

An hourglass with blue sand is positioned on a bed of dark, smooth pebbles. The hourglass is made of dark wood and has two glass bulbs. The sand is flowing from the top bulb to the bottom bulb. The background is a blurred, dark landscape.

Le temps c'est de l'argent ...
et des bugs aussi !





Nos partenaires et sponsors

Merci à eux !



GERAM-IT



Qui suis-je ?

Dev = j'écris du code

Software Craft = apprendre ensemble à faire bien



- gestion de personnel ferroviaire
- gestion de crise nucléaire
- appareils de mesure électrique
- suivi de travaux BTP
- gestion d'entrepôt logistique
- scanners à bagages aéroportuaires
- bornes de recharge de voiture électrique
- télémétrie de matériel médical
- analyse de trajets de camions routiers
- détection/surveillance de sites de phishing
- ...



Le temps

Un problème récurrent :

- toujours présent
- intrinsèque à toutes les activités humaines
- crucial dans de nombreux domaines
- si intuitif
- et pourtant si compliqué



Définition : le temps

Milieu indéfini et homogène dans lequel se situent les êtres et les choses et qui est caractérisé par sa double nature, à la fois continuité et succession. — cntrl.fr

Continuité indéfinie, milieu où se déroule la succession des évènements et des phénomènes, les changements, mouvements, et leur représentation dans la conscience. — dictionnaire.lerobert.com



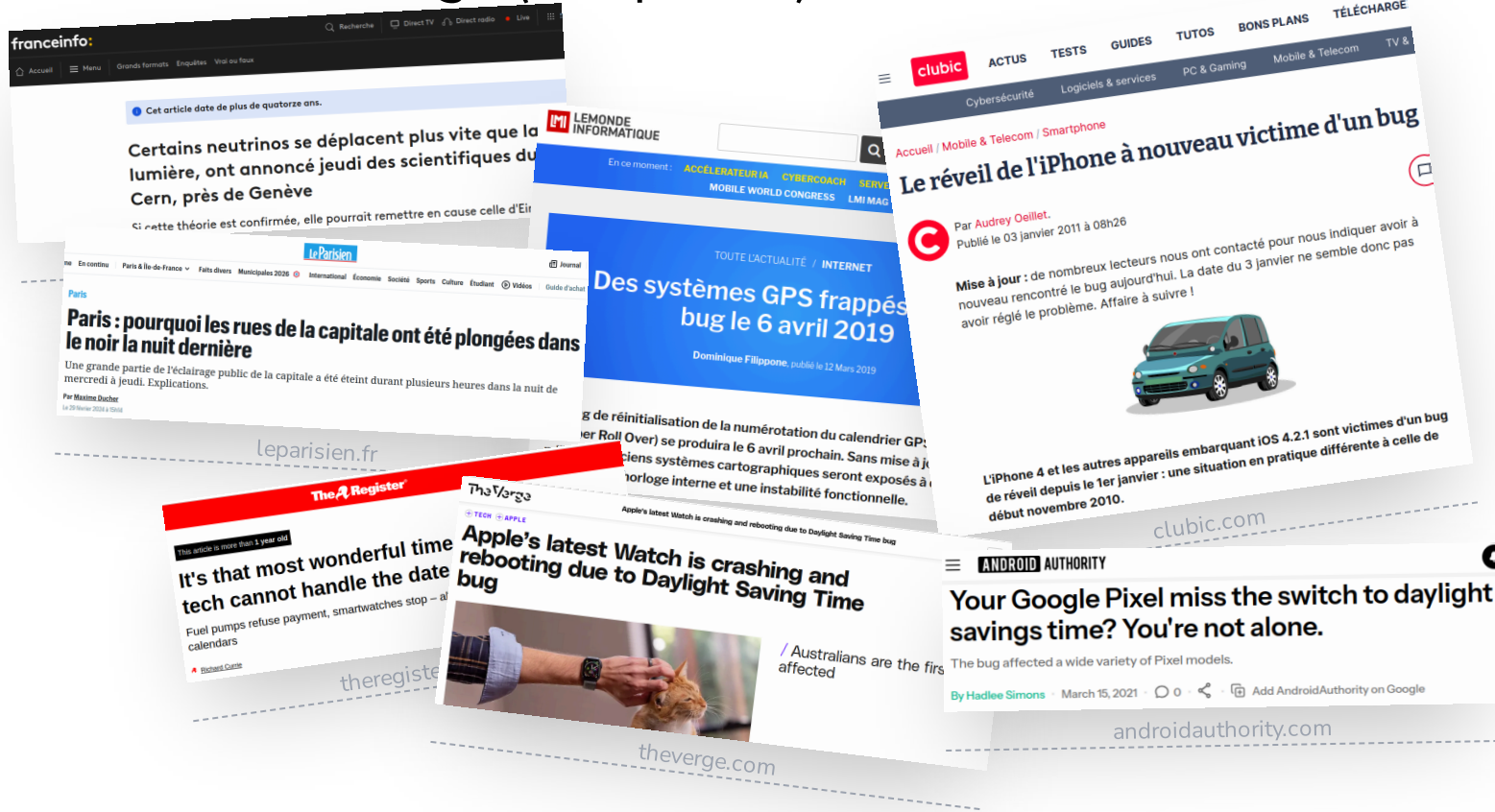
Ce qu'en dit la philo :

- opérationnaliste = c'est ce qu'on mesure, point
- substantialisme absolutiste = c'est compliqué

On va se passer de cette définition aujourd'hui ...

