**Set - only stores unique values**

my\_set = {1, 2, 3, 2, 1} # Becomes {1, 2, 3}

**Dictionary - stores unique keys with values**

my\_dict = {'a': 1, 'b': 2, 'a': 3} # Becomes {'a': 3, 'b': 2}  
Under the Hood:  
Both use hash tables internally, but dictionaries need to store the additional value for each key, which requires more memory.

The main difference between sets and dictionaries (hash tables) in Python lies in their structure and purpose:

* **Sets** store **only unique keys (elements)** without any associated values. They are primarily used for membership testing, removing duplicates, and performing mathematical set operations like union, intersection, and difference.
* **Dictionaries** store **unique keys with associated values** (key-value pairs). They are used for data storage and retrieval by key, allowing easy access, updating, and management of values related to each key.

**Key Differences**

|  |  |  |
| --- | --- | --- |
| Aspect | Set | Dictionary/Hash Table |
| Storage | Only keys | Key-value pairs |
| Purpose | Membership testing, uniqueness | Data storage and retrieval |
| Use Case | Removing duplicates, membership checks | Storing and accessing data by key |
| Syntax (Python) | {value1, value2, value3} | {key1: value1, key2: value2} |
| Memory | Less memory (stores only keys) | More memory (stores keys and values) |
| Operations | Union, intersection, difference | Key-based value access, updating |
| Example | my\_set = {1, 2, 3, 2} (becomes {1, 2, 3}) | my\_dict = {'a': 1, 'b': 2, 'a': 3} (becomes {'a': 3, 'b': 2}) |

**Under the Hood**

* Both sets and dictionaries use **hash tables** internally for efficient lookups.
* Dictionaries need to store an additional value for each key, which requires more memory, whereas sets only store keys.

**Additional Notes**

* Sets are unordered collections of unique elements.
* Dictionaries are ordered (from Python 3.7+) collections of key-value pairs.
* Both are mutable and do not allow duplicate keys or elements.

This distinction makes sets ideal for operations focusing on unique membership and dicts ideal for structured data storage and retrieval by keys.[[1]](#fn1)[[2]](#fn2)[[3]](#fn3)

⁂