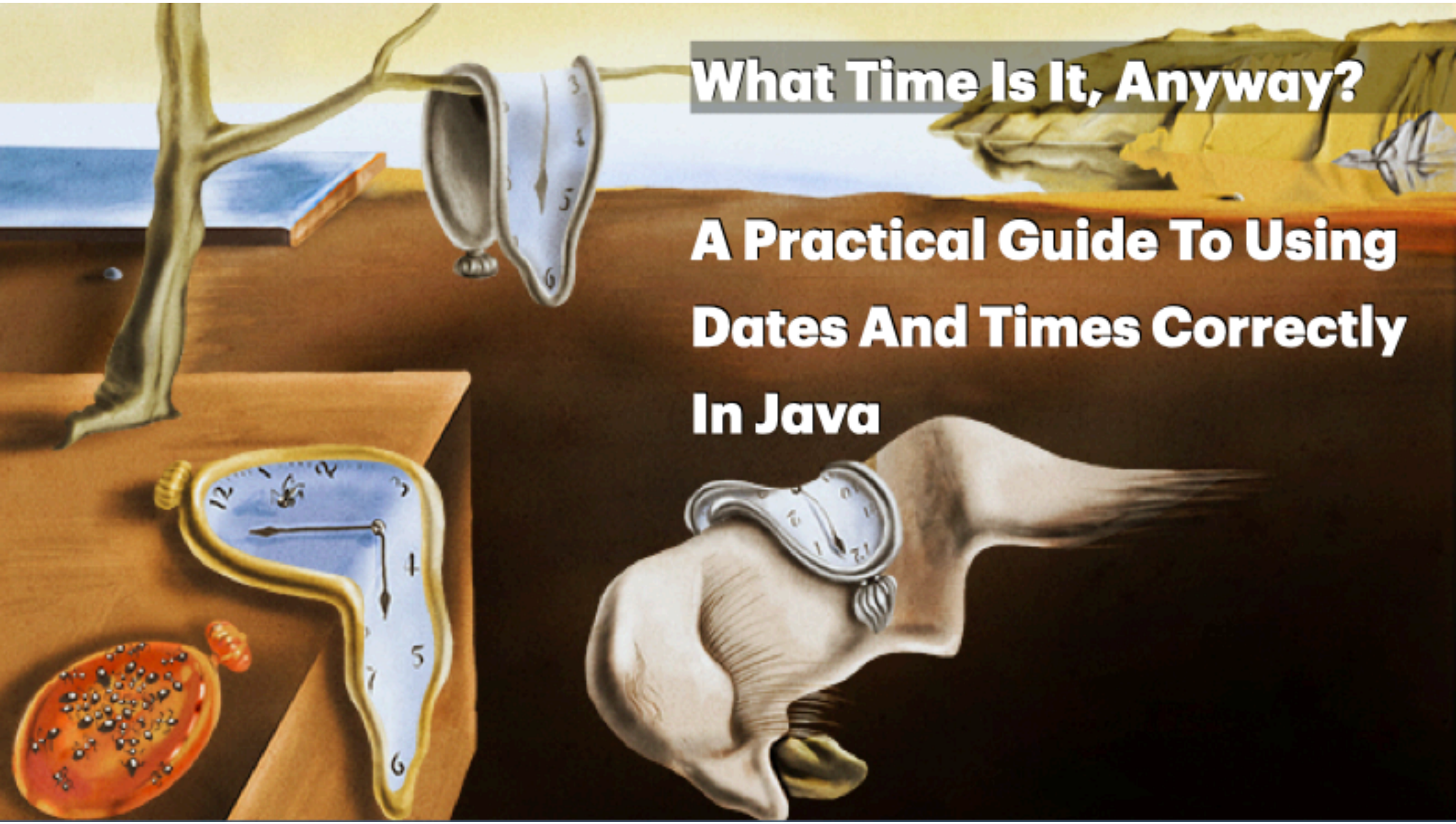

What Time Is It, Anyway?
A Practical Guide To Using Dates And Times Correctly In Java

Today's Agenda

- Introduction
- A Brief History Of Time
- UTC - Coordinated Universal Time
- ISO-8601
- Why early Java date/time needed work
- JodaTime
- The Java 8 Date and Time APIs
- ZonedDateTime, LocalDateTime, And More!
- Some Examples
- Suggestions for unit testing
- ThreeTen
- Resources