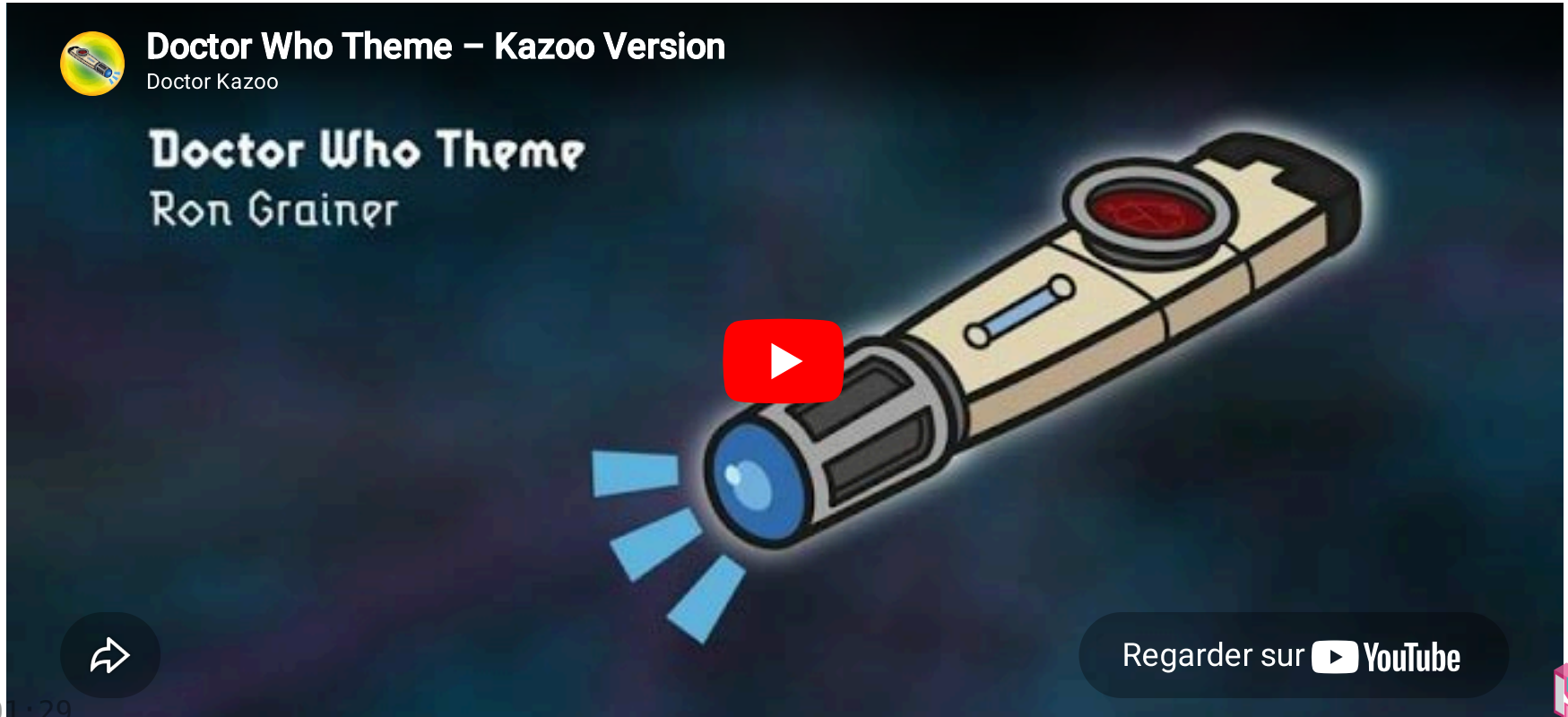


Définition : les bugs (temporels)




C'est parti pour un voyage ...

A travers le(s) temps !



Doctor Who Theme – Kazoo Version
Doctor Kazoo

Doctor Who Theme
Ron Grainer

Regarder sur  YouTube



Les problèmes commencent ...

- Heure de livraison : 02 h 30 (du matin), le Dimanche 29 Mars 2026
- Durée du trajet : 45 minutes + 1 heure !!!
- Heure de départ : 01 h 45
- Heure d'arrivée : 03 h 30 😨

Que s'est-il passé ? Décalage horaire : "daylight saving(s) time"



Avec du code

```
import datetime as dt

start_time = dt.datetime(2026, 3, 29, 1, 45, 0)
end_time = dt.datetime(2026, 3, 29, 3, 30, 0)
duration = end_time - start_time
print(duration)
# 1:45:00
```



Décalage horaire



Problèmes

- Non-continuité : DST



Et dans l'autre sens



Avec du code

```
# current_time=1:54:17 delta_time=15:00 delta_quantity=147 speed=9.8
# current_time=2:09:17 delta_time=15:00 delta_quantity=147 speed=9.8
# current_time=2:24:17 delta_time=15:00 delta_quantity=147 speed=9.8
# current_time=2:39:17 delta_time=15:00 delta_quantity=147 speed=9.8
# current_time=2:54:17 delta_time=15:00 delta_quantity=147 speed=9.8
# current_time=2:09:17 delta_time=-45:00 delta_quantity=147 speed=-2.66667
# current_time=2:24:17 delta_time=15:00 delta_quantity=147 speed=9.8
# current_time=2:39:17 delta_time=15:00 delta_quantity=147 speed=9.8
```



Problèmes

- Non-continuité : DST
- Non-unicité : DST
- Non-monotonicité : DST



Et encore dans l'autre sens



Problèmes

- Non-continuité : DST
- Non-unicité : DST
- Non-monotonicité : DST
- Non-homogénéité : DST

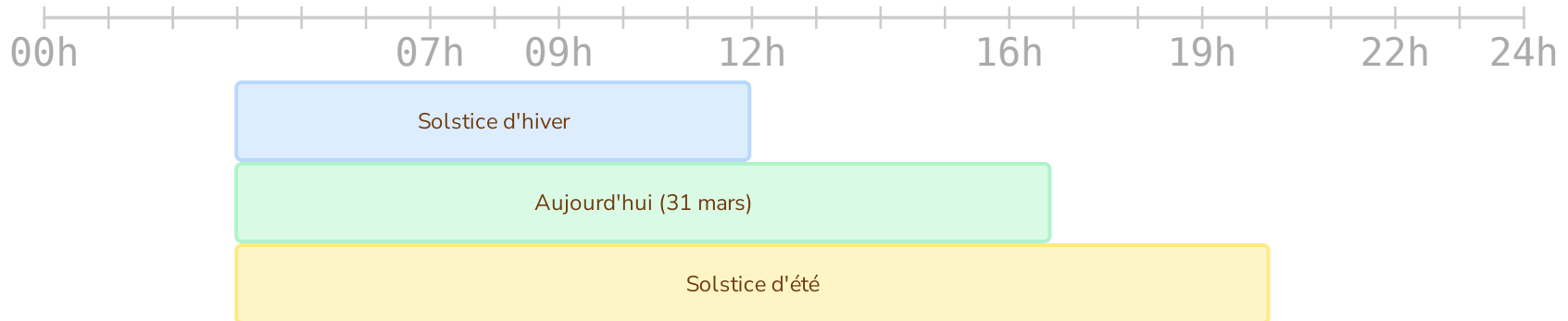


Pourquoi s'infliger ça ?

