Theory/Concept:**

**Hashing Basics**  
 A hashing function is a computational process that converts any size of input data (like text, file, or ; password) into a unique, fixed-length output string. This output is referred to as the **hash** or **digest**.

**About SHA-256**

* SHA-256 stands for Secure Hash Algorithm – 256 bit.
* Belongs to the SHA-2 family of cryptographic algorithms developed by NSA and standardized by NIST.
* Produces a hash of **256 bits (64 hexadecimal characters)** irrespective of the input size.

**Key Features**

* **Deterministic**: Same input → same hash every time
* **Avalanche Effect**: Even a small change in input drastically alters the output.
* **Irreversibility**: Hashes cannot be converted back to original data.
* **Security**: Useful for password storage, digital signatures, and data integrity verification.

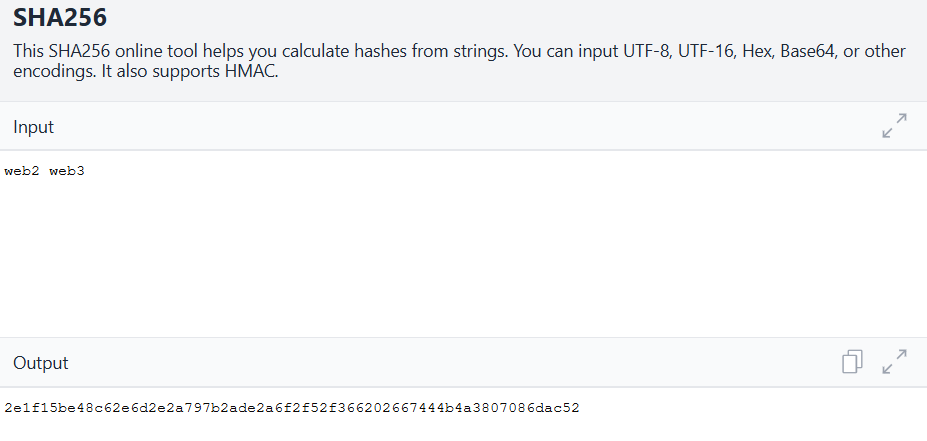
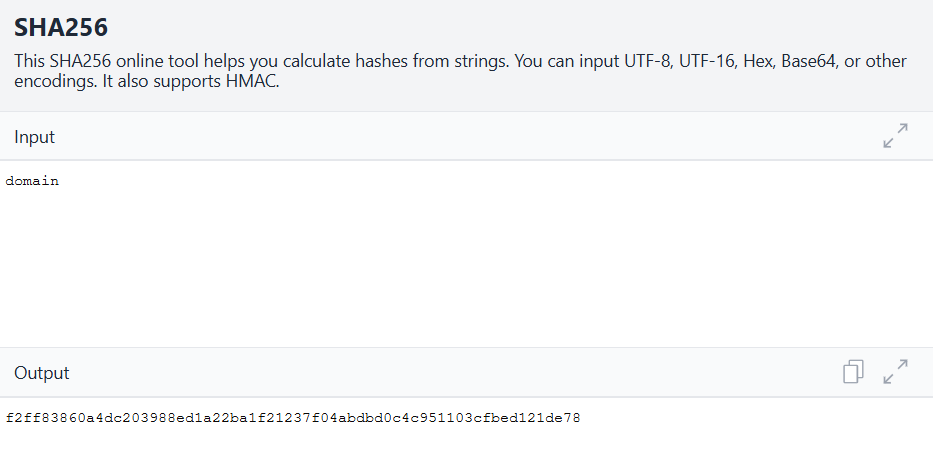


**Procedure:**

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

**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:**

* When the same text was entered multiple times, the tool consistently generated the same hash
* Even a minor alteration in input (such as adding a space) produced a completely different hash value.
* The algorithm provided evidence of the avalanche effect and the one-way nature of hashing.



