Flutter resumen documentación

# text widgets

## default text style

The text style to apply to descendant [Text](https://api.flutter.dev/flutter/widgets/Text-class.html) widgets which don't have an explicit style.

### Constructors

[DefaultTextStyle](https://api.flutter.dev/flutter/widgets/DefaultTextStyle/DefaultTextStyle.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [TextStyle](https://api.flutter.dev/flutter/painting/TextStyle-class.html) style, [TextAlign](https://api.flutter.dev/flutter/dart-ui/TextAlign.html)? textAlign, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) softWrap = true, [TextOverflow](https://api.flutter.dev/flutter/painting/TextOverflow.html) overflow = TextOverflow.clip, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? maxLines, [TextWidthBasis](https://api.flutter.dev/flutter/painting/TextWidthBasis.html) textWidthBasis = TextWidthBasis.parent, [TextHeightBehavior](https://api.flutter.dev/flutter/dart-ui/TextHeightBehavior-class.html)? textHeightBehavior, required [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html) child})

Creates a default text style for the given subtree.

const

[DefaultTextStyle.fallback](https://api.flutter.dev/flutter/widgets/DefaultTextStyle/DefaultTextStyle.fallback.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key})

A const-constructable default text style that provides fallback values.

const

### Properties

[child](https://api.flutter.dev/flutter/widgets/ProxyWidget/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)

The widget below this widget in the tree.

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[maxLines](https://api.flutter.dev/flutter/widgets/DefaultTextStyle/maxLines.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)?

An optional maximum number of lines for the text to span, wrapping if necessary. If the text exceeds the given number of lines, it will be truncated according to [overflow](https://api.flutter.dev/flutter/widgets/DefaultTextStyle/overflow.html).

final

[overflow](https://api.flutter.dev/flutter/widgets/DefaultTextStyle/overflow.html) → [TextOverflow](https://api.flutter.dev/flutter/painting/TextOverflow.html)

How visual overflow should be handled.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[softWrap](https://api.flutter.dev/flutter/widgets/DefaultTextStyle/softWrap.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether the text should break at soft line breaks.

final

[style](https://api.flutter.dev/flutter/widgets/DefaultTextStyle/style.html) → [TextStyle](https://api.flutter.dev/flutter/painting/TextStyle-class.html)

The text style to apply.

final

[textAlign](https://api.flutter.dev/flutter/widgets/DefaultTextStyle/textAlign.html) → [TextAlign](https://api.flutter.dev/flutter/dart-ui/TextAlign.html)?

How each line of text in the Text widget should be aligned horizontally.

final

[textHeightBehavior](https://api.flutter.dev/flutter/widgets/DefaultTextStyle/textHeightBehavior.html) → [TextHeightBehavior](https://api.flutter.dev/flutter/dart-ui/TextHeightBehavior-class.html)?

Defines how to apply [TextStyle.height](https://api.flutter.dev/flutter/painting/TextStyle/height.html) over and under text.

final

[textWidthBasis](https://api.flutter.dev/flutter/widgets/DefaultTextStyle/textWidthBasis.html) → [TextWidthBasis](https://api.flutter.dev/flutter/painting/TextWidthBasis.html)

The strategy to use when calculating the width of the Text.

final

## rich text

A paragraph of rich text.

The [RichText](https://api.flutter.dev/flutter/widgets/RichText-class.html) widget displays text that uses multiple different styles. The text to display is described using a tree of [TextSpan](https://api.flutter.dev/flutter/painting/TextSpan-class.html) objects, each of which has an associated style that is used for that subtree. The text might break across multiple lines or might all be displayed on the same line depending on the layout constraints.

Text displayed in a [RichText](https://api.flutter.dev/flutter/widgets/RichText-class.html) widget must be explicitly styled. When picking which style to use, consider using [DefaultTextStyle.of](https://api.flutter.dev/flutter/widgets/DefaultTextStyle/of.html) the current [BuildContext](https://api.flutter.dev/flutter/widgets/BuildContext-class.html) to provide defaults. For more details on how to style text in a [RichText](https://api.flutter.dev/flutter/widgets/RichText-class.html) widget, see the documentation for [TextStyle](https://api.flutter.dev/flutter/painting/TextStyle-class.html).

