JSON Roundtrip

Background

Data serialization allows structured information to be converted into a text for
mat suitable for storage or transmission. One common format is JSON (JavaScript
Object Notation). In this task, you will perform serialization and deserializati
on of a Python dictionary using JSON and verify the consistency of the process.

Task

You are provided with a dictionary representing a user profile. Your task is to:

1. Serialize the dictionary to a JSON string using Python's `json.dumps()`.
2. Deserialize the JSON string back into a dictionary using `json.loads()`.
3. Compare both the JSON string and the reconstructed dictionary to expected res
ults.
4. Print two specific output messages depending on whether the values match or n
ot.

Input

There is no input to be read. The dictionary is fixed as follows:

```
{
    "name": "Alice",
    "age": 30,
    "is_admin": True,
    "skills": ["Python", "Network Programming", "Digital Forensics"]
}
```

Example output

```
The JSON strings match. {"name": "Alice", "age": 30, "is_admin": true, "skills":
 ["Python", "Network Programming", "Digital Forensics"]} {"name": "Alice", "age"
: 30, "is_admin": true, "skills": ["Python", "Network Programming", "Digital For
ensics"]}
The dictionaries match. {'name': 'Alice', 'age': 30, 'is_admin': True, 'skills':
 ['Python', 'Network Programming', 'Digital Forensics']} {'name': 'Alice', 'age'
: 30, 'is_admin': True, 'skills': ['Python', 'Network Programming', 'Digital For
ensics']}
```

Implementation guide

Implement the following functions:

```
def json_to_variable(data):
    # Returns JSON string
    pass

def unjson_from_variable(json_data):
    # Returns reconstructed dictionary
    pass
```

Then write a main function that:

1. Serializes `test_data` using `json_to_variable()`.

2. Deserializes the result using `unjson_from_variable()`.
3. Prints the two required messages.