Kelly Morrison, Ph.D. Computer Engineering

Solution Architect - Daugherty Business Solutions

The University of Georgia

TEAM Daugherty Est. 1985

CNN

THE HOME DEPOT

COLORBUS

AUBURN UNIVERSITY

Turner

DIGITAL output MAGAZINE

PATHFIRE Enabling Digital Media

Edward Jones MAKING SENSE OF INVESTING

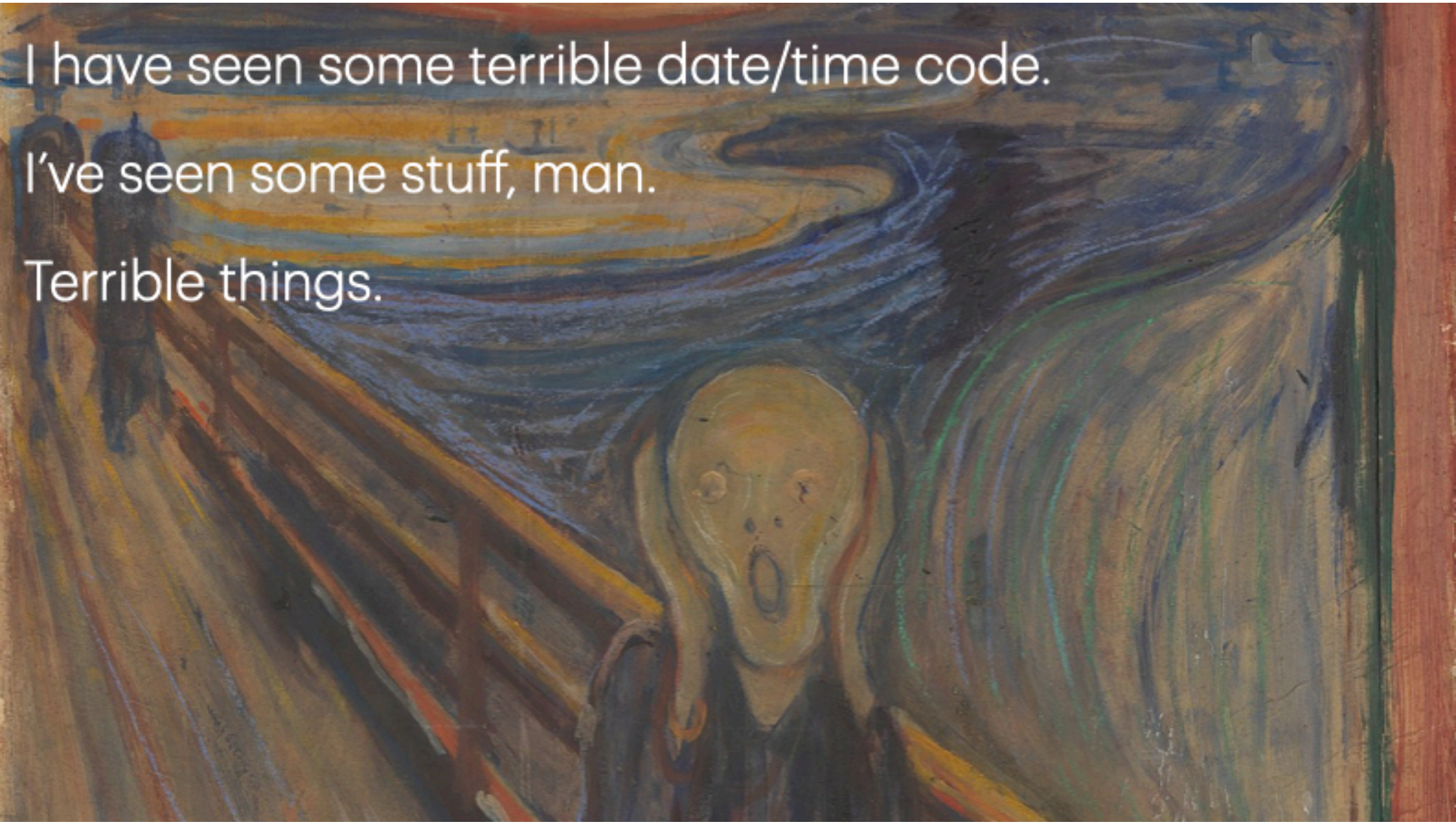
THE CLIMATE CORPORATION

NASA Marshall Space Flight Center

Performance MATTERS

BAYER


But... why?


I have seen some terrible date/time code.
I've seen some stuff, man.
Terrible things.

Example #1: “I don’t know Date(), but I know regex!”

```java
import java.util.*;
import java.util.regex.*;

public class Main {

    public static void main(String[] args) {
        Date currentDate = new Date();
        System.out.println("Current date = " + currentDate);

        String timePattern = "(\\d{2}:\\d{2}:\\d{2})"; // Pattern for hours:minutes:seconds
        Pattern pattern = Pattern.compile(timePattern);

        Matcher matcher = pattern.matcher(currentDate.toString());

        if (matcher.find()) {
            String time = matcher.group(1);
            System.out.println("Current time: " + time);
        } else {
            System.out.println("Could not get the time.");
        }
    }
}
```

```
Current date = Wed Apr 10 22:49:08 EDT 2024
Current time: 22:49:08
```

Example #2: “Formatting? I’ll do it myself.”