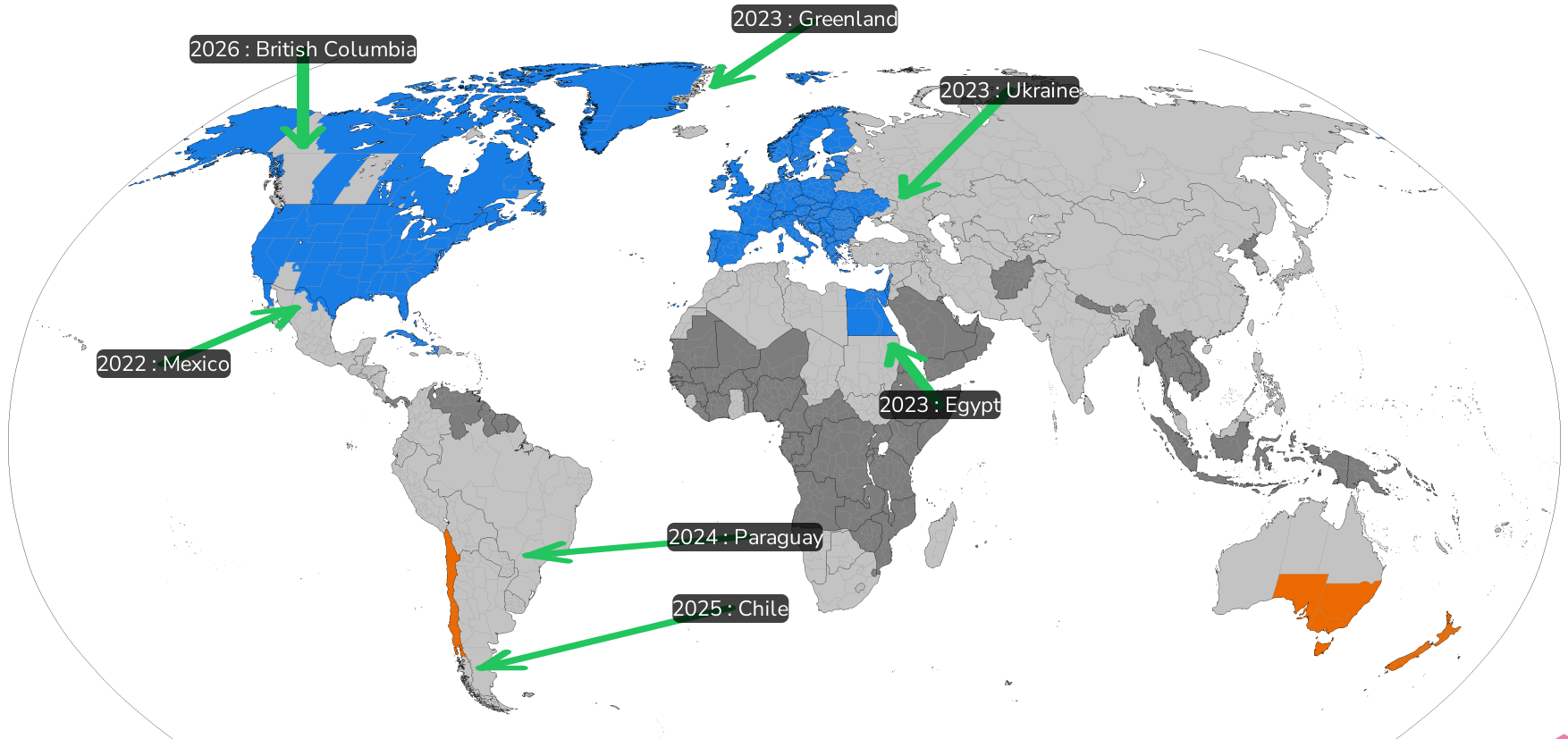
Réponse n°1 : 🗄️ 💰

Le changement d'heure a été instauré en France à la suite du choc pétrolier de 1973-1974. — [service-public.gouv.fr](https://www.service-public.gouv.fr)

Réponse n°2 : 🌅 🕒



Evolutions

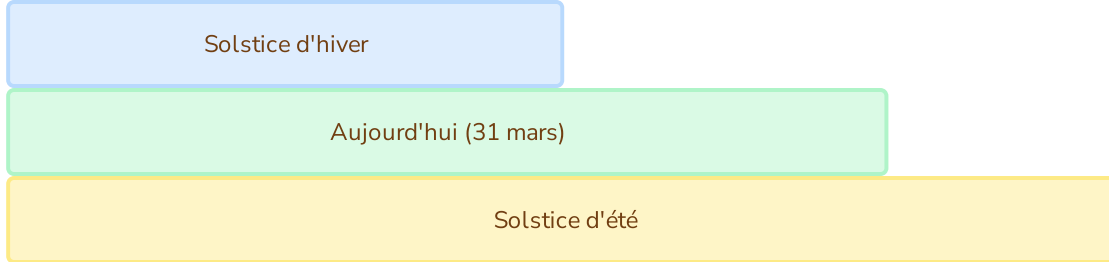


Problèmes

- Non-continuité : DST
- Non-unicité : DST
- Non-monotonie : DST
- Non-homogénéité : DST
- Non-constance : DST



Avez-vous remarqué ?



Le soleil, c'est compliqué

Vraiment très compliqué ...

```
today.replace(month=2) # ValueError: day is out of range for month  
# ne marche jamais
```



La périodicité naturelle

- le jour 🌞, la nuit 🌌🌙
- les 4 saisons astronomiques : 🌱🌻🍂🌳
- les saisons équatoriales: 🌧️🌞
- les années 📅



Oui, mais ...

- jour solaire ? civil ?
 - jour solaire : centré sur midi solaire (zénith)
 - aujourd'hui le 31 mars : 7:32 (ECST) --> 20:18 (ECST)
 - donc midi solaire à $(7:32+20:18)/2 = 13:55$ (ECST)
 - donc midi solaire à pas du tout "midi"
- année tropicale ? sidérale ? civile ?
 - tropicale (retour du soleil au même point dans le ciel) : ~365.24219 jours "éphémérides" (2000)
 - sidérale (retour de la Terre au même point sur son orbite) : 365.256363 jours (2025)
 - différence de ~20 minutes et 24.7 seconds
 - anodin ?



Une histoire de chocolat ?

- Jésus-Christ a ressuscité [\[citation needed\]](#)
 - les chrétiens ont souhaité commérer cette date
 - 1er dimanche après la pleine lune après le 21 Mars (approximation de l'équinoxe de Mars)
 - date mouvante = 🤪
- le calendrier Julien supposait qu'une année faisait exactement 365.25 jours
 - année tropicale ≈ 365.2422
 - les dates dérivent : le 21 Mars ne colle plus à l'équinoxe !
 - 1545 : on souhaite revenir aux dates de 325
- calendrier Grégorien !
 - transition : 10 jours sautés d'un coup

JULIEN	SEP 30	OCT 1	OCT 2	OCT 3	OCT 4	OCT 5	OCT 6	OCT 7	OCT 8
GRÉGORIEN	OCT 10	OCT 11	OCT 12	OCT 13	OCT 14	OCT 15	OCT 16	OCT 17	OCT 18

- année = 365.2425
- année = 365 ou 366 jours ?



