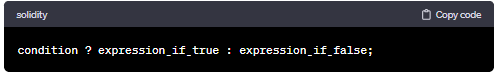
**Ternary Operator:**

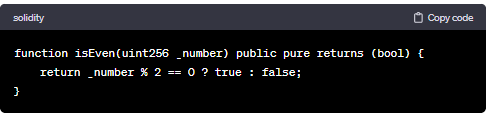
In Solidity, the ternary operator is used to write concise conditional statements.



**Use Case:**

* Suitable for simple, one-liner conditions where you want to assign a value based on the condition.
* Concise and often used for straightforward assignments.

Here's an example to illustrate its usage:



**Considerations:**

* **Readability:** Ternary operators are more concise but can be less readable if the condition and expressions are complex. If-else statements provide more structure for complex conditions.
* **Multiple Statements:** If your conditional logic involves more than one statement for each branch, an if-else statement might be more suitable, as it allows you to use multiple lines of code within each block.
* **Gas Costs:** Generally, the difference in gas costs between the two approaches is minimal. However, complex conditions and multiple statements may affect gas costs, so it's important to consider optimization.

**Code:**

//SPDX-License-Identifier: GPL-3.0

pragma solidity >= 0.5.0 < 0.9.0;

contract ter{

    function fun(uint \_x) public pure returns(string memory){

        string memory val;

        val = \_x>100 ? "greater" : "equal or smaller";

        return val;

    }

}

