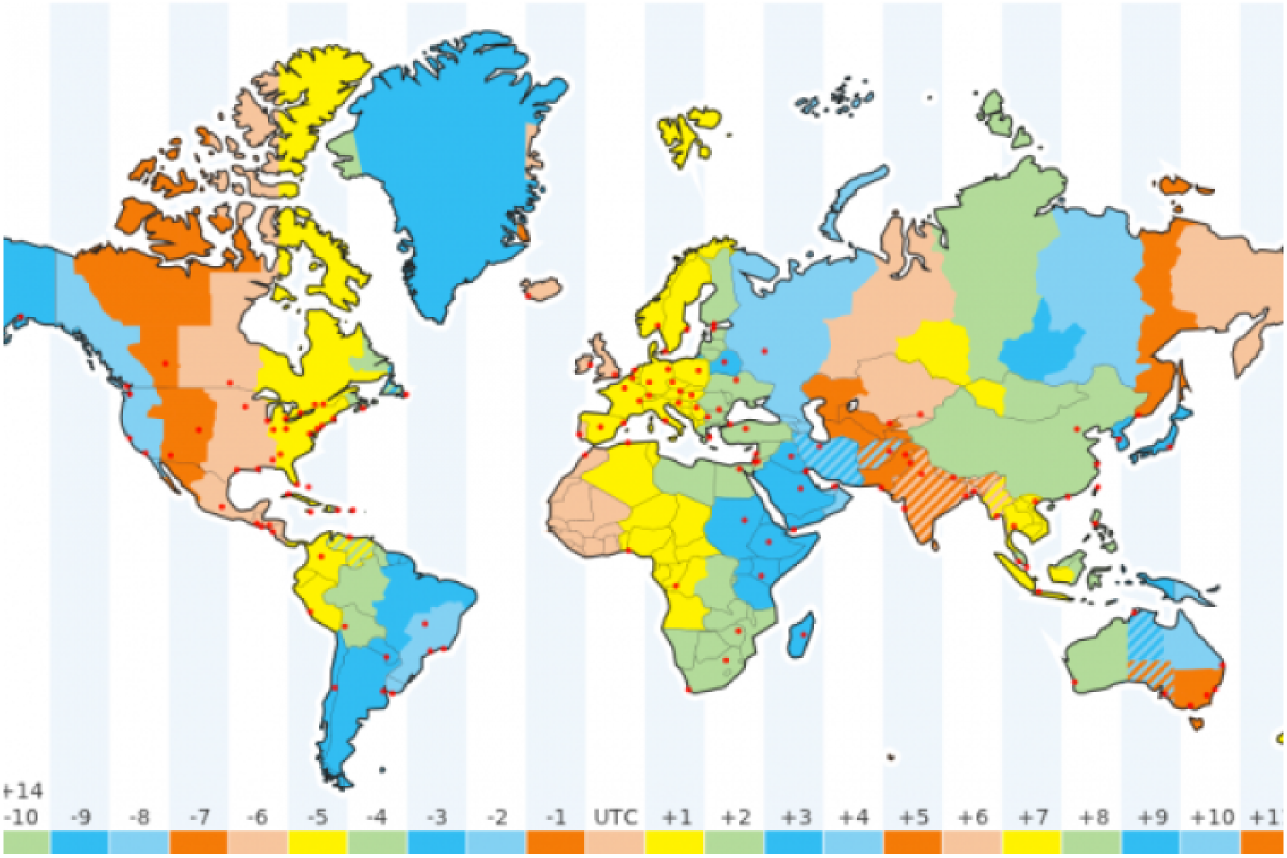
A brief introduction to time zone

and related concepts



A Time Zone:

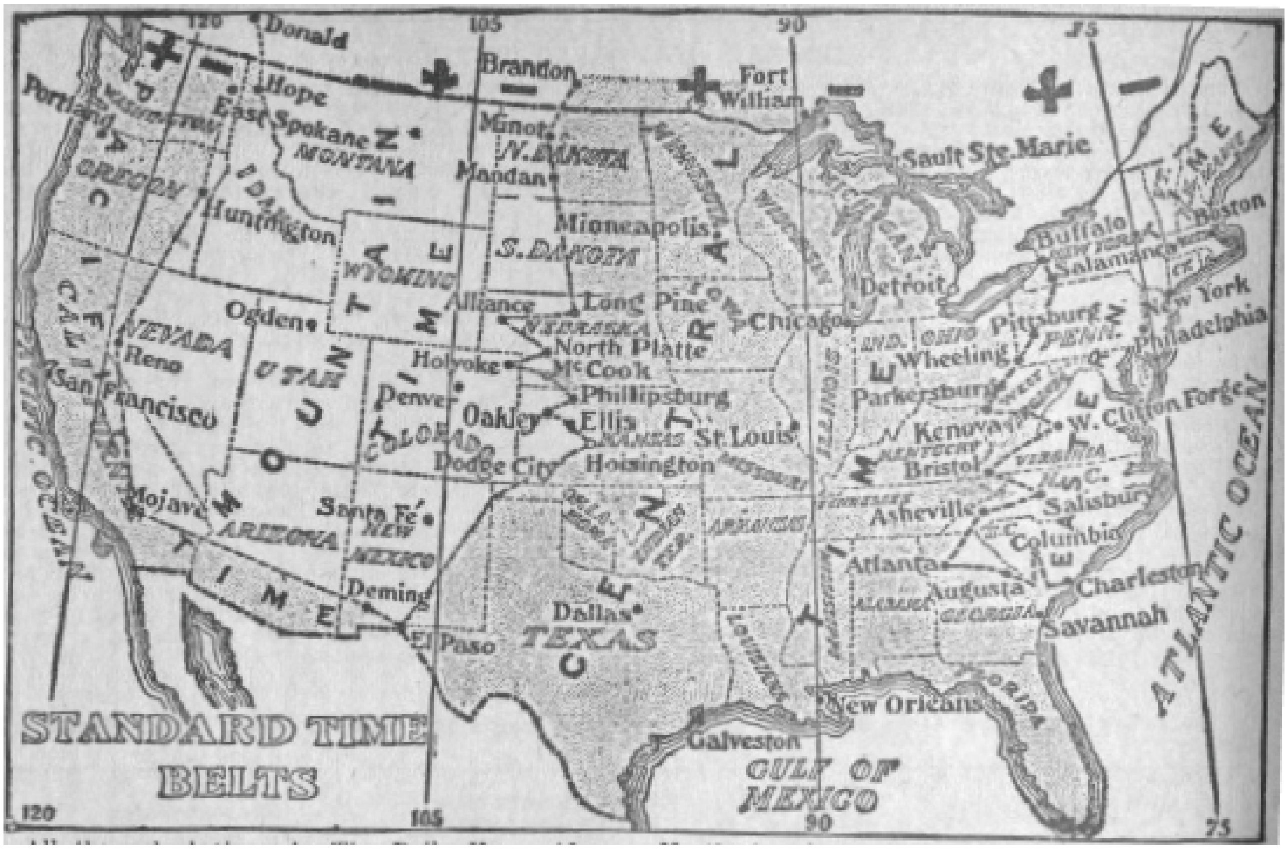
It is a region.
It might have offset from UTC (-12 to +14)

A time zone is a region of the globe that observes a uniform standard time for legal, commercial, and social purposes. Time zones tend to follow the boundaries of countries and their subdivisions because it is convenient for areas in close commercial or other communication to keep the same time.