Quiz !

1. Règle pour les années bissextiles ?


- divisible par 4
- mais pas par 100
- mais quand même par 400

2. Pourquoi "bissextile" ?

- car dans le calendrier Julien il s'insérait après le 24 Février

- ante diem bis sextum Kalendas Martias

- le sixième jour bis avant les kalendes (le premier jour) de mars

 En anglais : "leap years"

3. Année civile basée sur : tropicale ou sidérale ?

- tropicale
- on aime les équinoxes constants à travers les siècles (🤔)









Problèmes

- Non-continuité : DST
- Non-unicité : DST
- Non-monotonicité : DST
- Non-homogénéité : DST
- Non-constance : DST, *leap years*





Les constantes qui ne le sont pas

-  Musulman (Hijri) :
 - lunisolaire ("basé sur la lune")
 - la lune n'a pas une période synodique de X jours exactement (~29.53)
 - année de 12 mois lunaires : 354 ou 355 jours
 - $(6 \times 29 + 6 \times 30 = 354 ; 5 \times 29 + 7 \times 30 = 355)$
 - les saisons tournent sur 33 ans
-  Juif :
 - mois lunaires "intercalaires" pour approximer l'année tropicale
 - un mois supplémentaire tous les 2 ou 3 ans
 - cycle métonique : 7 mois en 19 ans
-  Orthodoxe :
 - resté sur le calendrier Julien
 - 13 jours de retard (non-rattrapés par le passage au Grégorien)
 - règle des bisextiles Julien : simplement divisible par 4
- calendriers  bouddhiste,  chinois,  hindou, ...

Le calendrier  grégorien est le standard dans l'industrie occidentale.



Problèmes

- Non-continuité : DST
- Non-unicité : DST
- Non-monotonicité : DST
- Non-homogénéité : DST, **leap years**
- Non-constance : DST, leap years



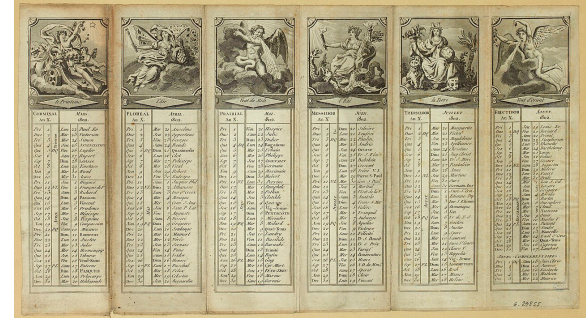
Des solutions ?

1st	January	February	March
	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
	8 9 10 11 12 13 14	8 9 10 11 12 13 14	8 9 10 11 12 13 14
	15 16 17 18 19 20 21	15 16 17 18 19 20 21	15 16 17 18 19 20 21
	22 23 24 25 26 27 28	22 23 24 25 26 27 28	22 23 24 25 26 27 28
29 30 31	31	31	
2nd	April	May	June
	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
	8 9 10 11 12 13 14	8 9 10 11 12 13 14	8 9 10 11 12 13 14
	15 16 17 18 19 20 21	15 16 17 18 19 20 21	15 16 17 18 19 20 21
	22 23 24 25 26 27 28	22 23 24 25 26 27 28	22 23 24 25 26 27 28
29 30	31	30	
3rd	July	August	September
	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
	8 9 10 11 12 13 14	8 9 10 11 12 13 14	8 9 10 11 12 13 14
	15 16 17 18 19 20 21	15 16 17 18 19 20 21	15 16 17 18 19 20 21
	22 23 24 25 26 27 28	22 23 24 25 26 27 28	22 23 24 25 26 27 28
29 30	31	30	
4th	October	November	December
	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
	8 9 10 11 12 13 14	8 9 10 11 12 13 14	8 9 10 11 12 13 14
	15 16 17 18 19 20 21	15 16 17 18 19 20 21	15 16 17 18 19 20 21
	22 23 24 25 26 27 28	22 23 24 25 26 27 28	22 23 24 25 26 27 28
29 30	31	31	

Symmetry 454

Semaine intercalaire !! 🤔

$$(52 \times \text{année} + 146) \bmod 293 < 52 \quad \text{🤔}$$



L'égalité ou rien !

12 mois de 30 jours ...

+ 5 jours épagomènes ("supplémentaires")
 "sansculottides" tous les ans
 +1 jour "sextile" tous les 6 ou 7 ans.

Bien d'autres ... mais **impossible de faire parfait.**



Quiz !

1. Quel jour et mois désignent `10/11/2025` ?
 - 10 novembre (mois n°11), format US `MM-DD-YYYY`
 - 11 octobre (mois n°10), format Européen
 - aucun moyen de savoir sans contexte
2. L'instant `14:01:29` est-il dans le passé ou le futur ?
 - dans le passé dans cette salle, car il est `15:01:29`
 - dans le futur à Rio de Janeiro, car il est `10:01:29`
 - aucun moyen de savoir sans contexte
3. Je dois me lever tôt ou me coucher tard pour une réunion à 9 heures ?
 - AM = matin ("ante meridiem")
 - PM = après-midi ("post meridiem")
 - aucun moyen de savoir sans contexte
4. Quel jour on sera "le mois prochain" ?
 - Mercredi 1er ? (premier jour)
 - Mardi 28 ? (même jour dans 4 semaines \approx 1 mois)
 - Jeudi 30 ? (même ordinal, ou presque)





La joie des standards

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)



PUBLIC SERVICE ANNOUNCEMENT:

OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.

