

[TabletopKit](#) / [TabletopShape](#)

## Structure

# TabletopShape

An object that represents the physical properties of the table.

visionOS 2.0+

```
struct TabletopShape
```

## Overview

To create a round table, use the [`round\(center:radius:thickness:in:\)`](#) initializer, or if you render the table using RealityKit, the [`round\(entity:\)`](#) initializer. For a rectangular table, use the equivalent [`rectangular\(center:width:height:thickness:in:\)`](#) or [`rectangular\(entity:\)`](#) initializer.

## Topics

### Creating a round or rectangular table

```
static func rectangular(center: Point3D, width: Float, height: Float,  
thickness: Float, in: UnitLength) -> TabletopShape
```

Creates a rectangular tabletop shape with the specified center and dimensions.

```
static func round(center: Point3D, radius: Float, thickness: Float, in:  
UnitLength) -> TabletopShape
```

Creates a round tabletop shape with the specified center, radius, and thickness.

## Creating a table that you render using an entity

```
static func rectangular(entity: Entity) -> TabletopShape
```

```
static func round(entity: Entity) -> TabletopShape
```

---

## Relationships

### Conforms To

Sendable, SendableMetatype

---

## See Also

### Essentials

{ } Creating tabletop games

Develop a spatial board game where multiple players interact with pieces on a table.

{ } Synchronizing group gameplay with TabletopKit

Maintain game state across multiple players in a race to capture all the coins.

`class TabletopGame`

An object that manages the setup and gameplay of a tabletop game.

`struct TableSetup`

An object that represents the arrangement of seats, equipment, and counters around the game table.

`protocol Tabletop`

A protocol for the table surface in your game.

`protocol EntityTabletop`

A protocol for the table surface in your game when you render it using RealityKit.