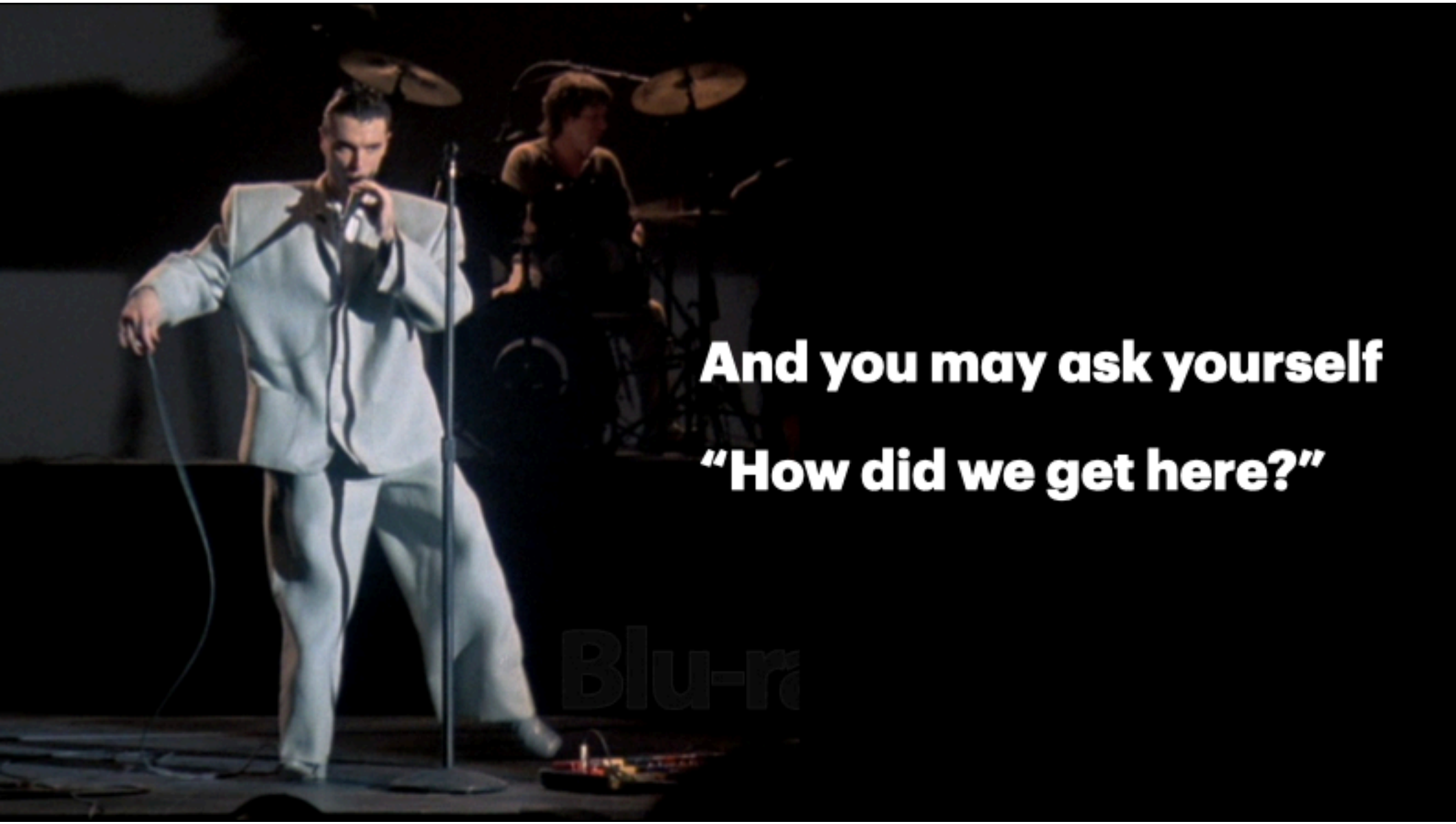
```java
import java.util.*;

public class Bad2 {

    public static void main(String[] args) {
        Date currentDate = new Date();

        String currentTime = String.format("%02d:%02d:%02d", currentDate.getHours(), currentDate.getMinutes(), currentDate.getSeconds());

        System.out.println("Current time: " + currentTime);
    }
}
```

```
Current time: 22:59:29
```

Example #3: The Day The Unit Tests Died

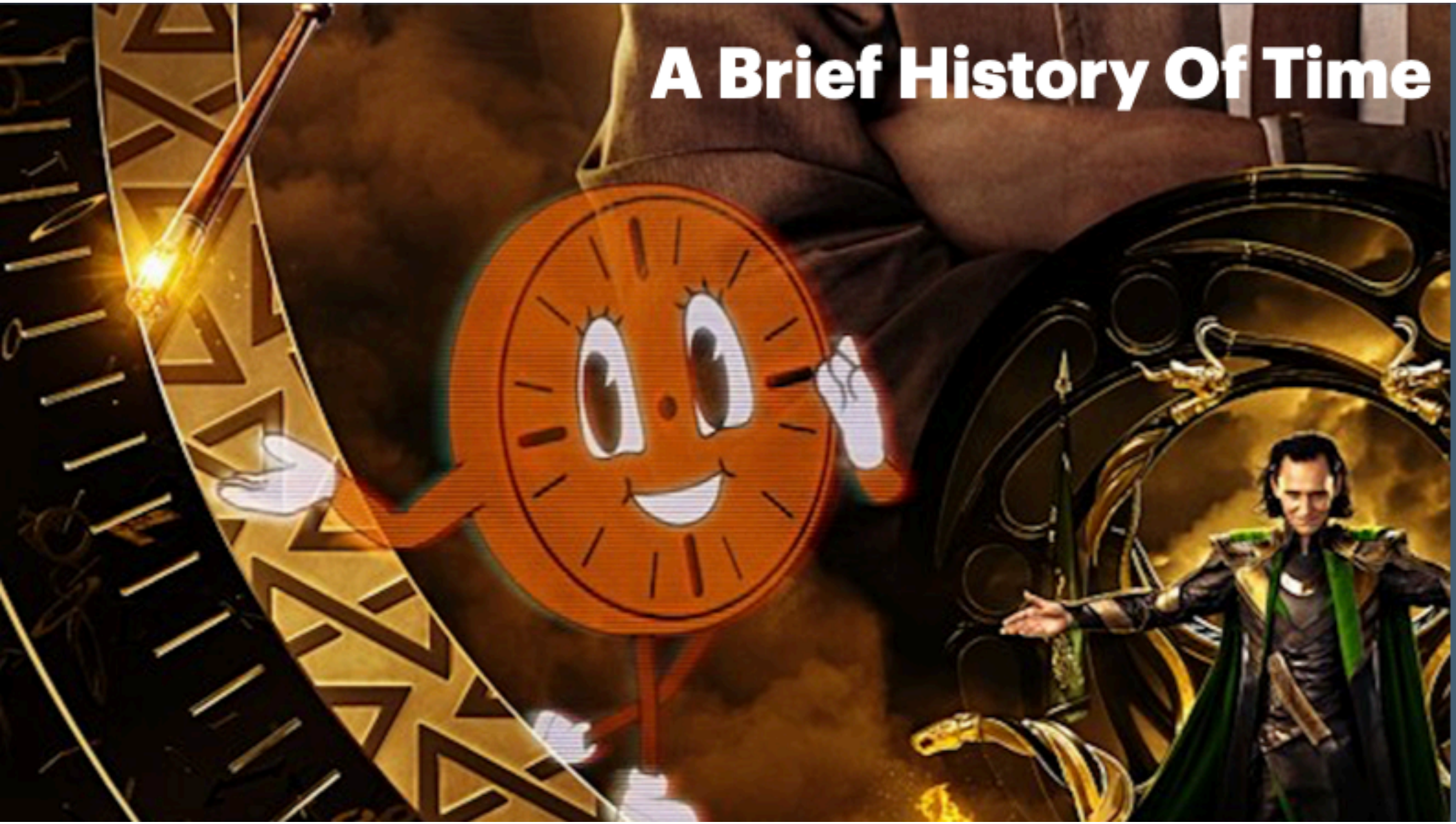
- I was working with a team on a project that scheduled appointments.
- One day, the unit tests started failing. Everywhere. Even on machines where nothing had changed.
- The culprit was in the unit tests themselves.
- The tests were checking to see if an appointment was in the summer...

```
Date firstDayOfSummer = new Date();
firstDayOfSummer.setMonth(5);
firstDayOfSummer.setDate(1);
firstDayOfSummer.setDate(2012);

Date today = new Date();

boolean isNotYetSummer = today.before(firstDayOfSummer);
```