THIS IS **THE** CORRECT WAY TO WRITE NUMERIC DATES:

2013-02-27

THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:

02/27/2013 02/27/13 27/02/2013 27/02/13
 20130227 2013.02.27 27.02.13 27-02-13
 27.2.13 2013.II.27. 27/2-13 2013.158904109
 MMXIII-II-XXVII MMXIII ^{LVII}/_{CCLXV} 1330300800
 ((3+3)×(111+1)-1)×3/3-1/3³ 2013
 10/1101/1101 02/27/20/13 0 1 2 3 4 5 6 7 8 miss



ISO-8601 : pas très accessible

- ISO 8601:1988
- ISO 8601:1988/COR 1:1991
- ISO 8601:2000
- ISO 8601:2004
- ISO 8601-1:2019 (basic rules)
- ISO 8601-2:2019 (extensions)
- ISO 8601-1:2019/Amd 1:2022
- ISO 8601-2:2019/Amd 1:2025

Only informative sections of standards are publicly available. To view the full content, you will need to purchase the standard by clicking on the "Buy" button. – [iso.org](https://www.iso.org)

CHF 181,00 ↔ EUR 199,15





ISO-8601 : de nombreux concepts

- années : `YYYY`
 - sachant que l'année `0000` n'existe pas dans l'ISO 8601
 - `YYY` est autorisé pour désigner une décade, `YY` pour une centaine, `Y` pour une dizaine
- dates calendaires : `YYYY-MM-DD` ou `YYYY-MM` ou `YYYY` (avec `T`)
- semaines de l'année : `YYYY-Www` ou `YYYYWww` (avec `T`)
 - la première de l'année contient le premier jour de la semaine
 - la semaine 01 de 2027 commencera le 04 janvier
- jours de semaine : `YYYY-Www-D` ou `YYYYWwwD` (avec `T`)
- dates ordinales : `YYY-DDD` ou `YYDDDD` (avec `DDI`)
- "times" (heures) : `T hh:mm:ss.sss` (ou sans `.sss`)
 - tellement de cas !
 - pas de limite au nombre de chiffres pour la partie décimale
- timezones : `Z` (prononcé "zulu") ou `±hh:mm` (ou `±hh`)
 - mais `-00` est interdit
 - depuis 2019 on peut ajouter `:ss` ou bien `:s`
- le `T` est obligatoire entre la partie "date" et la partie "time"
- durées : `PnYnMnDtnHnMnS` ou `PnW` ou `P<date>T<time>`
- intervalles : `<start>/<end>` ou `<start>/<duration>` ou `<duration>/<end>` ou juste `<duration>`
- répétitions : `Rn/<interval>` ou `R/<interval>`
- "Extended Date/Time Format" : incertitude, intervalles ouverts, exposants et chiffres significatifs, groupes de mois, dates multiples,

Current date and time expressed according to ISO 8601 [\[refresh\]](#)

Date in UTC	2026-03-31
Time in UTC	06:08:10Z T060810Z
Date and time in UTC	2026-03-31T06:08:10Z 20260331T060810Z
Date and time with the offset	2026-03-30T18:08:10-12:00 UTC-12:00 2026-03-31T06:08:10+00:00 UTC+00:00 2026-03-31T18:08:10+12:00 UTC+12:00
Week	2026-W14
Week with weekday	2026-W14-2
Ordinal date	2026-090

is")

ss.sss , ou .mmm , ou ...)

ne"





RFC 3339

- "Date and Time on the Internet: Timestamps"
- moins de versions, changements plus mineurs
- définitif depuis 2002
- accessible gratuitement
- censé être un "profil" de l'ISO-8601 (version 1988 et 2000)
- proche, mais différent de ISO-8601 :
 - divergence : timezone `-00` signifie "unknown"
 - divergence : un espace peut séparer la partie "date" de celle "time"
 - obligation : timezone offset
 - obligation : années sur 4 chiffres
- dans les faits, souvent confondus



Autres formats "standards"

- ISO/AWI 8601-1 : en cours de développement
- RFC 9557 : en cours de développement
- Unix timestamps : secondes écoulées depuis le 1er Janvier 1970 ("Epoch")
 - 1774962089
- Java timestamps : millisecondes écoulées depuis l'Epoch Unix
 - 1774962090505
- RFC (2)822 ("mail date") : format "human-readable"
 - Tue, 31 Mar 2026 15:01:29 +0200
- RFC 7231 ("http date") : basé sur RFC 2822, mais toujours en GMT
 - Tue, 31 Mar 2026 13:01:29 GMT
- CNS 7648 : adaptation taïwanaise d'ISO 8601 au calendrier ROC (Grégorien - 1911)
- ...



Problèmes

- Non-continuité : DST
- Non-unicité : DST
- Non-monotonie : DST
- Non-homogénéité : DST, leap years
- Non-constance : DST, leap years
- Non-standardized : dates and time formats



Le petit détail ...

C'est quoi "GMT" ?