Consider using the [Text](https://api.flutter.dev/flutter/widgets/Text-class.html) widget to integrate with the [DefaultTextStyle](https://api.flutter.dev/flutter/widgets/DefaultTextStyle-class.html) automatically. When all the text uses the same style, the default constructor is less verbose. The [Text.rich](https://api.flutter.dev/flutter/widgets/Text/Text.rich.html) constructor allows you to style multiple spans with the default text style while still allowing specified styles per span.

### Selections

To make this [RichText](https://api.flutter.dev/flutter/widgets/RichText-class.html) Selectable, the [RichText](https://api.flutter.dev/flutter/widgets/RichText-class.html) needs to be in the subtree of a [SelectionArea](https://api.flutter.dev/flutter/material/SelectionArea-class.html) or [SelectableRegion](https://api.flutter.dev/flutter/widgets/SelectableRegion-class.html) and a [SelectionRegistrar](https://api.flutter.dev/flutter/rendering/SelectionRegistrar-class.html) needs to be assigned to the [RichText.selectionRegistrar](https://api.flutter.dev/flutter/widgets/RichText/selectionRegistrar.html). One can use [SelectionContainer.maybeOf](https://api.flutter.dev/flutter/widgets/SelectionContainer/maybeOf.html) to get the [SelectionRegistrar](https://api.flutter.dev/flutter/rendering/SelectionRegistrar-class.html) from a context. This enables users to select the text in [RichText](https://api.flutter.dev/flutter/widgets/RichText-class.html)s with mice or touch events.

The [selectionColor](https://api.flutter.dev/flutter/widgets/RichText/selectionColor.html) also needs to be set if the selection is enabled to draw the selection highlights.

### Constructors

[RichText](https://api.flutter.dev/flutter/widgets/RichText/RichText.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [InlineSpan](https://api.flutter.dev/flutter/painting/InlineSpan-class.html) text, [TextAlign](https://api.flutter.dev/flutter/dart-ui/TextAlign.html) textAlign = TextAlign.start, [TextDirection](https://api.flutter.dev/flutter/dart-ui/TextDirection.html)? textDirection, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) softWrap = true, [TextOverflow](https://api.flutter.dev/flutter/painting/TextOverflow.html) overflow = TextOverflow.clip, @[Deprecated](https://api.flutter.dev/flutter/dart-core/Deprecated-class.html)('Use textScaler instead. ' 'Use of textScaleFactor was deprecated in preparation for the upcoming nonlinear text scaling support. ' 'This feature was deprecated after v3.12.0-2.0.pre.') [double](https://api.flutter.dev/flutter/dart-core/double-class.html) textScaleFactor = 1.0, [TextScaler](https://api.flutter.dev/flutter/painting/TextScaler-class.html) textScaler = TextScaler.noScaling, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? maxLines, [Locale](https://api.flutter.dev/flutter/dart-ui/Locale-class.html)? locale, [StrutStyle](https://api.flutter.dev/flutter/painting/StrutStyle-class.html)? strutStyle, [TextWidthBasis](https://api.flutter.dev/flutter/painting/TextWidthBasis.html) textWidthBasis = TextWidthBasis.parent, [TextHeightBehavior](https://api.flutter.dev/flutter/dart-ui/TextHeightBehavior-class.html)? textHeightBehavior, [SelectionRegistrar](https://api.flutter.dev/flutter/rendering/SelectionRegistrar-class.html)? selectionRegistrar, [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)? selectionColor})

Creates a paragraph of rich text.

### Properties

[children](https://api.flutter.dev/flutter/widgets/MultiChildRenderObjectWidget/children.html) → [List](https://api.flutter.dev/flutter/dart-core/List-class.html)<[Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)>

The widgets below this widget in the tree.

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[locale](https://api.flutter.dev/flutter/widgets/RichText/locale.html) → [Locale](https://api.flutter.dev/flutter/dart-ui/Locale-class.html)?

Used to select a font when the same Unicode character can be rendered differently, depending on the locale.

final

[maxLines](https://api.flutter.dev/flutter/widgets/RichText/maxLines.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)?