Most of the time zones on land are offset from Coordinated Universal Time (UTC) by a whole number of hours (UTC−12 to UTC+14), but a few zones are offset by 30 or 45 minutes (for example Newfoundland Standard Time is UTC−03:30, Nepal Standard Time is UTC+05:45, and Indian Standard Time is UTC+05:30). Some higher latitude and temperate zone countries use daylight saving time for part of the year, typically by adjusting local clock time by an hour. Many land time zones are skewed toward the west of the corresponding nautical time zones. This also creates a permanent daylight saving time effect.

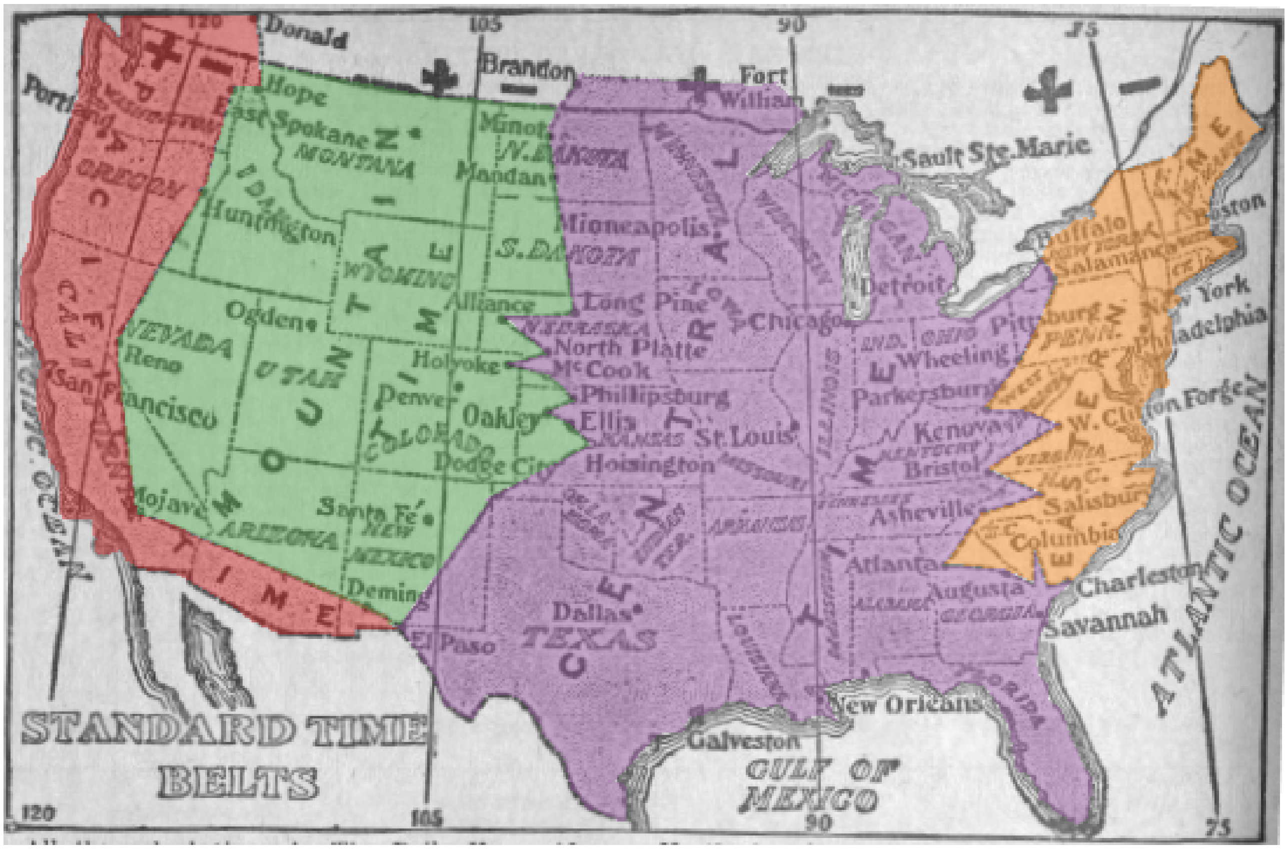
Railways Standard Time

Time Zones were initially adopted by Railways, both in Britain and US.