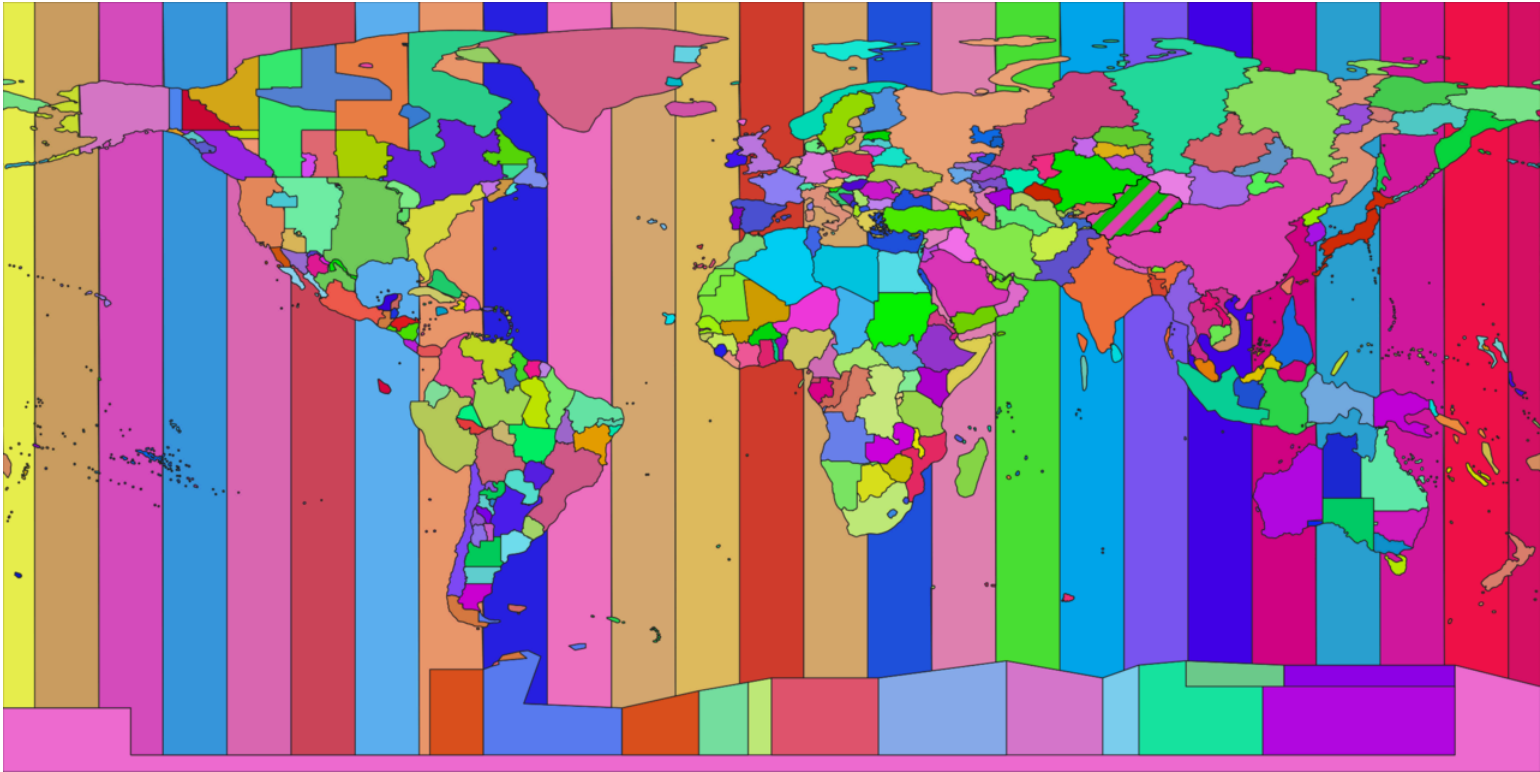


L'histoire des timezones

Pour faire (très) court :

- avant, le temps était très local
 - on suivait principalement le soleil
 - chaque ville avait son heure
 - on voyageait lentement
- puis les trains ont tout chamboulé
 - (les accidents mortels ont contribué)
- on a décidé d'unifier le temps, progressivement
- (je passe l'histoire chaotique pour chaque pays)
- aujourd'hui, le monde entier est découpé en timezones clairement définies





Bases et exemples

- "prime meridian" (IERS Reference Meridian) : Greenwich
 - (je vous passe l'histoire de Greenwich)
 - a servi à définir UTC
 - (je vous passe l'histoire d'UTC)
 - UTC = Universel Temps Coordonné / Universal Time Coordinated
- changement d'heure :
 - avant le 29 mars : European Central Time (ECT, UTC+1)
 - après le 29 mars : European Central Summer Time (ECST, UTC+2)
- voyage au Royaume-Uni :
 - UK suit le Greenwich Mean Time (GMT, UTC) ou le British Summer Time (BST, UTC+1)
 - a conservé son changement d'heure synchronisé avec le reste de l'Union Européenne
 - à l'aller je "perds" 1h, au retour je la regagne
- "offset" versus "timezone" :
 - China Standard Time (CST) = +08:00 ou Asia/Beijing
 - Eastern Standard Time (EST) = -05:00 ou America/Cancún
- ni offset ni timezone = "naïve"





Un sacré bazar !

- maintenu par l'IANA ("Internet Assigned Numbers Authority", filiale de l'ICANN), aussi connue pour :
 - les DNS root zones
 - les plage d'IP addresses
 - les AS numbers
 - servir de registry pour certains protocoles (port numbers, HTTP status codes, TLS cypher suites, DNS root zones, jeux de caractères pour `Content-Type` UTF-8, ISO-8859-1, ...)
- "IANA Time Zone Database" ou "tz" ou "tzinfo"
 - format riche
 - données historiques
 - mise-à-jour rapide
- inclus dans les OS
- inclus dans les libs des langages de programmation
- à mettre à jour régulièrement



Problèmes

- Non-continuité : DST
- Non-unicité : DST
- Non-monotonicité : DST
- Non-homogénéité : DST, leap years
- Non-constance : DST, leap years, **timezones**
- Non-standardized : dates and time formats



Problèmes

- Non-continuité : DST, **timezones**
- Non-unicité : DST, **timezones**
- Non-monotonicité : DST, **timezones**
- Non-homogénéité : DST, leap years, **timezones**
- Non-constance : DST, leap years, timezones
- Non-standardized : dates and time formats



Conseils

- c'est aussi important que l'unité dans une mesure
- jamais de datetime naïve !!
- privilégier UTC autant que possible
- ou sinon, bien savoir ce qu'on fait



Petit bonus : décalage solaire



Sans transition, les secondes !

- 15:01:29
- rien de plus simple : une toutes les secondes
- wait ...
- la seconde c'est l'unité SI de référence du temps
- la durée d'un nombre entier d'oscillations (9 192 631 770 exactement) liées à la fréquence de transition hyperfine de l'atome de césium 133 – [wikipedia.org](https://fr.wikipedia.org/wiki/Seconde)
- mesuré à partir d'horloges "atomiques"
- on obtient le TAI : Temps Atomique International
- → quelle différence avec UTC ? (Universel Temps Coordonné)





Le retour de la cosmologie !

- il faut d'abord parler de UT(1), et donc de UT0 ...
- à la base il y avait "GMT", différent de "GMT" (UTC+0)
- utilisé par les astronomes : chaque jour va de midi à midi, pour bosser la nuit
- transition à un calendrier minuit-minuit :
Midi 31/12/1925 des astronomes devint minuit 31/12/1925
- renommé en UT
- devenu UT0 quand on a introduit UT1
- (UT1 prend en compte une correction du mouvement polaire)
- (je passe sur UT2)
- UT1 sert à calculer UTC



Et du coup c'est quoi une seconde UTC ?

- une seconde UT1 une fraction (1/86400) de la durée de rotation entre deux zéniths
- mais la Terre n'a pas une rotation constante
(forces de marées, convection du noyau, fonte glaciaire, gros séismes)
- donc les secondes UT1 n'ont pas une durée constante par rapport à TAI
- or les secondes UTC sont des secondes TAI
- et UTC doit garantir $|\text{UTC} - \text{UT1}| < 0.9$
- donc c'est le nombre de secondes par jour qui change !
- 86400 ± 1
- seconde intercalaire ! ("leap second")



Et comment ça fonctionne ?

Ca dépend !