An optional maximum number of lines for the text to span, wrapping if necessary. If the text exceeds the given number of lines, it will be truncated according to [overflow](https://api.flutter.dev/flutter/widgets/RichText/overflow.html).

final

[overflow](https://api.flutter.dev/flutter/widgets/RichText/overflow.html) → [TextOverflow](https://api.flutter.dev/flutter/painting/TextOverflow.html)

How visual overflow should be handled.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[selectionColor](https://api.flutter.dev/flutter/widgets/RichText/selectionColor.html) → [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)?

The color to use when painting the selection.

final

[selectionRegistrar](https://api.flutter.dev/flutter/widgets/RichText/selectionRegistrar.html) → [SelectionRegistrar](https://api.flutter.dev/flutter/rendering/SelectionRegistrar-class.html)?

The [SelectionRegistrar](https://api.flutter.dev/flutter/rendering/SelectionRegistrar-class.html) this rich text is subscribed to.

final

[softWrap](https://api.flutter.dev/flutter/widgets/RichText/softWrap.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether the text should break at soft line breaks.

final

[strutStyle](https://api.flutter.dev/flutter/widgets/RichText/strutStyle.html) → [StrutStyle](https://api.flutter.dev/flutter/painting/StrutStyle-class.html)?

The strut style to use. Strut style defines the strut, which sets minimum vertical layout metrics.

final

[text](https://api.flutter.dev/flutter/widgets/RichText/text.html) → [InlineSpan](https://api.flutter.dev/flutter/painting/InlineSpan-class.html)

The text to display in this widget.

final

[textAlign](https://api.flutter.dev/flutter/widgets/RichText/textAlign.html) → [TextAlign](https://api.flutter.dev/flutter/dart-ui/TextAlign.html)

How the text should be aligned horizontally.

final

[textDirection](https://api.flutter.dev/flutter/widgets/RichText/textDirection.html) → [TextDirection](https://api.flutter.dev/flutter/dart-ui/TextDirection.html)?

The directionality of the text.

final

[textHeightBehavior](https://api.flutter.dev/flutter/widgets/RichText/textHeightBehavior.html) → [TextHeightBehavior](https://api.flutter.dev/flutter/dart-ui/TextHeightBehavior-class.html)?

Defines how to apply [TextStyle.height](https://api.flutter.dev/flutter/painting/TextStyle/height.html) over and under text.

final

[textScaleFactor](https://api.flutter.dev/flutter/widgets/RichText/textScaleFactor.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)

Deprecated. Will be removed in a future version of Flutter. Use [textScaler](https://api.flutter.dev/flutter/widgets/RichText/textScaler.html) instead.

read-only

[textScaler](https://api.flutter.dev/flutter/widgets/RichText/textScaler.html) → [TextScaler](https://api.flutter.dev/flutter/painting/TextScaler-class.html)

The font scaling strategy to use when laying out and rendering the text.

final

[textWidthBasis](https://api.flutter.dev/flutter/widgets/RichText/textWidthBasis.html) → [TextWidthBasis](https://api.flutter.dev/flutter/painting/TextWidthBasis.html)

Defines how to measure the width of the rendered text.

## text

A run of text with a single style.

The [Text](https://api.flutter.dev/flutter/widgets/Text-class.html) widget displays a string of text with single style. The string might break across multiple lines or might all be displayed on the same line depending on the layout constraints.

The [style](https://api.flutter.dev/flutter/widgets/Text/style.html) argument is optional. When omitted, the text will use the style from the closest enclosing [DefaultTextStyle](https://api.flutter.dev/flutter/widgets/DefaultTextStyle-class.html). If the given style's [TextStyle.inherit](https://api.flutter.dev/flutter/painting/TextStyle/inherit.html) property is true (the default), the given style will be merged with the closest enclosing [DefaultTextStyle](https://api.flutter.dev/flutter/widgets/DefaultTextStyle-class.html). This merging behavior is useful, for example, to make the text bold while using the default font family and size.

Using the [Text.rich](https://api.flutter.dev/flutter/widgets/Text/Text.rich.html) constructor, the [Text](https://api.flutter.dev/flutter/widgets/Text-class.html) widget can display a paragraph with differently styled [TextSpan](https://api.flutter.dev/flutter/painting/TextSpan-class.html)s. The sample that follows displays "Hello beautiful world" with different styles for each word.