[Official TZ map of USA in 1913]

Colorized version of the image above:



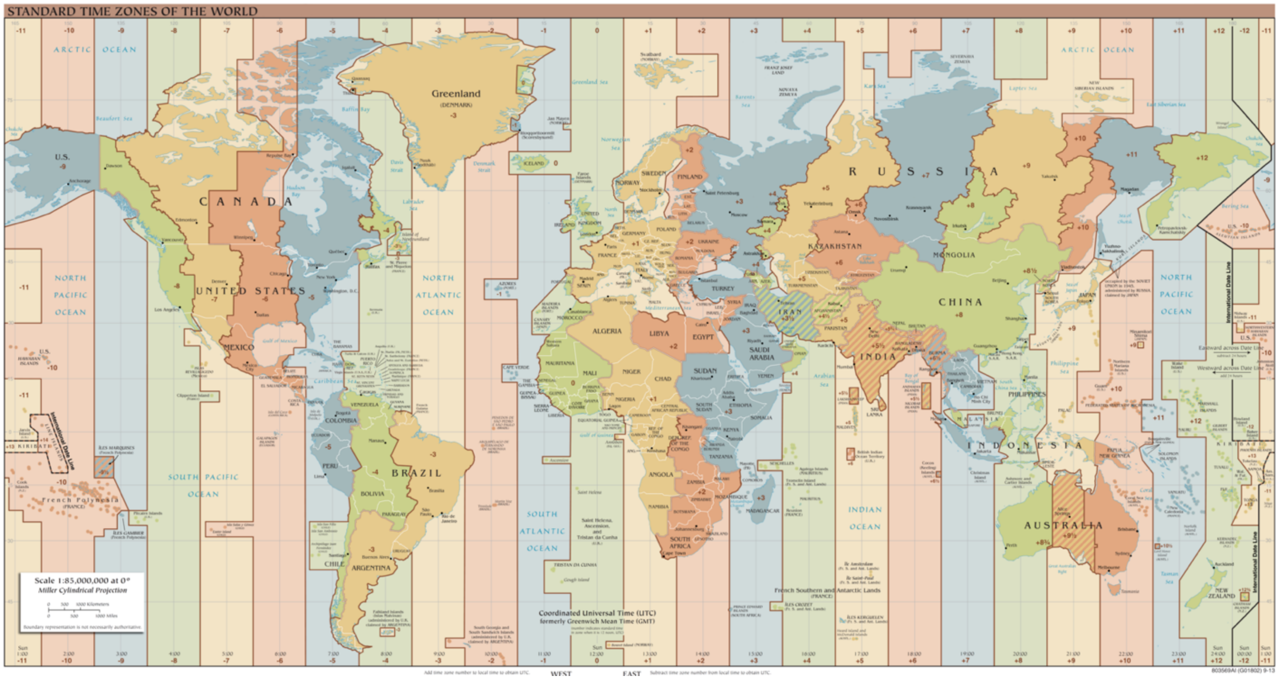
In US, The confusion of times came to an end when Standard zone time was formally adopted by the U.S. Congress in the Standard Time Act of March 19, 1918.

First country to adopt a standard time zone.
New Zealand became the first country to officially adopt a standard time zone on November 2nd, 1868.

