

[SwiftUI](#) / Tables

API Collection

Tables

Display selectable, sortable data arranged in rows and columns.

Overview

Use a table to display multiple values across a collection of elements. Each element in the collection appears in a different row of the table, while each value for a given element appears in a different column. Narrow displays may adapt to show only the first column of the table.



When you create a table, you provide a collection of elements, and then tell the table how to find the needed value for each column. In simple cases, SwiftUI infers the element for each row, but you can also specify the row elements explicitly in more complex scenarios. With a small amount of additional configuration, you can also make the items in the table selectable, and the columns sortable.

Like a [List](#), a table includes implicit vertical scrolling that you can configure using the view modifiers described in [Scroll views](#). For design guidance, see [Lists and tables](#) in the Human Interface Guidelines.

Topics

Creating a table

`{}` Building a great Mac app with SwiftUI

Create engaging SwiftUI Mac apps by incorporating side bars, tables, toolbars, and several other popular user interface elements.

`struct Table`

A container that presents rows of data arranged in one or more columns, optionally providing the ability to select one or more members.

`func tableStyle<S>(S) -> some View`

Sets the style for tables within this view.

Creating columns

`struct TableColumn`

A column that displays a view for each row in a table.

`protocol TableColumnContent`

A type used to represent columns within a table.

`struct TableColumnAlignment`

Describes the alignment of the content of a table column.

`struct TableColumnBuilder`

A result builder that creates table column content from closures.

`struct TableColumnForEach`

A structure that computes columns on demand from an underlying collection of identified data.

Customizing columns

`func tableColumnHeaders(Visibility) -> some View`

Controls the visibility of a Table's column header views.

`struct TableColumnCustomization`

A representation of the state of the columns in a table.

```
struct TableColumnCustomizationBehavior
```

A set of customization behaviors of a column that a table can offer to a user.

Creating rows

```
struct TableRow
```

A row that represents a data value in a table.

```
protocol TableRowContent
```

A type used to represent table rows.

```
struct TableHeaderRowContent
```

A table row that displays a single view instead of columned content.

```
struct TupleTableRowContent
```

A type of table column content that creates table rows created from a Swift tuple of table rows.

```
struct TableForEachContent
```

A type of table row content that creates table rows created by iterating over a collection.

```
struct EmptyTableRowContent
```

A table row content that doesn't produce any rows.

```
protocol DynamicTableRowContent
```

A type of table row content that generates table rows from an underlying collection of data.

```
struct TableRowBuilder
```

A result builder that creates table row content from closures.

Adding progressive disclosure

```
struct DisclosureTableRow
```

A kind of table row that shows or hides additional rows based on the state of a disclosure control.

```
struct TableOutlineGroupContent
```

An opaque table row type created by a table's hierarchical initializers.

See Also

View layout

- ☰ Layout fundamentals
Arrange views inside built-in layout containers like stacks and grids.
- ☰ Layout adjustments
Make fine adjustments to alignment, spacing, padding, and other layout parameters.
- ☰ Custom layout
Place views in custom arrangements and create animated transitions between layout types.
- ☰ Lists
Display a structured, scrollable column of information.
- ☰ View groupings
Present views in different kinds of purpose-driven containers, like forms or control groups.
- ☰ Scroll views
Enable people to scroll to content that doesn't fit in the current display.