### Interactivity

To make [Text](https://api.flutter.dev/flutter/widgets/Text-class.html) react to touch events, wrap it in a [GestureDetector](https://api.flutter.dev/flutter/widgets/GestureDetector-class.html) widget with a [GestureDetector.onTap](https://api.flutter.dev/flutter/widgets/GestureDetector/onTap.html) handler.

In a Material Design application, consider using a [TextButton](https://api.flutter.dev/flutter/material/TextButton-class.html) instead, or if that isn't appropriate, at least using an [InkWell](https://api.flutter.dev/flutter/material/InkWell-class.html) instead of [GestureDetector](https://api.flutter.dev/flutter/widgets/GestureDetector-class.html).

To make sections of the text interactive, use [RichText](https://api.flutter.dev/flutter/widgets/RichText-class.html) and specify a [TapGestureRecognizer](https://api.flutter.dev/flutter/gestures/TapGestureRecognizer-class.html) as the [TextSpan.recognizer](https://api.flutter.dev/flutter/painting/TextSpan/recognizer.html) of the relevant part of the text.

### Selection

[Text](https://api.flutter.dev/flutter/widgets/Text-class.html) is not selectable by default. To make a [Text](https://api.flutter.dev/flutter/widgets/Text-class.html) selectable, one can wrap a subtree with a [SelectionArea](https://api.flutter.dev/flutter/material/SelectionArea-class.html) widget. To exclude a part of a subtree under [SelectionArea](https://api.flutter.dev/flutter/material/SelectionArea-class.html) from selection, once can also wrap that part of the subtree with [SelectionContainer.disabled](https://api.flutter.dev/flutter/widgets/SelectionContainer/SelectionContainer.disabled.html).

### Constructors

[Text](https://api.flutter.dev/flutter/widgets/Text/Text.html)([String](https://api.flutter.dev/flutter/dart-core/String-class.html) data, {[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [TextStyle](https://api.flutter.dev/flutter/painting/TextStyle-class.html)? style, [StrutStyle](https://api.flutter.dev/flutter/painting/StrutStyle-class.html)? strutStyle, [TextAlign](https://api.flutter.dev/flutter/dart-ui/TextAlign.html)? textAlign, [TextDirection](https://api.flutter.dev/flutter/dart-ui/TextDirection.html)? textDirection, [Locale](https://api.flutter.dev/flutter/dart-ui/Locale-class.html)? locale, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? softWrap, [TextOverflow](https://api.flutter.dev/flutter/painting/TextOverflow.html)? overflow, @[Deprecated](https://api.flutter.dev/flutter/dart-core/Deprecated-class.html)('Use textScaler instead. ' 'Use of textScaleFactor was deprecated in preparation for the upcoming nonlinear text scaling support. ' 'This feature was deprecated after v3.12.0-2.0.pre.') [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? textScaleFactor, [TextScaler](https://api.flutter.dev/flutter/painting/TextScaler-class.html)? textScaler, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? maxLines, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? semanticsLabel, [TextWidthBasis](https://api.flutter.dev/flutter/painting/TextWidthBasis.html)? textWidthBasis, [TextHeightBehavior](https://api.flutter.dev/flutter/dart-ui/TextHeightBehavior-class.html)? textHeightBehavior, [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)? selectionColor})

Creates a text widget.

const

[Text.rich](https://api.flutter.dev/flutter/widgets/Text/Text.rich.html)([InlineSpan](https://api.flutter.dev/flutter/painting/InlineSpan-class.html) textSpan, {[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [TextStyle](https://api.flutter.dev/flutter/painting/TextStyle-class.html)? style, [StrutStyle](https://api.flutter.dev/flutter/painting/StrutStyle-class.html)? strutStyle, [TextAlign](https://api.flutter.dev/flutter/dart-ui/TextAlign.html)? textAlign, [TextDirection](https://api.flutter.dev/flutter/dart-ui/TextDirection.html)? textDirection, [Locale](https://api.flutter.dev/flutter/dart-ui/Locale-class.html)? locale, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? softWrap, [TextOverflow](https://api.flutter.dev/flutter/painting/TextOverflow.html)? overflow, @[Deprecated](https://api.flutter.dev/flutter/dart-core/Deprecated-class.html)('Use textScaler instead. ' 'Use of textScaleFactor was deprecated in preparation for the upcoming nonlinear text scaling support. ' 'This feature was deprecated after v3.12.0-2.0.pre.') [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? textScaleFactor, [TextScaler](https://api.flutter.dev/flutter/painting/TextScaler-class.html)? textScaler, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? maxLines, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? semanticsLabel, [TextWidthBasis](https://api.flutter.dev/flutter/painting/TextWidthBasis.html)? textWidthBasis, [TextHeightBehavior](https://api.flutter.dev/flutter/dart-ui/TextHeightBehavior-class.html)? textHeightBehavior, [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)? selectionColor})

