#### 代码及说明

**1、代码**

import json

def json\_to\_python(json\_str):

"""

将 JSON 字符串转换为 Python 数据类型。

:param json\_str: str, JSON 格式的字符串

:return: 转换后的 Python 数据（dict、list、str、int、float、bool 或 None）

:raises ValueError: 如果输入不是合法的 JSON 字符串

"""

try:

return json.loads(json\_str)

except json.JSONDecodeError as e:

raise ValueError(f"Invalid JSON input: {e}")

# 测试

if \_\_name\_\_ == "\_\_main\_\_":

test\_cases = {

'null': "null",

'boolean\_true': "true",

'boolean\_false': "false",

'integer': "123",

'float': "12.34",

'string': '"hello"',

'list': '[1, 2, 3, "a", null]',

'dict': '{"name": "Alice", "age": 25, "is\_student": false}'

}

for name, json\_input in test\_cases.items():

try:

result = json\_to\_python(json\_input)

print(f"{name}: {result} (type: {type(result)})")

except ValueError as e:

print(f"{name}: Error - {e}")

**2、说明**

定义了 json\_to\_python 函数，该函数使用 Python 内置的 json.loads() 方法解析 JSON 字符串，并将其转换为相应的 Python 数据类型：

* null → None
* true → True
* false → False
* JSON 数字 → Python int 或 float
* JSON 字符串 → Python str
* JSON 数组 → Python list
* JSON 对象 → Python dict

如果输入的 JSON 无效，json.loads() 会抛出 JSONDecodeError，我们将其转换为 ValueError 以提供更清晰的错误信息。

**二、 测试**

在 if \_\_name\_\_ == "\_\_main\_\_": 代码块中，定义了一些测试用例，包括：

* null
* true/false
* 整数
* 浮点数
* 字符串
* JSON 数组
* JSON 对象

每个测试用例都输出转换后的 Python 值及其类型，以验证转换是否正确。运行代码后，输出如下

