# Text class

A run of text with a single style.

The [Text](https://docs.flutter.io/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://docs.flutter.io/flutter/widgets/Text/style.html) argument is optional. When omitted, the text will use the style from the closest enclosing [DefaultTextStyle](https://docs.flutter.io/flutter/widgets/DefaultTextStyle-class.html). If the given style's [TextStyle.inherit](https://docs.flutter.io/flutter/painting/TextStyle/inherit.html) property is true, the given style will be merged with the closest enclosing [DefaultTextStyle](https://docs.flutter.io/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 new TextSpan.rich constructor, the [Text](https://docs.flutter.io/flutter/widgets/Text-class.html) widget can also be created with a [TextSpan](https://docs.flutter.io/flutter/painting/TextSpan-class.html) to display text that use multiple styles (e.g., a paragraph with some bold words).

## Sample code

**new** Text(

'Hello, $\_name! How are you?',

textAlign: TextAlign.center,

overflow: TextOverflow.ellipsis,

style: **new** TextStyle(fontWeight: FontWeight.bold),

)

## Interactivity

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

In a material design application, consider using a [FlatButton](https://docs.flutter.io/flutter/material/FlatButton-class.html) instead, or if that isn't appropriate, at least using an [InkWell](https://docs.flutter.io/flutter/material/InkWell-class.html) instead of [GestureDetector](https://docs.flutter.io/flutter/widgets/GestureDetector-class.html).

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

See also:

* [RichText](https://docs.flutter.io/flutter/widgets/RichText-class.html), which gives you more control over the text styles.
* [DefaultTextStyle](https://docs.flutter.io/flutter/widgets/DefaultTextStyle-class.html), which sets default styles for [Text](https://docs.flutter.io/flutter/widgets/Text-class.html) widgets.

Inheritance

* [Object](https://docs.flutter.io/flutter/dart-core/Object-class.html)
* [Diagnosticable](https://docs.flutter.io/flutter/foundation/Diagnosticable-class.html)
* [DiagnosticableTree](https://docs.flutter.io/flutter/foundation/DiagnosticableTree-class.html)
* [Widget](https://docs.flutter.io/flutter/widgets/Widget-class.html)
* [StatelessWidget](https://docs.flutter.io/flutter/widgets/StatelessWidget-class.html)
* Text

## Constructors

[**Text**](https://docs.flutter.io/flutter/widgets/Text/Text.html)([String](https://docs.flutter.io/flutter/dart-core/String-class.html) data, { [Key](https://docs.flutter.io/flutter/foundation/Key-class.html) key, [TextStyle](https://docs.flutter.io/flutter/painting/TextStyle-class.html) style, [TextAlign](https://docs.flutter.io/flutter/dart-ui/TextAlign-class.html) textAlign, [TextDirection](https://docs.flutter.io/flutter/dart-ui/TextDirection-class.html) textDirection, [bool](https://docs.flutter.io/flutter/dart-core/bool-class.html) softWrap, [TextOverflow](https://docs.flutter.io/flutter/rendering/TextOverflow-class.html) overflow, [double](https://docs.flutter.io/flutter/dart-core/double-class.html) textScaleFactor, [int](https://docs.flutter.io/flutter/dart-core/int-class.html) maxLines })

Creates a text widget. [[...]](https://docs.flutter.io/flutter/widgets/Text/Text.html)

*const*

[**Text.rich**](https://docs.flutter.io/flutter/widgets/Text/Text.rich.html)([TextSpan](https://docs.flutter.io/flutter/painting/TextSpan-class.html) textSpan, { [Key](https://docs.flutter.io/flutter/foundation/Key-class.html) key, [TextStyle](https://docs.flutter.io/flutter/painting/TextStyle-class.html) style, [TextAlign](https://docs.flutter.io/flutter/dart-ui/TextAlign-class.html) textAlign, [TextDirection](https://docs.flutter.io/flutter/dart-ui/TextDirection-class.html) textDirection, [bool](https://docs.flutter.io/flutter/dart-core/bool-class.html) softWrap, [TextOverflow](https://docs.flutter.io/flutter/rendering/TextOverflow-class.html) overflow, [double](https://docs.flutter.io/flutter/dart-core/double-class.html) textScaleFactor, [int](https://docs.flutter.io/flutter/dart-core/int-class.html) maxLines })

Creates a text widget with a [TextSpan](https://docs.flutter.io/flutter/painting/TextSpan-class.html).