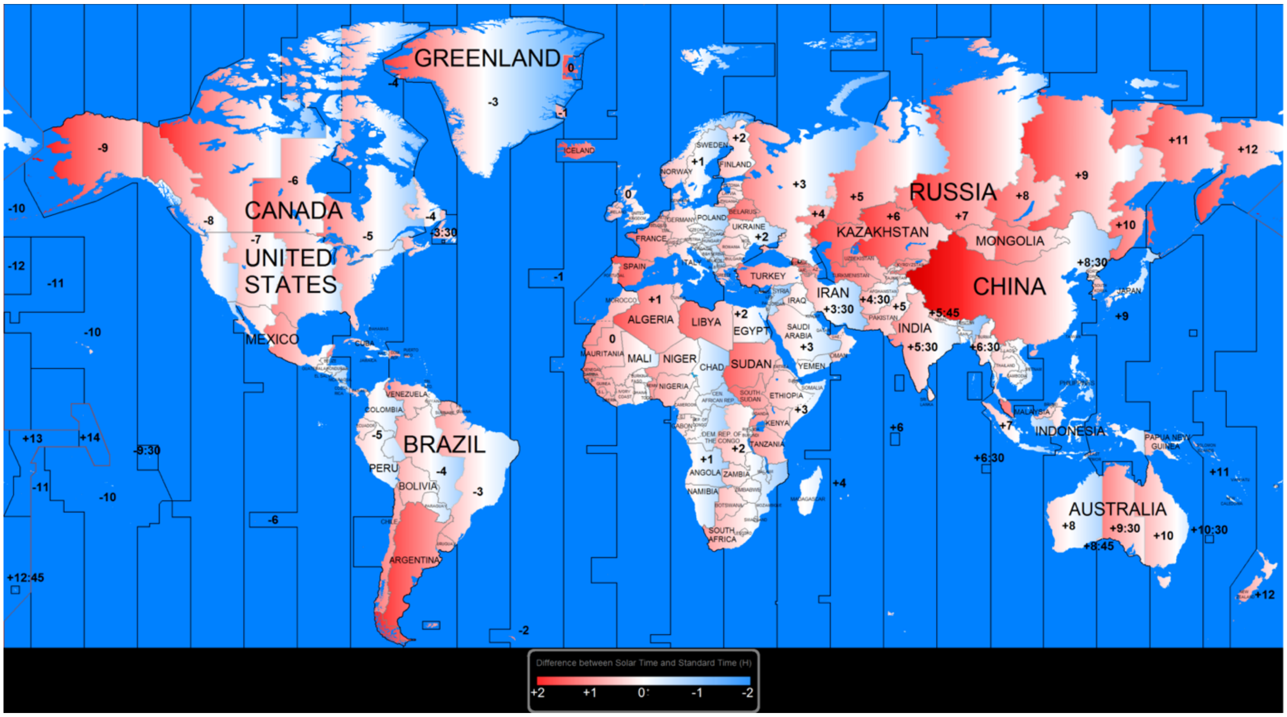
Britain accepted GMT as the standard time zone on August 2nd, 1880; even though it came up with GMT. Guess we know why they're not winning the world cup anytime soon!

Last country to adopt a standard time zone.
By 1929, most major countries had adopted hourly time zones. **Nepal** was the last country to adopt a standard offset, shifting slightly to UTC+5:45 in 1986.

