

[Foundation](#) / `RelativeDateTimeFormatter`

## Class

# RelativeDateTimeFormatter

A formatter that creates locale-aware string representations of a relative date or time.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.1+ | macOS 10.15+ | tvOS 13.0+ | visionOS 1.0+ | watchOS 6.0+

```
class RelativeDateTimeFormatter
```

## Overview

Use the strings that the formatter produces, such as “1 hour ago”, “in 2 weeks”, “yesterday”, and “tomorrow” as standalone strings. Embedding them in other strings may not be grammatically correct.

## Topics

### Converting Dates to Formatted Strings

```
func localizedString(for: Date, relativeTo: Date) -> String
```

Formats the date interval from the reference date to the specified date using the formatter’s calendar.

```
func localizedString(from: DateComponents) -> String
```

Formats a relative time represented by the specified date components.

```
func localizedString(fromTimeInterval: TimeInterval) -> String
```

Formats the specified time interval using the formatter's calendar.

```
func string(for: Any?) -> String?
```

Creates a formatted string for a date relative to the current date and time.

## Configuring Formatter Options

```
var calendar: Calendar!
```

The calendar to use for formatting values that don't have an inherent calendar of their own.

```
var locale: Locale!
```

The locale to use when formatting the date.

```
var dateTimeStyle: RelativeDateTimeFormatter.DateTimeStyle
```

The style to use when describing a relative date, for example "yesterday" or "1 day ago".

```
enum DateTimeStyle
```

A type that represents the style to use when formatting relative dates, such as "1 week ago" or "last week".

```
var unitsStyle: RelativeDateTimeFormatter.UnitsStyle
```

The style to use when formatting the quantity or the name of the unit, such as "1 day ago" or "one day ago".

```
enum UnitsStyle
```

A type that represents the style to use when formatting the units of relative dates.

```
var formattingContext: Formatter.Context
```

A description of where the formatted string will appear, allowing the formatter to capitalize the output appropriately.

---

## Relationships

### Inherits From

Formatter

### Conforms To

CVarArg  
CustomDebugStringConvertible  
CustomStringConvertible  
Equatable  
Hashable  
NSCoding  
NSCopying  
NSObjectProtocol

---

## See Also

### Dates and times

`class DateFormatter`

A formatter that converts between dates and their textual representations.

`class DateComponentsFormatter`

A formatter that creates string representations of quantities of time.

`class DateIntervalFormatter`

A formatter that creates string representations of time intervals.

`class ISO8601DateFormatter`

A formatter that converts between dates and their ISO 8601 string representations.