Time loop variables

NOTE: *The word "**heart**" appears approximately 1,000 times in the Bible.*

Channeled messages from Coeus, God of Intellect

Please READ THEM IN ORDER. You might want to read each one a couple times; it helps. Coeus has a huge vocabulary and is very selective of his words.

Coeus — God of Intellect (12/28/17)

Just when I thought communicating through light language was interesting, now Coeus is throwing time loop variables at me, and I had no idea what they were, or even existed for that matter.

This was once again, another message that I was about to scrap, prior to transcription. My brain felt so scrambled that I wasn't sure anything even made sense.

But before we proceed, I'd like you to read this scripture from the Bible, just in case you didn't read the 12/24/17 channeled message from Coeus.

Revelation 10:6

*And he swore by him who lives to the eternity of eternities, him who created Heaven and that which is in it, and The Earth, and that which is in it: **"Time shall be no more."***

(D) I'm sorry, I thought that I was dreaming, but then I realized it was you talking.

Fortunately for you, within the confines of reality, you cease to know or understand the significance of the events that are occurring simultaneously throughout your society's structures.

(D) This seems redundant, like you just said it again. This is weird.

(D) Sorry, there it goes again. Something about time slippage occurring.

Time slipping is occurring during your dream and waking states. That is why you must adhere to strict timelines resulting in your…

(D) This is really coming in quick and duplicating.

 ...ability to relay accurate…ability to relay accurate…

(D) It's like pulsing in two times in a row.

 Two mints, two mints, in one.

(D) Isn't that an old gum commercial?

(D) OMG, this is going so fast! My heart is beating in my head again.

 That's your epicenter for you to comprehend these fine messages and truths. This knowingness often takes time to bring together your abilities to accurately receive time loop variables,...

(D) This is fascinating.

 …occurring within your; to receive these messages without your brains capacity to dissect and receive to convey the information accurately.

(D) The information is coming in extremely fast and pulsing and duplicating. It might be coming in the beginning of the heart beat and the end of the heart beat. For example, if I press my hand down on the bed, the release it, it duplicates.

 What an interesting day this is going to prove or be.

(D) Now that was interesting. At first I thought I was dreaming, but then I realized it was you talking, yet it seemed like a dream, but I'm not asleep yet.

 This all came about today, when you asked about your friend Cindy's ability to effectively communicate the changes of both your scheduled performances. You should explain to them.

(D) So anyway, the water was shut off to the houseboat last week and I was upset, because I wasn't notified. If I knew, I would have filled my water tanks up. Cindy insisted that she notified me via text and the phone. I told her that I sent her an email and called her, but neither of us got anything. I thought it was strange that she would do that as

she's efficient.

(D) When I talked to her, I told her that I thought this was a shift in dimensions or a time slippage. She said that she believes we're in end times and that doesn't seem that strange. Her daughter speaks to the angels.

Before you get too ahead of yourself, perhaps we should look at **time events occurring simultaneously**. Throughout your societies delicate needs for discerning truths. Those(?) can be cancerous for stating the truths within the confines of ? in the ? systems decay…

(D) It's like some of the words are "sling-shoting" around. He wants me to explain what's happening.

It's important to relay the details accurately.

Time loop variable defined.

The first part, initialization, is a piece of code that's run once when the loop starts. In this case, you're creating a new variable i, and giving it the value 0, so you can keep track of how many times the loop has been run.

The second part, the condition, is checked each time the loop starts; if it's true, the loop code (In this case, doSomething();) gets run. If it's false, then the for loop is over, and the program continues past it.

The last part, the afterthought, is a piece of code that's run each time the loop runs. It's usually used to increment our counter variable; in this case it adds 1 to the i variable.

If we put this together, then this particular for loop says:
• Make a variable called i; give it the value 0.
• Is i less than exponent? If so, run the code doSomething();.

If not, stop.
• If we ran the code, also add 1 to i.
• Then go back to step 2, until we encounter an iteration where iis not less than 23.

There's a couple of important things to note here.

Firstly, any variable created in the initialization stays in the for loop. When you create the variable i, it exists only within this particular for loop. This concept is called "scope", and you'll see it a lot.

Secondly, you don't have to use the initialization to create

a variable. You could increment some unrelated variable instead, or you could run a function, or you could even do nothing at all. Same goes for the condition and afterthought; you could check the time of day via a system function in the condition, and you could increase a variable a which you made outside this for loop or run a function in the afterthought.

In fact, you need not even fill in all three parts. For example, I could write the following:

```
for ( ; ; ) {
doSomething();
}
```
That's just saying:
• Don't do anything special on the first run.
• There's nothing to check for when you should stop (a blank condition is considered to be always true).
• Don't do anything when each iteration of the loop finishes.

Basically, an infinite loop.

Lastly, the ; marks the end of a statement. Each complete statement must end in a semicolon. A statement is just one action in code. An entire for loop is one statement, for example, and so is an entire if/else statement (including the code to run in each case).