*const*

## Properties

[**data**](https://docs.flutter.io/flutter/widgets/Text/data.html) → [String](https://docs.flutter.io/flutter/dart-core/String-class.html)

The text to display. [[...]](https://docs.flutter.io/flutter/widgets/Text/data.html)

*final*

[**maxLines**](https://docs.flutter.io/flutter/widgets/Text/maxLines.html) → [int](https://docs.flutter.io/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://docs.flutter.io/flutter/widgets/Text/maxLines.html)

*final*

[**overflow**](https://docs.flutter.io/flutter/widgets/Text/overflow.html) → [TextOverflow](https://docs.flutter.io/flutter/rendering/TextOverflow-class.html)

How visual overflow should be handled.

*final*

[**softWrap**](https://docs.flutter.io/flutter/widgets/Text/softWrap.html) → [bool](https://docs.flutter.io/flutter/dart-core/bool-class.html)

Whether the text should break at soft line breaks. [[...]](https://docs.flutter.io/flutter/widgets/Text/softWrap.html)

*final*

[**style**](https://docs.flutter.io/flutter/widgets/Text/style.html) → [TextStyle](https://docs.flutter.io/flutter/painting/TextStyle-class.html)

If non-null, the style to use for this text. [[...]](https://docs.flutter.io/flutter/widgets/Text/style.html)

*final*

[**textAlign**](https://docs.flutter.io/flutter/widgets/Text/textAlign.html) → [TextAlign](https://docs.flutter.io/flutter/dart-ui/TextAlign-class.html)

How the text should be aligned horizontally.

*final*

[**textDirection**](https://docs.flutter.io/flutter/widgets/Text/textDirection.html) → [TextDirection](https://docs.flutter.io/flutter/dart-ui/TextDirection-class.html)

The directionality of the text. [[...]](https://docs.flutter.io/flutter/widgets/Text/textDirection.html)

*final*

[**textScaleFactor**](https://docs.flutter.io/flutter/widgets/Text/textScaleFactor.html) → [double](https://docs.flutter.io/flutter/dart-core/double-class.html)

The number of font pixels for each logical pixel. [[...]](https://docs.flutter.io/flutter/widgets/Text/textScaleFactor.html)

*final*

[**textSpan**](https://docs.flutter.io/flutter/widgets/Text/textSpan.html) → [TextSpan](https://docs.flutter.io/flutter/painting/TextSpan-class.html)

The text to display as a TextSpan. [[...]](https://docs.flutter.io/flutter/widgets/Text/textSpan.html)

*final*

[*hashCode*](https://docs.flutter.io/flutter/dart-core/Object/hashCode.html) → [int](https://docs.flutter.io/flutter/dart-core/int-class.html)

The hash code for this object. [[...]](https://docs.flutter.io/flutter/dart-core/Object/hashCode.html)

*read-only, inherited*

[*key*](https://docs.flutter.io/flutter/widgets/Widget/key.html) → [Key](https://docs.flutter.io/flutter/foundation/Key-class.html)

Controls how one widget replaces another widget in the tree. [[...]](https://docs.flutter.io/flutter/widgets/Widget/key.html)

*final, inherited*

[*runtimeType*](https://docs.flutter.io/flutter/dart-core/Object/runtimeType.html) → [Type](https://docs.flutter.io/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

*read-only, inherited*

## Methods

[**build**](https://docs.flutter.io/flutter/widgets/Text/build.html)([BuildContext](https://docs.flutter.io/flutter/widgets/BuildContext-class.html) context) → [Widget](https://docs.flutter.io/flutter/widgets/Widget-class.html)

Describes the part of the user interface represented by this widget. [[...]](https://docs.flutter.io/flutter/widgets/Text/build.html)

[**debugFillProperties**](https://docs.flutter.io/flutter/widgets/Text/debugFillProperties.html)([DiagnosticPropertiesBuilder](https://docs.flutter.io/flutter/foundation/DiagnosticPropertiesBuilder-class.html) properties) → void

[*createElement*](https://docs.flutter.io/flutter/widgets/StatelessWidget/createElement.html)( → [StatelessElement](https://docs.flutter.io/flutter/widgets/StatelessElement-class.html)

