🧠 Block By Block: Don’t Just Be Right—Be Clear

WHAT❓

Readable code tells the reader **what’s happening** and **why**, rather than just works correctly. Writing tests and conditions that mirror our **mental model** makes logic easier to understand.

HOW❓

✅ Express the Subject of the Check First

🔴 Less readable:

```python
assert expiration_time > time.time(), "Transaction expired"
```

🟢 More readable:

```python
assert time.time() < expiration_time, "Transaction expired"
```

💬 Why?

We’re checking whether the **current time is still valid**—that’s the subject, so it should come first. Both versions are logically equivalent, but the second version **reads like a natural sentence**: “We’re before the expiration time”, rather than “The expiration time is in the future”.

✅ Express Nested Logic Clearly

🔴 Less readable 1:

```python
assert has_write_permissions or is_read_operation, "No write permissions."
```

🔴 Less readable 2:

```python
if not has_write_permissions:
   assert is_read_operation, "No write permissions."
```

🟢 More readable:

```python
if is_write_operation:
   assert has_write_permissions, "No write permissions."
```

💬 Why?

All are logically equivalent—but the last version **tells the story**. It reads:

“If the user is performing a write operation, we require that they have write permissions”.

The other two versions read as “Check that the user has a write permission or they are performing a read operation” and “If the user doesn’t have write permissions, check that the operation is read”.