1. Ajout d'une seconde :



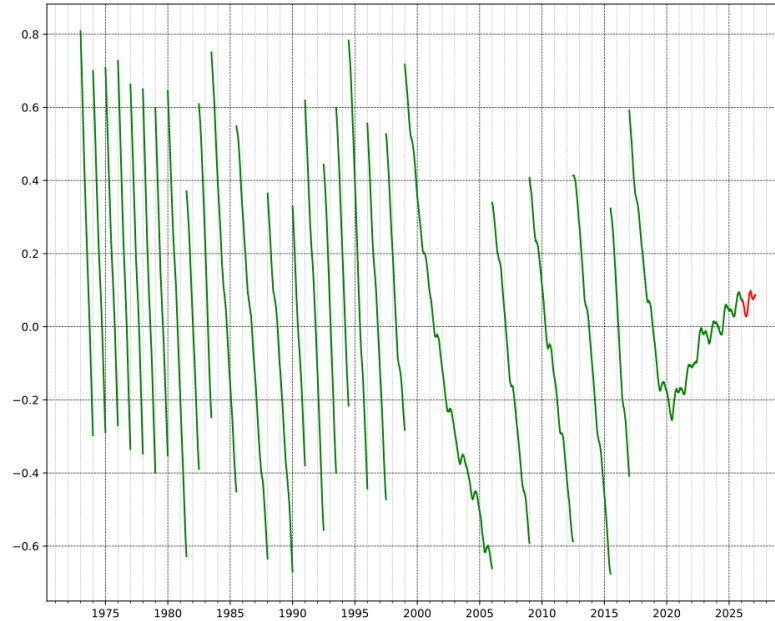
2. Smearing d'une seconde :



3. Répétition d'une seconde :



Résultat



Bientôt la toute première seconde intercalaire négative ?

Abolition en 2035 ? Offset fixe entre TAI et UTC



Problèmes

- Non-continuité : DST, timezones, **leap seconds**
- Non-unicité : DST, timezones, **leap seconds**
- Non-monotonicité : DST, timezones, **leap seconds**
- Non-homogénéité : DST, leap years, timezones, **leap seconds**
- Non-constance : DST, leap years, timezones, **leap seconds**
- Non-standardized : dates and time formats, **leap seconds**



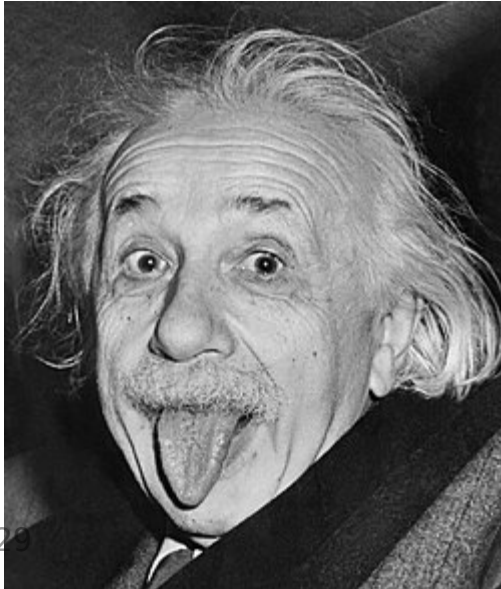
Comment être toujours à l'heure ?

- appeler "~~l'horloge parlante~~" (ah non, plus depuis 2022)
- avoir sa propre horloge atomique (📡)
- avoir une montre mécanique ou au quartz
- calendrier solaire
- se brancher à un serveur NTP ("network time protocol")
OS, FAI, Français, ...
- se brancher à un serveur PTP ("precise time protocol")
- monter son propre serveur (stratum)
- écouter la fréquence radio adéquate (SFTS : Standard frequency and time signal service)
- écouter les signaux GPS
- ...



Le GPS c'est compliqué !

- ce sont des horloges atomiques en orbite autour de la terre
- elles signalent leur position orbitale, et leur "GPS Time" :
nombre de secondes atomiques depuis le 6 Janvier, 1980, 00:00:00 UTC
- atomiques et pas UTC, donc il y a un décalage entre les deux !
- et il s'agit de quadrianguler la position du récepteur dans l'espace/temps 🤪





Tout est relatif !

- la relativité restreinte : référentiels inertiels (sans gravité)
 - espace-temps
 - paradoxe des jumeaux
 - satellites GPS ou ISS
 - Interstellar (spoiler !)
- la relativité générale : référentiels en accélération (gravité)
 - espace-temps courbé
 - lignes droites hyperboliques dans l'espace
 - vitesse constante dans l'espace-temps
 - satellites GPS (45 $\mu\text{s}/\text{jour}$) ou dans l'ISS
 - Interstellar (spoiler !)
- voyage dans le temps :
 - uniquement vers le futur, plus ou moins vite que les autres

15:01:29 ■ Oleg Kononenko (1111 jours) : a veilli 0.02+ secondes de moins



Problèmes

- Non-continuité : DST, timezones, leap seconds
- Non-unicité : DST, timezones, leap seconds
- Non-monotonicité : DST, timezones, leap seconds
- Non-homogénéité : DST, leap years, timezones, leap seconds, **general and special relativity**
- Non-constance : DST, leap years, timezones, leap seconds
- Non-standardized : dates and time formats





Il n'y a pas que la Terre

- Temps Lunaire
- Temps martien
- ...





Les problèmes continuent

- Pourquoi il y a 2 dates de début du Ramadan en 2026 ?
- le mois commence avec l'apparition dans le ciel du premier quartier de lune
- l'observation peut donc différer d'un jour ou deux selon la longitude d'observation
- ► Détails des difficultés, d'après Wikipedia
- donc désaccords entre pays et communautés
- contribué aux recherches astronomiques par les scientifiques du monde arabe
- est-ce que l'observatoire d'Arabie Saoudite a plus raison que les autres ?
- la Grande Mosquée de Paris se base sur une méthode traditionnelle (mi-observation, mi-scientifique)
- alors que le Conseil français du culte musulman préfère une méthode seulement scientifique



Choix politiques

- Arrêter les secondes intercalaires
- Arrêter les DST
- Arrêter les changements brusques de timezones



Solution : data

Principe : éviter la perte d'information

1. toujours prendre en input un datetime AVEC timezone, refuser autrement
2. immédiatement convertir ce datetime en UTC, avant tout traitement
3. ne faire que des traitements en UTC
4. stocker le datetime, avec sa timezone originale à-côté
5. ne faire que des traitements en UTC
6. seulement au moment de l'affichage, convertir dans la timezone locale du user



Solution : data++

1. toujours prendre en input un datetime AVEC timezone, refuser autrement
2. immédiatement convertir ce datetime en UTC, avant tout traitement, **et noter la version IANA utilisée**
3. ne faire que des traitements en UTC
4. stocker le datetime, avec sa timezone originale à côté, **et la version IANA correspondante**
et à chaque changement de version IANA, re-convertir le datetime UTC vers sa nouvelle valeur
5. ne faire que des traitements en UTC
6. seulement au moment de l'affichage, convertir dans la timezone locale du user



