Staking Rewards

Henry Tirla

Blockchain Anlayst Challenge

I have a series of connected pipes through which water flows. Each pipe represents an operation or line of code in my Pipescript program. Now, let me consider the scenario where an error occurs at a specific point in the flow.

Without the "skip on error" directive, the error would cause the water flow to stop entirely, none of the subsequent pipes would receive water, and the process would halt. It is similar to how an error in a line of code would typically stop the execution of the program.

However, with the "skip on error" directive, Pipescript handles the error differently. It introduces a bypass valve at the point of error. When an error occurs at that point, the bypass valve is activated, and the water flow is diverted away from the problematic pipe.

This allows the water to continue flowing through the rest of the pipes, unaffected by the error. The subsequent pipes in the flow still receive water and continue their operations. Only the specific pipe where the error occurred and the pipes that depend directly or indirectly on that point are bypassed.

Regarding Pipescript, the "skip on error" directive acts as the bypass valve. It ensures that if an error occurs at a specific line of code, the execution continues from the following line, skipping the problematic line and any subsequent lines that depend on it.

Here's a simple diagram to illustrate my understanding of the concept:

```
     Error
       |
   +---v---+
   | Pipe 1|
   +---+---+
       |
   +---v---+
   | Pipe 2|
   +---+---+
   /   |
  /    v
 |  Skip on Error
  \   |
   \  v
   +---+---+
   | Pipe 3|
   +---+---+
       |
       v
   Rest of the Pipes
       |
       v
   +---+---+
   | Pipe 4|
   +-------+
```

In this diagram, Pipe 2 encounters an error. Without the "skip on error" directive, the execution would stop at Pipe 2. However, with the directive, Pipe 2 is skipped, and the execution continues from Pipe 3 and beyond.