Dates & Times

```
LocalDateTime.now(); //2023-05-14T19:56:29.738748
ZonedDateTime.now(); //2023-05-14T19:56:46.627004+02:00[Europe/Zuri
```

ZonedDateTime

```
2021-10-25T09:13:07.769-05:00[America/New_York]

2023-12-16T14:30:00.000-05:00[America/New_York]
```

GMT vs UTC

- GMT - Greenwich Mean Time
- UTC - Coordinated Universal Time

Duration

```
jshell> java.time.Duration.ofSeconds(758768437)
$54 ==> PT210769H37S
```

toString

PT24H : starts with **PT**

I can't use `Duration` with `LocalDate`!

```
var date = LocalDate.of(2023, 1, 21);
var days = Duration.ofDays(1);
// Unsupported unit: Seconds
//I can NOT use Duration with LocalDate
//Exception in thread "main" java.time.temporal.UnsupportedTemporal
System.out.println(date.plus(days));
```

Period

Usage Of Period

static vs instance

- `Period.of()`: static method (does not chain);
- `Period.plus()`: instance, does chain

```
Period oneDay = Period.ofDays(1);
//this is an instance method, so this I can chain
Period p2 = oneDay.plusDays(2);
//example of chaining
p2 = oneDay.plusYears(2).plusMonths(3).plusDays(5);
System.out.println(p2);  //P2Y3M6D
```

`Period.of()` does not chain:

```
//Only the last method is considered!
Period period = Period.ofYears(1).ofMonths(2).ofDays(1);
System.out.println(period);  //P1D
```

The `Period` class does not have methods such as `getSeconds()`, `getMinutes()`, `getHours()`.

toString

P1D : starts with **P** example:

```
P2Y3M6D
```

Instant

The Instant class represents a specific moment in time in the GMT time zone.

Instant

Format

```
YYYY-MM-DDTHH:mm:ss.SSSZ
```

Example:

```
2024-03-01T14:02:30.123Z
```

Creation

Only ZonedDateTime has the toInstant() method.

```
//from ZonedDateTime
Instant instantZRH = zdtZurich.toInstant();
//with now()
Instant now = Instant.now();  //2024-01-18T11:02:16.628552400Z
```

parse

```
Instant instant = Instant.parse("2024-03-01T14:02:30.00Z");
System.out.println(instant); //2024-03-01T14:02:30Z
```

plus

While an Instant represents a specific moment in time using GMT, Java only allows adding or removing units of DAYS or smaller.

```
Instant now = Instant.now();
Instant then = now.plus(1, ChronoUnit.DAYS); //    //2024-01-08T07:4

//adding a year - throws an exception
//Exception in thread "main" java.time.temporal.UnsupportedTemporalT
Instant inOneYear = now.plus(1, ChronoUnit.YEARS);
```

ChronoUnit

between

```
ZonedDateTime zdt1 = ZonedDateTime.of(date, time, zone);
ZonedDateTime zdt2 = zdt1.plus(1, ChronoUnit.HOURS);
long between = ChronoUnit.HOURS.between(zdt1, zdt2);
```

ChronoUnit