Solution : outillage

Temporal ... learned from the many failures and dead bodies that came before it. And it had lots of good implementations to look at:
 Joda Time, Chrono, etc. – news.ycombinator.com

- JavaScript
 - ⚠ danger ! (jsdate.wtf)
 - utiliser [Temporal \(TC39\)](https://tc39.github.io/Temporal/) ([CanIUse](https://caniuse.com/Temporal) ou polyfills), ou sinon Moment.js, Luxon, js-joda, ...
- Rust crates : std chrono+time, datettime, jiff, speedate, hifitime, ...
- Python : ~~stdlib datetime~~, stdlib zoneinfo (3.9), Whenever, Pendulum, Arrow, pytz, ...
- Java : ~~java.util~~, java.time, Joda, ~~Time4j~~
- C++ : Chrono
- .Net : Noda (cf Joda)
- libs spécifiques pour l'astronomie (Astropy), l'histoire, ...
- [ICU framework](https://icu4c.org/) (Chrome, Firefox, Qt, Android, MacOS/iOS, ...)
- Ruby : std time, Rails time
- comportement par défaut pour les DBs (Mongo renvoie des naïve par défaut, SQL `WITH TIMEZONE`)



Test

- connaître les risques ✓
- les IAs font très souvent des erreurs basiques
(dépend de la qualité pré-existante de la codebase)
- mais sont très pratiques pour :
 - générer des tests de couverture
 - brainstormer des test cases
 - générer des tests property-based
- `freezetime` et autres
- `libfaketime` (syscall-level) via `LD_PRELOAD`



Conclusion

- déterminer le temps dans le passé n'est pas simple
- déterminer si deux instants passés sont simultanés n'est pas simple
- déterminer dans combien de temps se produira un instant futur est impossible
- en fait, le temps est toujours très délicat à manipuler
- complexité essentielle méconnue
- complexité accidentelle méconnue
- si vous voulez vous entraîner : FalsehoodsAboutTime.com
 - [Falsehoods programmers believe about time \(34\)](#)
 - [More falsehoods programmers believe about time; "wisdom of the crowd" edition \(79\)](#)



Remerciements

- Meetup QA de Grenoble (et Rachel en particulier)
- Claude
- Slidev.js
- Petter Salminen - It's All About Time 🕒 - What developers need to know about time
- Le personnel du bar "Le Not a Beer"



Crédits photo

- Photo of a croissant by [personalgraphic.com](https://www.unsplash.com) on [Unsplash](https://www.unsplash.com)
- Images des évolutions des DST par Wikipedia : [DST_Countries_Map.png](#)
- Noël en Australie, © [myflag.com.au](https://www.myflag.com.au)
- Image du calendrier Symmetry454 par Wikipedia : [page Symmetry454](#)
- Image du calendrier républicain par Wikimedia : [Deuxième partie d'un calendrier de 1801-1802, avec les correspondances grégoriennes, orné de signes allégoriques, G.29855.jpg](#)
- Image de multiples calendriers par Wikipedia : [Gregorian calendar](#)
- Image des timezones par Wikipedia : [Timezone-boundary-builder release 2023d.png](#)
- Image des timezones de l'Arizona par : [travelness.com](https://www.travelness.com)
- Image des timezones autour de l'International date line par Wikipedia : [International Date Line.png](#)
- Image du décalage entre UTC et heure solaire par [24timezones.com](https://www.24timezones.com)
- Image du décalage entre UT1 et UTC par Wikipedia : [Leapsecond.ut1-utc.svg](#)
- Photo d'Albert Einstein par Wikimedia : [Albert Einstein sticks his tongue 1951.jpg](#)



Ressources pour aller + loin

Culture

- [Erik Naggum - The Long, Painful History of Time](#) : très complet, surtout pour la vision historique
- [Revue du monde - Le futur n'a pas toujours existé](#) : vidéo qui démonte notre rapport au temps, très lié à notre époque/culture

Timezones

- [qntm - So You Want To Abolish Time Zones](#) : courte fiction sur les problèmes introduits par l'abolition des timezones
- [qntm - So You Want Continuous Time Zones](#) : courte fiction sur les problèmes causés par la multiplication des timezones
- ["Discussion" sur Hacker News des raisons de choisir une time zone ou une autre](#) : en résumé, un "political move" vis-à-vis de l'administration
- [Colin Eberhardt - Exploring 120 years of timezones](#) : récit et visualisations de l'évolution des timezones au cours du temps
- [Judah Levine - Everyday Time and Atomic Time: Part 4](#) : l'histoire de l'ancien GMT
- [Computerphile - The Problem with Time & Timezones](#) : vidéo un peu dramatique des problèmes de timezones
- [The Supreme Guide to World Time Zones](#) : très complet et avec de jolies images



Astronomie et astrophysique

- [Launch Pad Astronomy - The Sky Part 1: Local Sky and Alt-Az / Horizon Coordinates](#) : explication des coordonnées célestes Altitude-Azimuth + très jolie visualisation des mouvements célestes en 2nde partie
- [FloatHeadPhysics - "I finally understood why time even exists! \(My mind is blown\)"](#) : explication du "temps propre relativiste" plutôt claire
- [Le Sens of Wonder - Le voyage dans le temps est possible](#) : une réponse sérieuse (et décevante) à la question
- [ScienceClic - Que verrait-on en tombant dans un Trou Noir ?](#) : de magnifiques images pour comprendre les effets de la courbure de l'espace-temps
- [ScienceEtonnante - Le paradoxe des jumeaux : rajeunir à la vitesse de la lumière ?](#) : le célèbre paradoxe, expliqué assez clairement
- [Matthias Kreier - Relative time for GPS satellites](#) : des calculs détaillés pour comprendre les tailles d'effet

Tooling

- [Temporal](#) : propose un glossaire parfait, devenu standard
- [Arie Bovenberg - Ten Python datetime pitfalls, and what libraries are \(not\) doing about it](#) : des exemples de bugs causés par de mauvais outils, ayant mené à la création de `whenever`
- [BurntSushi - about leap seconds](#) : Why leap seconds wont be supported in jiff, and why they are complicated and not very useful, also some reading recommendations
- [Open Dataset des Années bissextiles sur gouv.fr](#) : la version officielle en France de si une année est bissextile ou pas



Questions et slides

[feedback sur Slido](#)



[code des slides sur GitHub](#)