Modern times

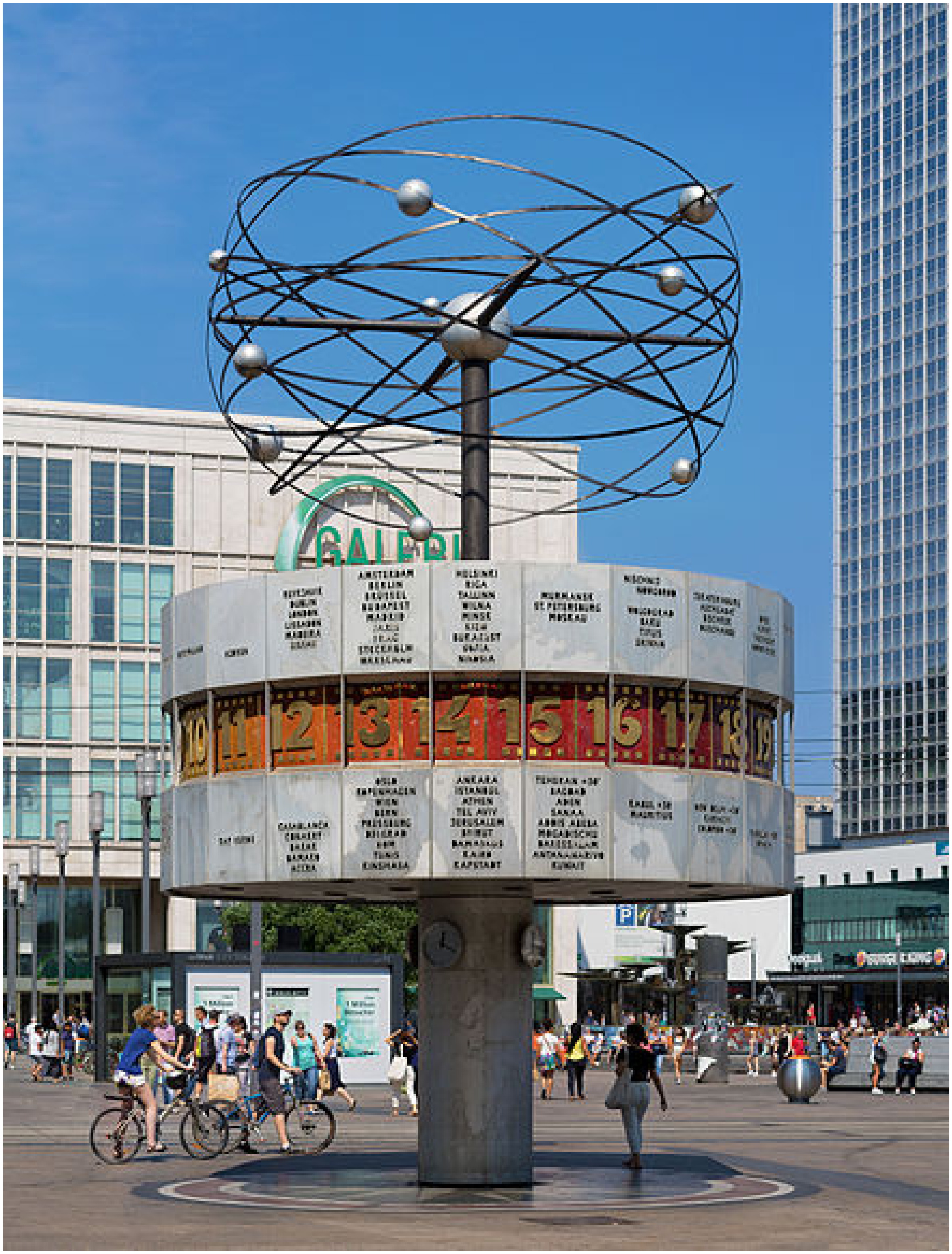


Time zones of today (circa 2016).



Difference between Solar and official time.

World clock



The *Weltzeituhr* (World Time Clock) at Alexanderplatz in Berlin

Computers

Computers and applications usually deal with UTC internally but show the user the local time. Most data is stored in UTC format.

Database

Records that include a timestamp typically use UTC, especially when the database is part of a system that spans multiple time zones. The use of local time for time-stamping records is not recommended for time zones that implement daylight saving time because once a year there is a one-hour period when local times are ambiguous.

JavaScript

There is very little in the way of time zone support for JavaScript. Essentially the programmer has to extract the UTC offset by instantiating a time object, getting a GMT time from it, and differencing the two. This does not provide a solution for daylight saving variations.

Unix time

Unix time (also known as **POSIX** time or **epoch** time) is a system for describing instants in time, defined as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970.

It does not handle leap seconds; thus, it is neither a linear representation of time nor a true representation of UTC.