Creates a text widget with a [InlineSpan](https://api.flutter.dev/flutter/painting/InlineSpan-class.html).

const

### Properties

[data](https://api.flutter.dev/flutter/widgets/Text/data.html) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)?

The text to display.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[locale](https://api.flutter.dev/flutter/widgets/Text/locale.html) → [Locale](https://api.flutter.dev/flutter/dart-ui/Locale-class.html)?

Used to select a font when the same Unicode character can be rendered differently, depending on the locale.

final

[maxLines](https://api.flutter.dev/flutter/widgets/Text/maxLines.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)?

An optional maximum number of lines for the text to span, wrapping if necessary. If the text exceeds the given number of lines, it will be truncated according to [overflow](https://api.flutter.dev/flutter/widgets/Text/overflow.html).

final

[overflow](https://api.flutter.dev/flutter/widgets/Text/overflow.html) → [TextOverflow](https://api.flutter.dev/flutter/painting/TextOverflow.html)?

How visual overflow should be handled.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[selectionColor](https://api.flutter.dev/flutter/widgets/Text/selectionColor.html) → [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)?

The color to use when painting the selection.

final

[semanticsLabel](https://api.flutter.dev/flutter/widgets/Text/semanticsLabel.html) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)?

An alternative semantics label for this text.

final

[softWrap](https://api.flutter.dev/flutter/widgets/Text/softWrap.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)?

Whether the text should break at soft line breaks.

final

[strutStyle](https://api.flutter.dev/flutter/widgets/Text/strutStyle.html) → [StrutStyle](https://api.flutter.dev/flutter/painting/StrutStyle-class.html)?

The strut style to use. Strut style defines the strut, which sets minimum vertical layout metrics.

final

[style](https://api.flutter.dev/flutter/widgets/Text/style.html) → [TextStyle](https://api.flutter.dev/flutter/painting/TextStyle-class.html)?

If non-null, the style to use for this text.

final

[textAlign](https://api.flutter.dev/flutter/widgets/Text/textAlign.html) → [TextAlign](https://api.flutter.dev/flutter/dart-ui/TextAlign.html)?

How the text should be aligned horizontally.

final

[textDirection](https://api.flutter.dev/flutter/widgets/Text/textDirection.html) → [TextDirection](https://api.flutter.dev/flutter/dart-ui/TextDirection.html)?

The directionality of the text.

final

[textHeightBehavior](https://api.flutter.dev/flutter/widgets/Text/textHeightBehavior.html) → [TextHeightBehavior](https://api.flutter.dev/flutter/dart-ui/TextHeightBehavior-class.html)?

Defines how to apply [TextStyle.height](https://api.flutter.dev/flutter/painting/TextStyle/height.html) over and under text.

final

[textScaleFactor](https://api.flutter.dev/flutter/widgets/Text/textScaleFactor.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

Deprecated. Will be removed in a future version of Flutter. Use [textScaler](https://api.flutter.dev/flutter/widgets/Text/textScaler.html) instead.

final

[textScaler](https://api.flutter.dev/flutter/widgets/Text/textScaler.html) → [TextScaler](https://api.flutter.dev/flutter/painting/TextScaler-class.html)?

The font scaling strategy to use when laying out and rendering the text.

final

[textSpan](https://api.flutter.dev/flutter/widgets/Text/textSpan.html) → [InlineSpan](https://api.flutter.dev/flutter/painting/InlineSpan-class.html)?

The text to display as a [InlineSpan](https://api.flutter.dev/flutter/painting/InlineSpan-class.html).

final

[textWidthBasis](https://api.flutter.dev/flutter/widgets/Text/textWidthBasis.html) → [TextWidthBasis](https://api.flutter.dev/flutter/painting/TextWidthBasis.html)?

Defines how to measure the width of the rendered text.