And you may ask yourself
“How did we get here?”

Well… it *is* confusing.


datetimeformatter
epoch
offsetdatetime
zonedtime
date
seconds
iso
leap
zoneid
period
calendar
duration
gmt
simpledateformat
instant
timezone
chronounit
zoneddate
unix
utc
zoneddatetime


A Brief History Of Time

Where did these numbers come from?

24 hours in a day?
60 minutes in an hour?
60 seconds in a minute?

Why 24? Why 60?


Egypt - 1500 BC

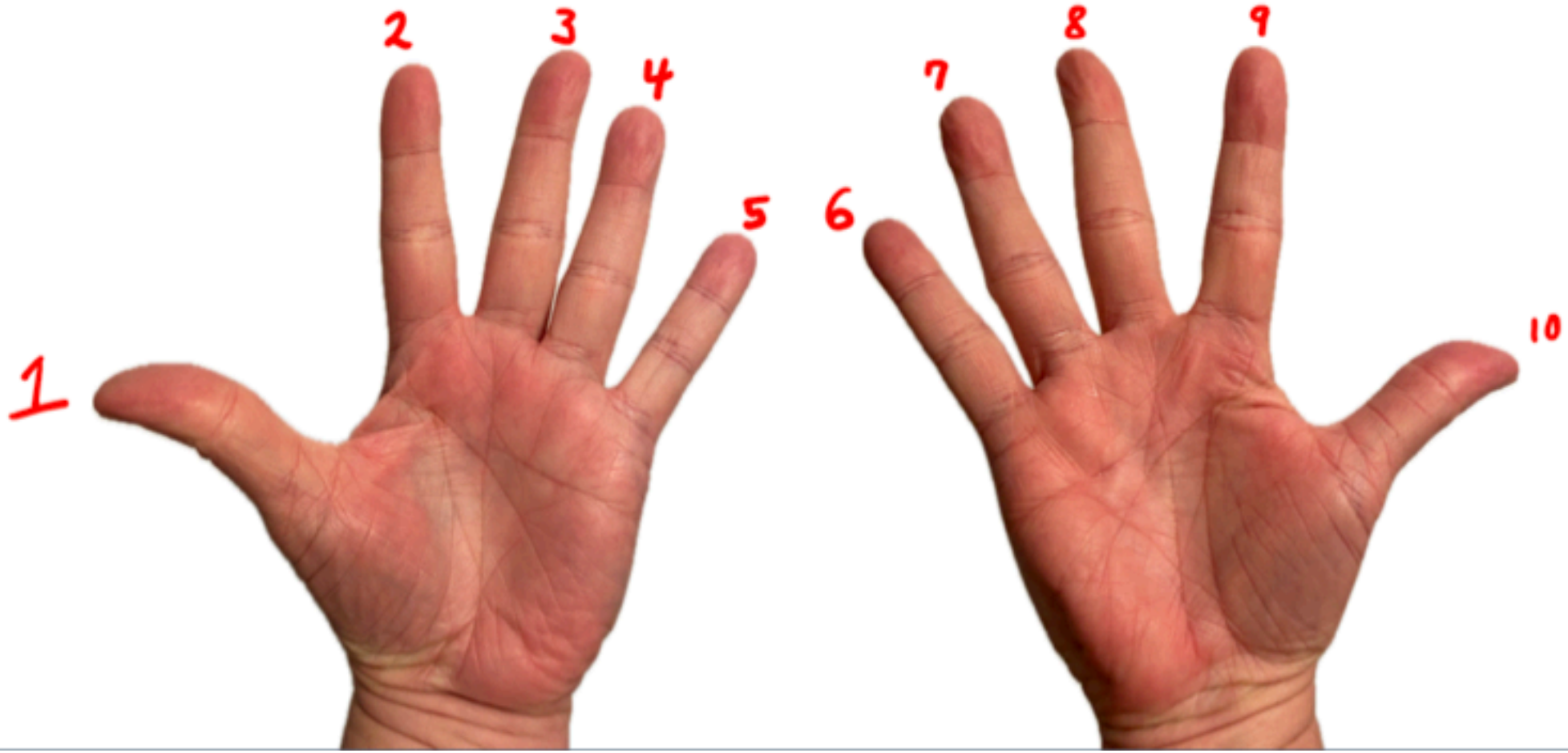
Sun Dial

Each had

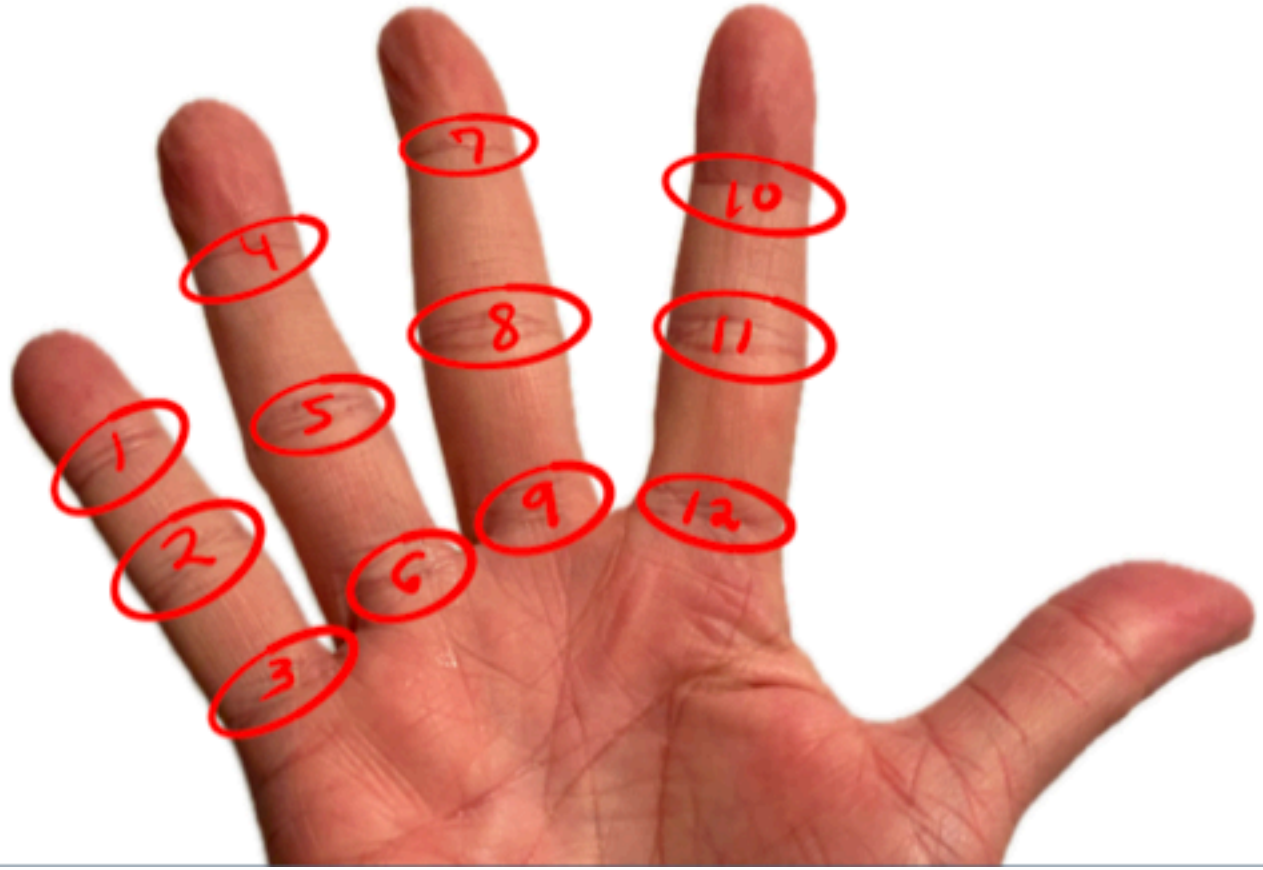
12 divisions.