The 32-bit representation of Unix time will end after the completion of 2,147,483,647 ((2^{31} - 1) seconds from the beginning(00:00:00 1 January 1970), i.e., on 19 January, 2038 03:14:08 GMT. This is referred to as the "Year 2038 problem" where the 32-bit Unix time will overflow and will take the actual count to negative.

The problem

The **Year 2038 problem** is an issue for computing and data storage situations in which time values are stored or calculated as a signed 32-bit integer, and this number is interpreted as the number of seconds since 00:00:00 UTC on 1 January 1970 (the epoch).[1] Such implementations cannot encode times after 03:14:07 UTC on 19 January 2038, a problem similar to but not entirely analogous to the Y2K problem (also known as the Millennium Bug), in which 2-digit values representing the number of years since 1900 could not encode the year 2000 or later. Most 32-bit Unix-like systems store and manipulate time in this Unix time format, so the year 2038 problem is sometimes referred to as the **Unix Millennium Bug** by association.

In Space

Orbiting spacecraft typically experience many sunrises and sunsets in a 24-hour period, or in the case of Apollo program astronauts travelling to the moon, none. Thus it is not possible to calibrate time zones with respect to the sun, and still respect a 24-hour sleep/wake cycle. A common practice for space exploration is to use the Earth-based time zone of the launch site or mission control. This keeps the sleeping cycles of the crew and controllers in sync.
The International Space Station normally uses Greenwich Mean Time (GMT).

Timekeeping on Mars can be more complex, since the planet has a solar day of approximately 24 hours and 39 minutes, known as a sol. Earth controllers for some Mars missions have synchronized their sleep/wake cycles with the Martian day, because solar-powered rover activity on the surface was tied to periods of light and dark. The difference in day length caused the sleep/wake cycles to slowly drift with respect to the day/night cycles on Earth, repeating approximately once every 36 days.

Time Zone Vs Offset

A **time zone** is a geographical region in which residents observe the same standard **time.** A **time offset** is an amount of **time** subtracted from or added to Coordinated Universal **Time** (UTC) **time** to get the current civil **time**, whether it is standard **time** or daylight saving **time** (DST).

