

Localization

Format java.time

There is a `DateTimeFormatter` class, but not a `DateFormatter`.

ofPattern

```
DateTimeFormatter dateTimeFormatter = DateTimeFormatter.ofPattern("d-MMMM-yyyy HH:mm:ss");
LocalDateTime now = LocalDateTime.now();
String time = now.format(dateTimeFormatter);
System.out.println(time); //22-09-2023 - 08:44:55
```

[format java time](#)

ofLoc

//TBC

DecimalFormat

- # Omit position if no digit exists for it
- 0 Put 0 in position if no digit exists for it.

```
double wealth = 500_000.45;
final String format = "###,###,###.000000";
NumberFormat nf = new DecimalFormat(format);
String result = nf.format(wealth);
System.out.println(result); //500,000.450000
```

[UsageOfDecimalFormat](#)

CompactNumberFormat

A `CompactNumberFormat` rounds to the nearest whole number within the **three-digit** range.

```
NumberFormat compactNumberInstance = NumberFormat.getCompactNumberInstance();
String format = compactNumberInstance.format(1_000_000);
System.out.println(format); //1M
```

```
NumberFormat.getCompactNumberInstance(Locale.getDefault(), Style.SHORT);
NumberFormat.getCompactNumberInstance(Locale.getDefault(), Style.LONG);
```

//either you pass both parameters or none

```
NumberFormat.getCompactNumberInstance();
```

Formatting Rules

- First it determines the highest range for the number, such as thousand (K), million (M), billion (B), or trillion (T).
- It then returns up to the first three digits of that range, rounding the last digit as needed.
- Finally, it prints an identifier. If SHORT is used, a symbol is returned. If LONG is used, a space followed by a word is returned.

Examples:

- 1_900_000 -> M -> (1),9->2M
- 23_021 -> K - (23),0 -> 23K
- 21_365_891 -> M (21),3 ->21M
- 123,5 - (N/A) -> (123),5 -> 124
- 960_010 -> K -> (960),0 -> 960K
- 0,123 -> (N/A) ->0

Currency

```
double amount = 12.3D;
Locale.setDefault(Locale.US);
Locale.setDefault(Category.FORMAT, Locale.ITALY);
Locale.setDefault(Category.DISPLAY, Locale.UK);
// For this reason, the ITALY locale is used to format the data with
System.out.print(NumberFormat.getCurrencyInstance()
    .format(amount)); //12,30 €
```

Formatting Currency

Locale

In Java, a locale can be represented by a language code in lowercase, or a language and country code, with language in lowercase and country in uppercase.

Locale	Valid
--------	-------

it	:white_check_mark:
IT_it	:x:
it_IT	:white_check_mark:
IT_IT	:x:
it_it	:x:
IT	:x: (missing language)

```
Local wrong_enUS = new Locale("en_US")); //wrong way to Load
Local right_enUS = new Locale("en", "US")); //proper way
```

Creation of Locale

Locale entities

Oracle defines a locale as a:

- geographical region
- political region
- cultural region

Locale Category

- **DISPLAY** : Category used for displaying data about locale
- **FORMAT** : Category used for formatting dates, numbers, or currencies

```
Locale.setDefault(Category.FORMAT, Locale.ITALY);
Locale.setDefault(Category.DISPLAY, Locale.UK);
```

Resource Bundle

- Dog.properties
- Dog_de.properties
- Dog_en.properties

```
Locale.setDefault(new Locale("en", "US"));
var b = ResourceBundle.getBundle("Dog", fr);
```

Java will use **Dog_en.properties** as the matching resource bundle. Since there is no match for French, the default locale is used.

Methods

```
String value = bundle.getString("grapes");  
Object mango = bundle.getObject("mango");  
String[] papayas = bundle.getStringArray("papayas");
```

MessageFormat

```
var helloByName=Hello, {0} and {1};  
var message = MessageFormat.format(format, "John", "Mark"));  
//Hello, John and Mark  
System.out.print(message);
```