DEVELHOPE

#codeforimpact

Date and Time

DEVELHOPE

#codeforimpact

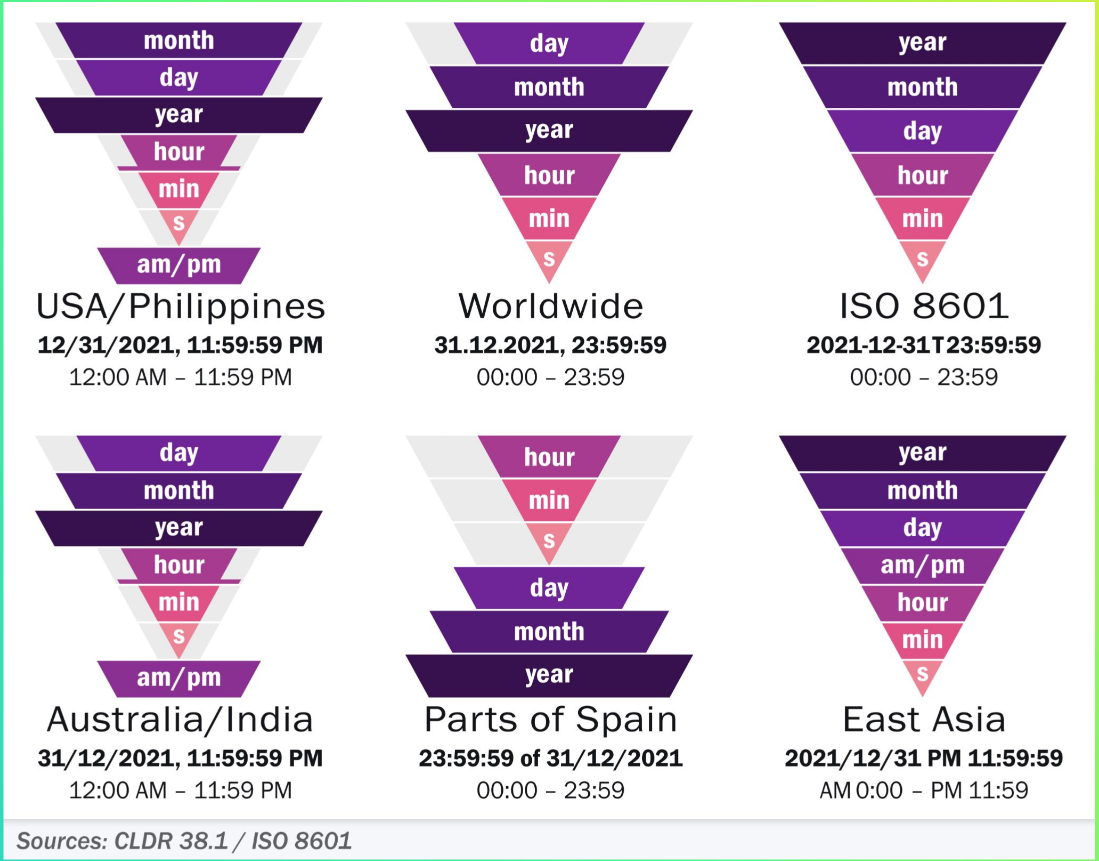
ISO Standard

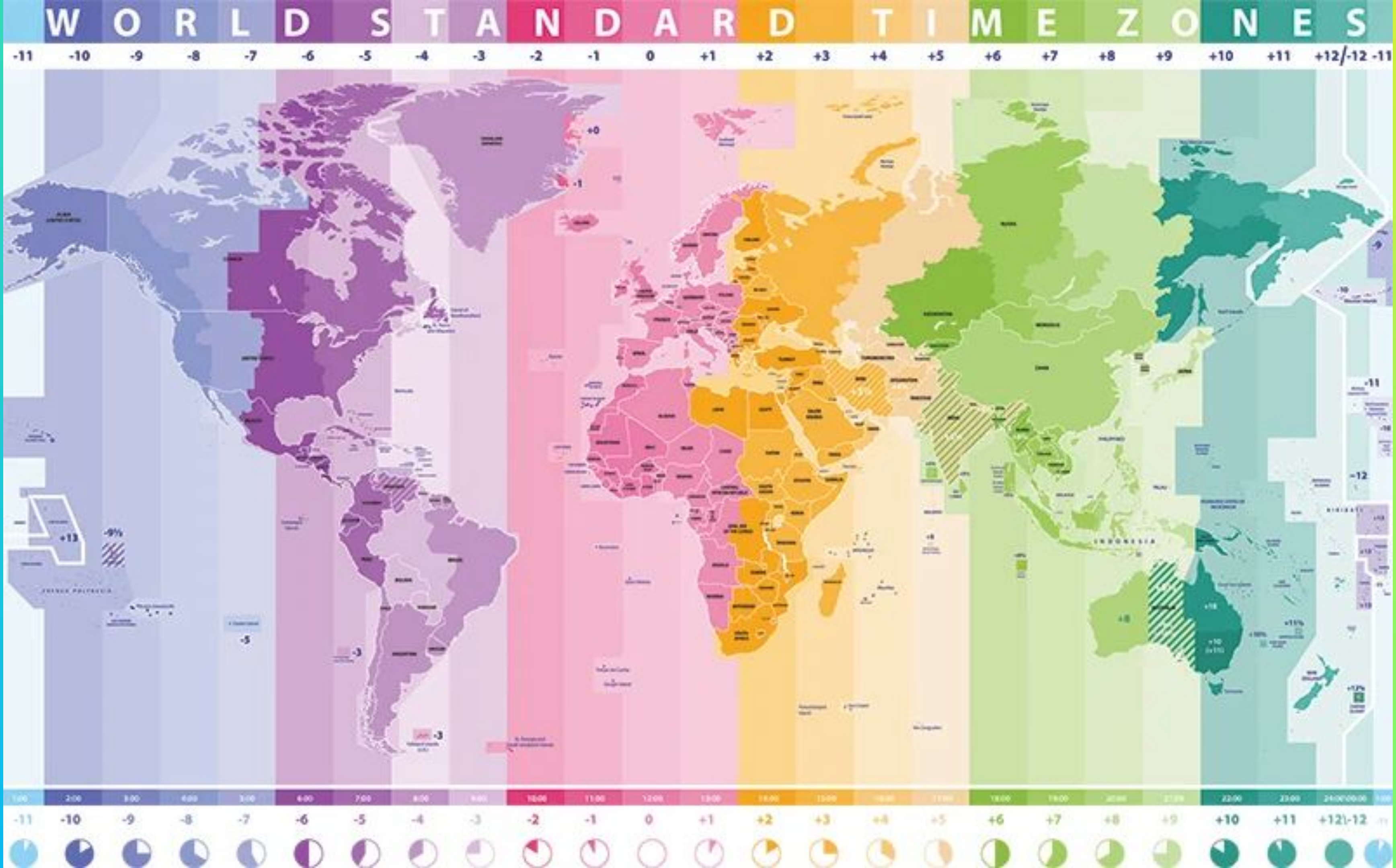
There is a standard format to write date times in all programming languages, it is

ISO 8601

YYYY-MM-DDTHH:mm:ss.mmm+ZZ:zz

https://it.wikipedia.org/wiki/ISO_8601




WORLD STANDARD TIME ZONES
-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11 +12/-12 -11
1:00 2:00 3:00 4:00 5:00 6:00 7:00 8:00 9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00 21:00 22:00 23:00 24:00/00:00 1:00
INDONESIA
FRENCH POLYNESIA
KIRIBATI
+13
-9½
-5
-3
+0
-1
+8
+10
+9½
+10½
+11
+12¾
-10
-11
-12

DEVELHOPE

#codeforimpact

Date and Time in Java

Java doesn't have built-in date and time features, so you can have them from the `java.time` package.