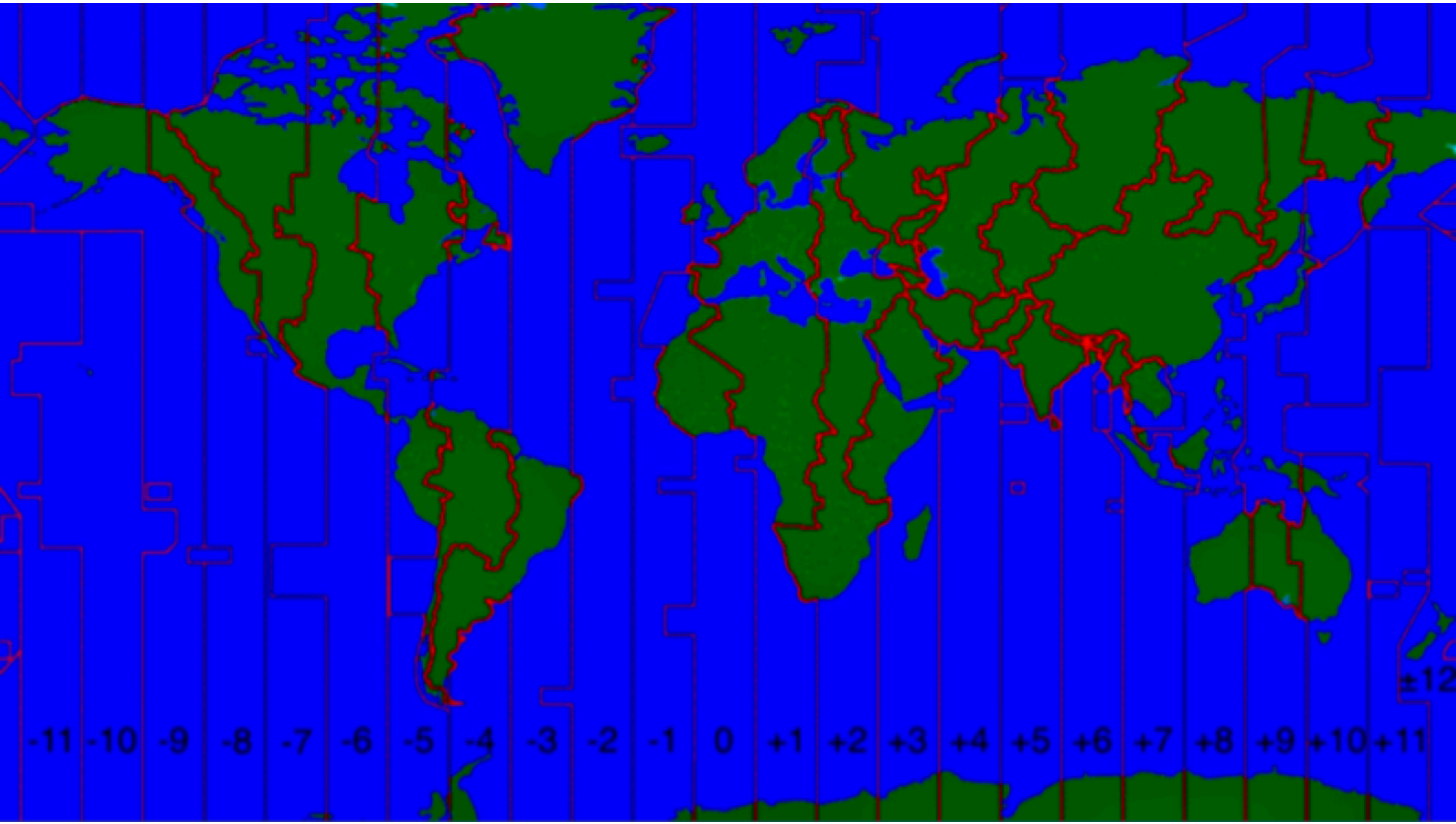
Water Clock

Why 12? Why not 10? We're physically made to count to 10!



There is some evidence that Egyptians counted to 12 on each hand by counting *joints*, not *fingers*.




-11
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
0
+1
+2
+3
+4
+5
+6
+7
+8
+9
+10
+11
±12

Babylon (Iran) - 1000 AD

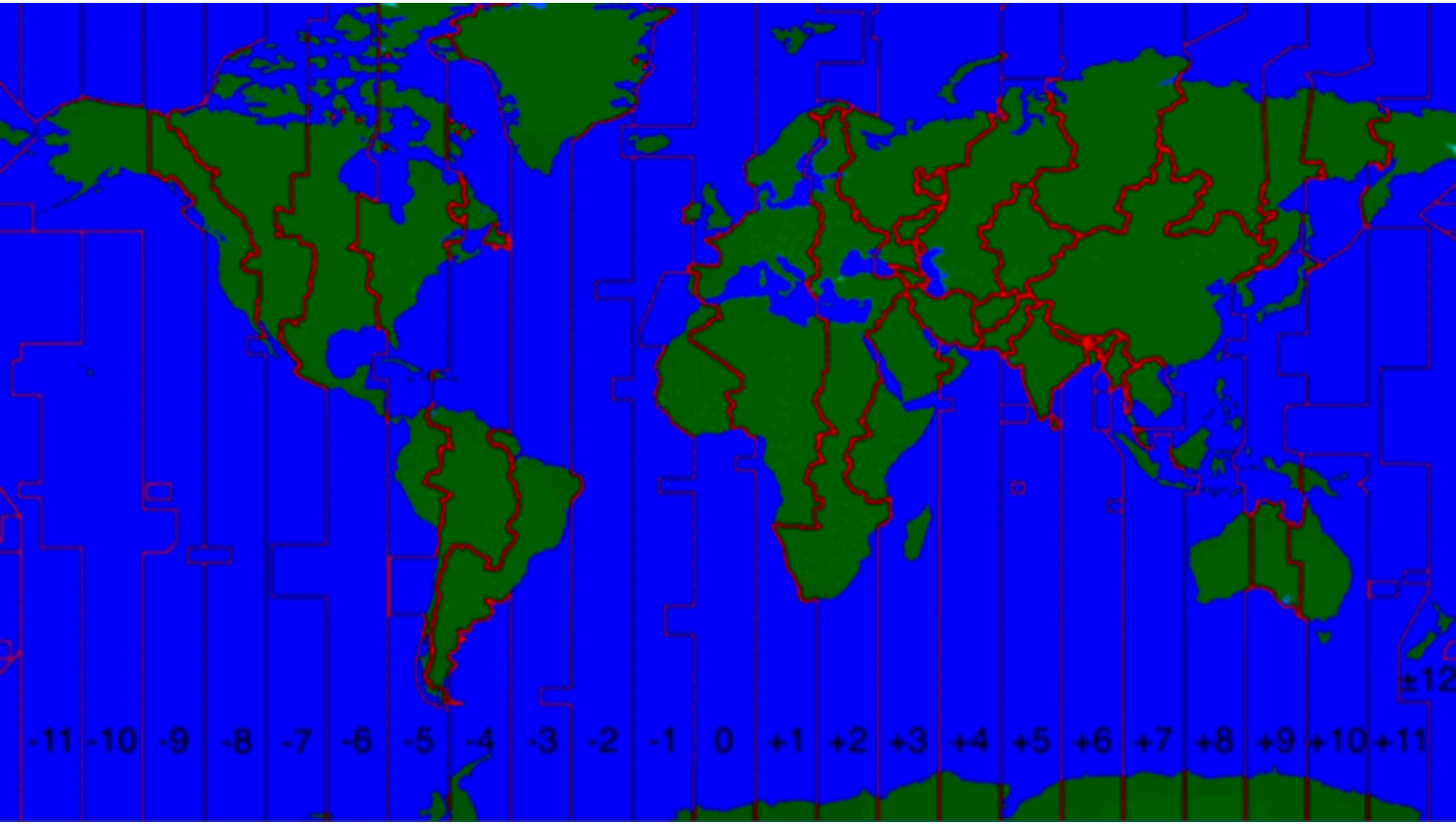
- They used base 60, not base 10.
- al-Biruni split hours into minutes, seconds, thirds, and fourths
- These came from the Latin terms:
  - *pars minuta prima* - "first small part"
  - *pars minuta secunda* - "second small part"
  - each part was 1/60th of the previous

