# LocalTime

Diferente das datas, podemos também ver as horas.

## Date Time Formatter

Você deve ter notado que os resultados estão em formatos que não estamos acostumados a trabalhar. Podemos então trabalhar com diversos formatadores de datas existentes!

Exemplos de formatos de data e hora:

| **Formato** | **Exemplo** |
| --- | --- |
| MM/dd/yyyy | 19/11/2022 |
| dd-M-yyyy hh:mm:ss | 19-11-22 12:07:23 |
| dd MMMM yyyy | 19 de novembro de 2022 |
| dd MMMM yyyy zzzz | 19 de novembro de 2022, Horário Padrão de Brasília |
| E, dd MMM yyyy HH:mm:ss z | Sáb, 19 de novembro de 2022, 12:07:23 BRT |

Além disso, também podemos mudar a forma de exibir o mês. Se utilizarmos letras em quantidades diferentes, consequentemente o resultado também é. Vejamos:

| **Tipo** | **Padrão** | **Exemplo** |
| --- | --- | --- |
| Mês completo | MMMM | Setembro |
| Mês abreviado | MMM | Set. |
| Mês numérico | MM | 09 |

# Formatando os dados

Para formatarmos, vamos utilizar o **DateTimeFormatter**. Existem diversos já prontos, mas há ainda a alternativa de você criar o seu próprio formato no padrão já conhecido de **dd/MM/yyyy hh:mm:ss**.

Para fazer isso basta você utilizar o método ofPattern. Como vimos nas tabelas anteriores, podemos escolher como será impresso.

**DateTimeFormatter** formatador = DateTimeFormatter.ofPattern("dd/MM/yyyy hh:mm:ss");

**public** **class** **Horarios** {

**public** **static** void **main**(String[] args) {

**LocalTime** hoje = **LocalTime**.**now**();

**System**.out.**println**(hoje);

}

}

SAIDA:

**06:02:21.029460400**

Agora, vamos criar um novo horário para representar as horas em que Alice nasceu:

**import** java.time.LocalTime;

**public** **class** **Horarios** {

**public** **static** **void** **main**(String[] args) {

**LocalTime** hoje = LocalTime.now();

System.out.println(hoje);

**LocalTime** aniversarioHoraAlice = LocalTime.of(22, 33, 15);

System.out.println(aniversarioHoraAlice);

}

}

O resultado será no formato hh:mm:ss, pois colocamos dessa forma no LocalTime.of() :

### Horário e data atual

O localDateTime representa uma data com a hora, visto como ano-mês-dia-hora-minuto-segundo. O tempo é representado como yyyy-MM-dd-HH-mm-ss.zzz.

**import** java.time.LocalDateTime;

**public** **class** **DataHorario** {

**public** **static** **void** **main**(String[] args) {

**LocalDateTime** hoje = LocalDateTime.now();

System.out.println(hoje);

}

}

**2022-11-19T04:11:17.108565200**