The following is a list of useful classes you could use:

- `java.time.LocalDate`
- `java.time.LocalTime`
- `java.time.LocalDateTime`
- `java.time.format.DateTimeFormatter`

java.time.LocalDate

The class `LocalDate` has a lot of methods for working with dates.

We will show just some of the most used methods in the example.

```
LocalDate todaysDate = LocalDate.now(); //  current date from the system clock

System.out.println(todaysDate);                     // prints the current date, e.g. 2021-11-07
System.out.println(todaysDate.getMonth());          // current month, e.g. NOVEMBER
System.out.println(todaysDate.getMonthValue());     // current month as a number, e.g. 11
System.out.println(todaysDate.getDayOfMonth());     // current day in the month, e.g. 7
System.out.println(todaysDate.getYear());           // current year, e.g. 2021
System.out.println(todaysDate.lengthOfMonth());     // in days, e.g. 30 for November
LocalDate oldDate = LocalDate.of(1990, 1, 8);       // aaaa-mm-dd
System.out.println(oldDate.isBefore(todaysDate));   // true, it's before
System.out.println(oldDate.isAfter(todaysDate));    // false
```

DEVELHOPE

#codeforimpact

java.time.LocalTime

The class `LocalTime` has a lot of methods for working with time.

We will show just some of the most used methods in the example.

```
LocalTime time = LocalTime.now(); //  current time from the system clock

System.out.println(time);                 // prints the current date, e.g. 09:30:04.215368
System.out.println(time.getHour());       // prints the current hour, e.g. 9
System.out.println(time.getMinute());     // prints the current minute, e.g. 30
System.out.println(time.getSecond());     // prints the current second, e.g. 4
System.out.println(time.getNano());       // prints the current nanosecond, e.g. 215368000

LocalTime previousTime = LocalTime.of(04, 12);
System.out.println(time.isAfter(previousTime));
System.out.println(time.isBefore(previousTime));
```

java.time.LocalDateTime

The class `LocalDateTime` has a lot of methods for working with date and time.

We will show just some of the most used methods in the example.

```
LocalDateTime time = LocalDateTime.now(); // current date and time from the system clock
System.out.println(time);                  // e.g. 2021-11-07T09:42:58.484911
System.out.println(time.getDayOfMonth());   // e.g. 7
System.out.println(time.getDayOfWeek());    // e.g. SUNDAY
System.out.println(time.getDayOfYear());    // e.g. 311
System.out.println(time.getMonthValue());   // e.g. 11
System.out.println(time.getMonth());        // e.g. NOVEMBER

LocalDateTime previousTime = LocalDateTime.of(1980, 12, 31, 14, 23); // aaaa-mm-dd hh-mm
System.out.println(previousTime.isBefore(time));      // true
```

DEVELHOPE

#codeforimpact

java.time.format.DateTimeFormatter

You probably noticed that `LocalDateTime.now()` returns a `LocalDateTime` that has a particular format, with a `T` in the middle to separate date from time: e.g. `2021-11-07`**`T`**`09:42:58.484911`.

It's not so great and you maybe want to have the date in `dd-MM-yyyy`, so the `DateTimeFormatter` gives you the ability to change the format of a `LocalDateTime`.

```
LocalDateTime time = LocalDateTime.now(); // current date and time from the system clock

System.out.println("Not formatted: " + time); // Not formatted: 2021-11-07T10:05:08.768634

DateTimeFormatter f1 = DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss");
DateTimeFormatter f2 = DateTimeFormatter.ofPattern("MM-dd-yyyy HH:mm");

System.out.println("f1 format: " + time.format(f1)); // f1 format: 07-11-2021 10:05:08
System.out.println("f2 format: " + time.format(f2)); // f2 format: 11-07-2021 10:05
```