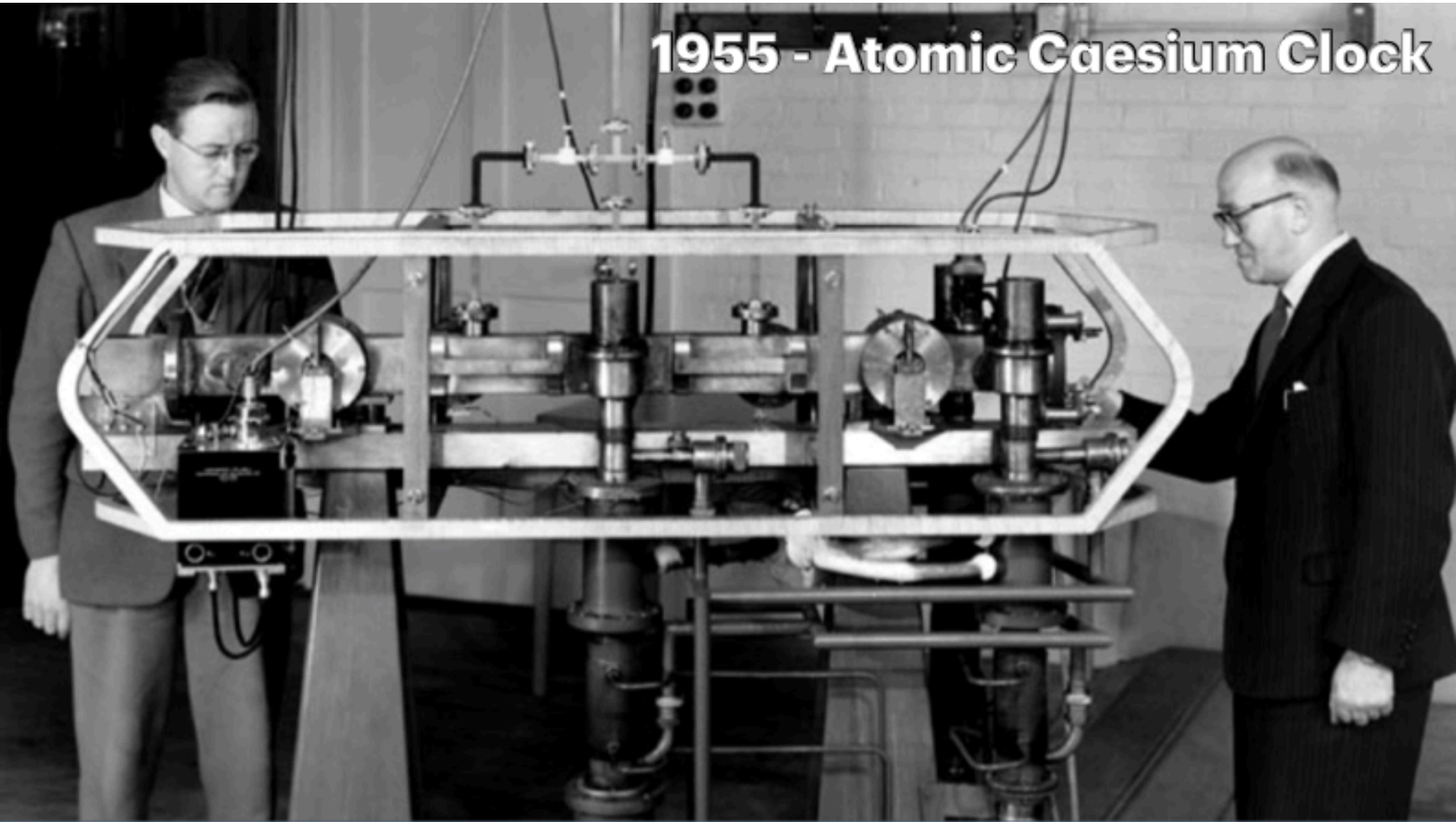
Abu Rayhan al-Biruni

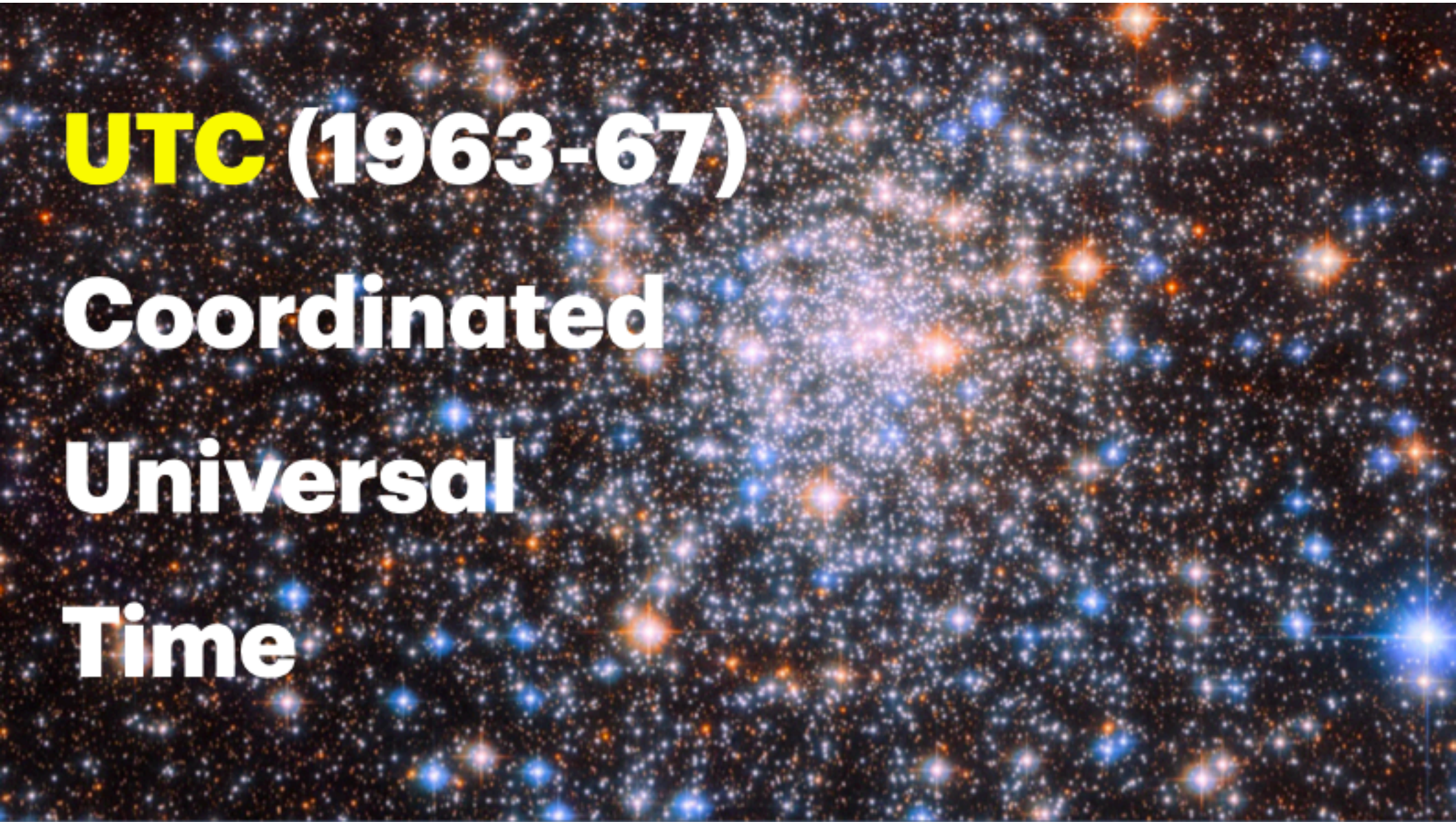
ابوریحان محمد بن احمد البیرونی

An imaginary rendition of Al Biruni on a 1973 Soviet postage stamp


GMT - Greenwich Mean Time (1675)


-11
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
0
+1
+2
+3
+4
+5
+6
+7
+8
+9
+10
+11
±12


1955 - Atomic Caesium Clock


UTC (1963-67)
Coordinated
Universal
Time

UTC

- Tries to track the atomic clock closely
- However it occasionally adds or subtracts "leap seconds" to try to make sure it stays synchronized with the Earth's rotation
- This is because the Earth doesn't rotate at a constant speed
  - It slows up or speeds down based on many factors (irregular shape, gravity interactions with the Sun and moon, etc.)
- Since we can't predict leap seconds, it means calculating the difference between a UTC time now and in the future won't be exact

Epochs

- An epoch is a fixed point in time that computers use as a reference.
- The most famous is the "UNIX epoch", which starts at January 1, 1970. Computers count the number of seconds that have elapsed since then.
- We may have a "Year 2038" problem when the number of seconds will overflow the 32-bit integer UNIX uses to store it.
  - Most UNIX systems are now 64-bit, which is good for 292 billion years.

ISO 8601 (1988)

- International standard for worldwide exchange of date and time data
- There isn't a single ISO 8601 "format": instead, it provides a description of how to carefully specify date and time formats

Some ISO 8601 examples

- Date only: `2024-04-10`
- Date and time separated by T: `2024-04-10T12:30:45`