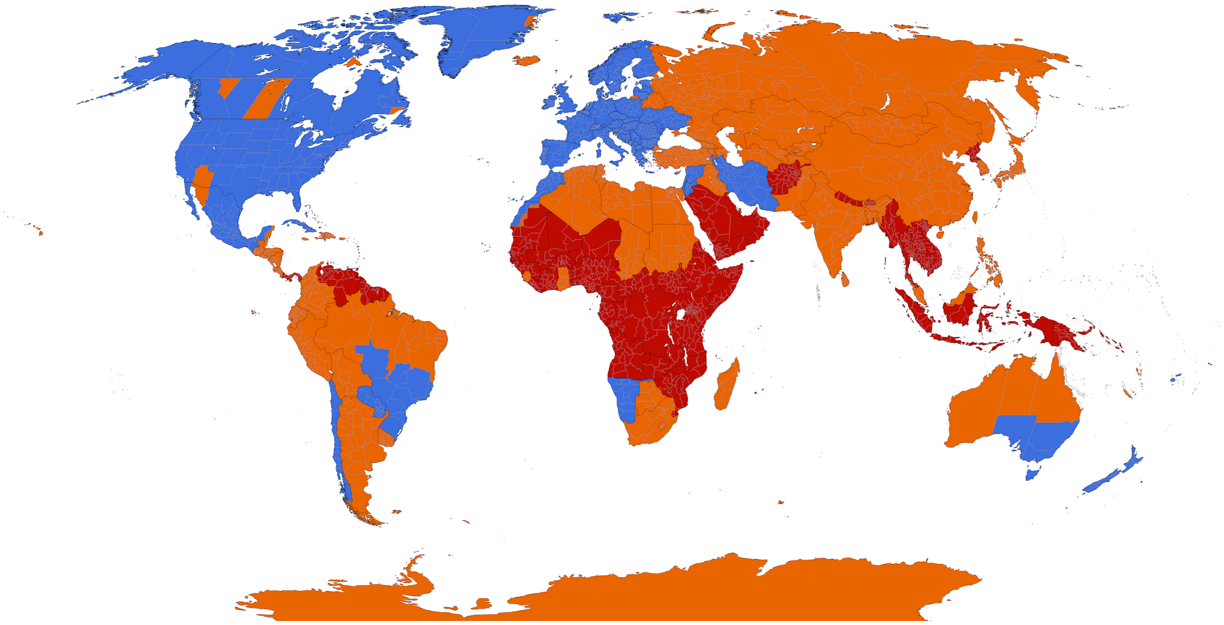
Jet Lag

Jet lag, medically referred to as **desynchronosis** and rarely as **circadian dysrhythmia**, is a physiological condition which results from alterations to the body's circadian rhythms resulting from rapid long-distance trans-meridian (east–west or west–east) travel on high-speed aircraft. For example, someone travelling from New York to London feels as if the time were five hours earlier than local time.

The term "jet lag" is used because before the arrival of passenger jet aircraft, it was uncommon to travel far and fast enough to cause jet lag. Travel by propeller-driven aircraft, by ship or by train was slower and of more limited distance than jet flights, and thus did not contribute widely to the problem.

Daylight Saving Time

Several regions in North America, Europe and Australia use daylight saving time (DST). The UTC offset during observation of DST is typically obtained by adding one hour to standard time. Central European Time UTC+01:00 is replaced by Central European Summer Time UTC+02:00, and Pacific Standard Time UTC−08:00 is replaced by Pacific Daylight Time UTC−07:00.



DST observed | DST formerly observed | DST never observed

UTC Offset

The **UTC offset** is the difference in hours and minutes from Coordinated Universal Time (UTC) for a particular place and date. It is generally shown in the format ±[hh]:[mm], ±[hh][mm], or ±[hh]. So if the time being described is one hour ahead of UTC (such as the time in Berlin during the winter), the UTC offset would be "+01:00", "+0100", or simply "+01".

Every inhabited place in the world has a UTC offset that is a multiple of 15 minutes, and the majority of offsets (as well as all nautical time zones) are measured in whole hours.

Libraries

Joda Time

http://www.joda.org/joda-time/ -
https://github.com/JodaOrg/joda-time
The standard date and time classes prior to Java SE 8 are poor. By tackling this problem head-on, Joda-Time became the de facto standard date and time library for Java prior to Java SE 8. **Note that from Java SE 8 onwards, users are asked to migrate to java.time (JSR-310) - a core part of the JDK which replaces this project.**
The design allows for multiple calendar systems, while still providing a simple API. The “default” calendar is the ISO8601 standard which is used by many other standards. The Gregorian, Julian, Buddhist, Coptic, Ethiopic and Islamic calendar systems are also included. Supporting classes include time zone, duration, format and parsing.

Noda time

http://nodatime.org/ - https://github.com/nodatime/nodatime
Noda Time is an alternative date and time API for .NET. It helps you to think about your data more clearly, and express operations on that data more precisely.

MomentJS

https://momentjs.com/timezone/ -
https://github.com/moment/moment-timezone/
Parse and display dates in any timezone.

References and more to read

1. https://en.wikipedia.org/wiki/Time_zone
2. https://en.wikipedia.org/wiki/Coordinated_Universal_Time
3. https://en.wikipedia.org/wiki/ISO_8601
4. https://en.wikipedia.org/wiki/Jet_lag
5. https://en.wikipedia.org/wiki/UTC_offset
6. https://momentjs.com/
7. http://www.joda.org/joda-time/
8. http://nodatime.org/
9. https://en.wikipedia.org/wiki/Lists_of_time_zones
10. https://en.wikipedia.org/wiki/World_clock
11. https://en.wikipedia.org/wiki/Year_2038_problem