Creates a [StatelessElement](https://docs.flutter.io/flutter/widgets/StatelessElement-class.html) to manage this widget's location in the tree. [[...]](https://docs.flutter.io/flutter/widgets/StatelessWidget/createElement.html)

*inherited*

[*debugDescribeChildren*](https://docs.flutter.io/flutter/foundation/DiagnosticableTree/debugDescribeChildren.html)( → [List](https://docs.flutter.io/flutter/dart-core/List-class.html)<[DiagnosticsNode](https://docs.flutter.io/flutter/foundation/DiagnosticsNode-class.html)>

Returns a list of [DiagnosticsNode](https://docs.flutter.io/flutter/foundation/DiagnosticsNode-class.html) objects describing this node's children. [[...]](https://docs.flutter.io/flutter/foundation/DiagnosticableTree/debugDescribeChildren.html)

*@protected, inherited*

[*noSuchMethod*](https://docs.flutter.io/flutter/dart-core/Object/noSuchMethod.html)([Invocation](https://docs.flutter.io/flutter/dart-core/Invocation-class.html) invocation) → dynamic

Invoked when a non-existent method or property is accessed. [[...]](https://docs.flutter.io/flutter/dart-core/Object/noSuchMethod.html)

*inherited*

[*toDiagnosticsNode*](https://docs.flutter.io/flutter/foundation/DiagnosticableTree/toDiagnosticsNode.html)([String](https://docs.flutter.io/flutter/dart-core/String-class.html) name, [DiagnosticsTreeStyle](https://docs.flutter.io/flutter/foundation/DiagnosticsTreeStyle-class.html) style }) → [DiagnosticsNode](https://docs.flutter.io/flutter/foundation/DiagnosticsNode-class.html)

Returns a debug representation of the object that is used by debugging tools and by [toStringDeep](https://docs.flutter.io/flutter/foundation/DiagnosticableTree/toStringDeep.html). [[...]](https://docs.flutter.io/flutter/foundation/DiagnosticableTree/toDiagnosticsNode.html)

*inherited*

[*toString*](https://docs.flutter.io/flutter/foundation/Diagnosticable/toString.html)([DiagnosticLevel](https://docs.flutter.io/flutter/foundation/DiagnosticLevel-class.html) minLevel: DiagnosticLevel.debug }) → [String](https://docs.flutter.io/flutter/dart-core/String-class.html)

Returns a string representation of this object.

*inherited*

[*toStringDeep*](https://docs.flutter.io/flutter/foundation/DiagnosticableTree/toStringDeep.html)([String](https://docs.flutter.io/flutter/dart-core/String-class.html) prefixLineOne: '', [String](https://docs.flutter.io/flutter/dart-core/String-class.html) prefixOtherLines, [DiagnosticLevel](https://docs.flutter.io/flutter/foundation/DiagnosticLevel-class.html) minLevel: DiagnosticLevel.debug }) → [String](https://docs.flutter.io/flutter/dart-core/String-class.html)

Returns a string representation of this node and its descendants. [[...]](https://docs.flutter.io/flutter/foundation/DiagnosticableTree/toStringDeep.html)

*inherited*

[*toStringShallow*](https://docs.flutter.io/flutter/foundation/DiagnosticableTree/toStringShallow.html)([String](https://docs.flutter.io/flutter/dart-core/String-class.html) joiner: ', ', [DiagnosticLevel](https://docs.flutter.io/flutter/foundation/DiagnosticLevel-class.html) minLevel: DiagnosticLevel.debug }) → [String](https://docs.flutter.io/flutter/dart-core/String-class.html)

Returns a one-line detailed description of the object. [[...]](https://docs.flutter.io/flutter/foundation/DiagnosticableTree/toStringShallow.html)

*inherited*

[*toStringShort*](https://docs.flutter.io/flutter/widgets/Widget/toStringShort.html)( → [String](https://docs.flutter.io/flutter/dart-core/String-class.html)

A short, textual description of this widget.

*inherited*

## Operators

[*operator ==*](https://docs.flutter.io/flutter/dart-core/Object/operator_equals.html)(dynamic other) → [bool](https://docs.flutter.io/flutter/dart-core/bool-class.html)

The equality operator. [[...]](https://docs.flutter.io/flutter/dart-core/Object/operator_equals.html)

*inherited*