- Date and time with timezone offset: `2024-04-10T12:30:45+02:00`
- Date and time with Zulu time (UTC): `2024-04-10T12:30:45Z`
- Date and time with fractional seconds: `2024-04-10T12:30:45.123`
- Date and time with fractional seconds and timezone: `2024-04-10T12:30:45.123+02:00`
- Date with week number: `2024-W15-3`
- Date with week number and day: `2024-W15-3T12:30:45`
- Ordinal date (day of the year): `2024-100`
- Ordinal date with time: `2024-100T12:30:45`
- Duration: `P3Y6M4DT12H30M5S` (3 years, 6 months, 4 days, 12 hours, 30 minutes, and 5 seconds)
- Duration with negative sign: `-P3Y6M4DT12H30M5S`
- Date range: `2024-04-10/2024-04-20`
- Date range with times: `2024-04-10T12:00:00/2024-04-20T18:00:00`
- Period starting from a date: `2024-04-10/P1Y` (1 year period starting from April 10, 2024)


UTC
Coordinated
“Universal”
Time
(not you, moon)

LTC (TBD)

Coordinated

Lunar

Time

White House directs NASA to create a new time zone for the moon

News By Sharmila Kuthunur published 2 days ago

The moon may have its own time zone by the end of 2026.

Comments (4)

Artist's illustration of two Artemis astronauts at work on the lunar surface. (Image credit: NASA)

The White House has tasked NASA with creating a new time zone for the moon by the end of 2026, as part of the United States' broader goal to establish international norms in space.

The direction to set up a lunar time zone comes amid growing global interest for humanity to establish a long-term presence on the moon in the coming years — a chief priority of NASA's Artemis program.

QR link to White House announcement