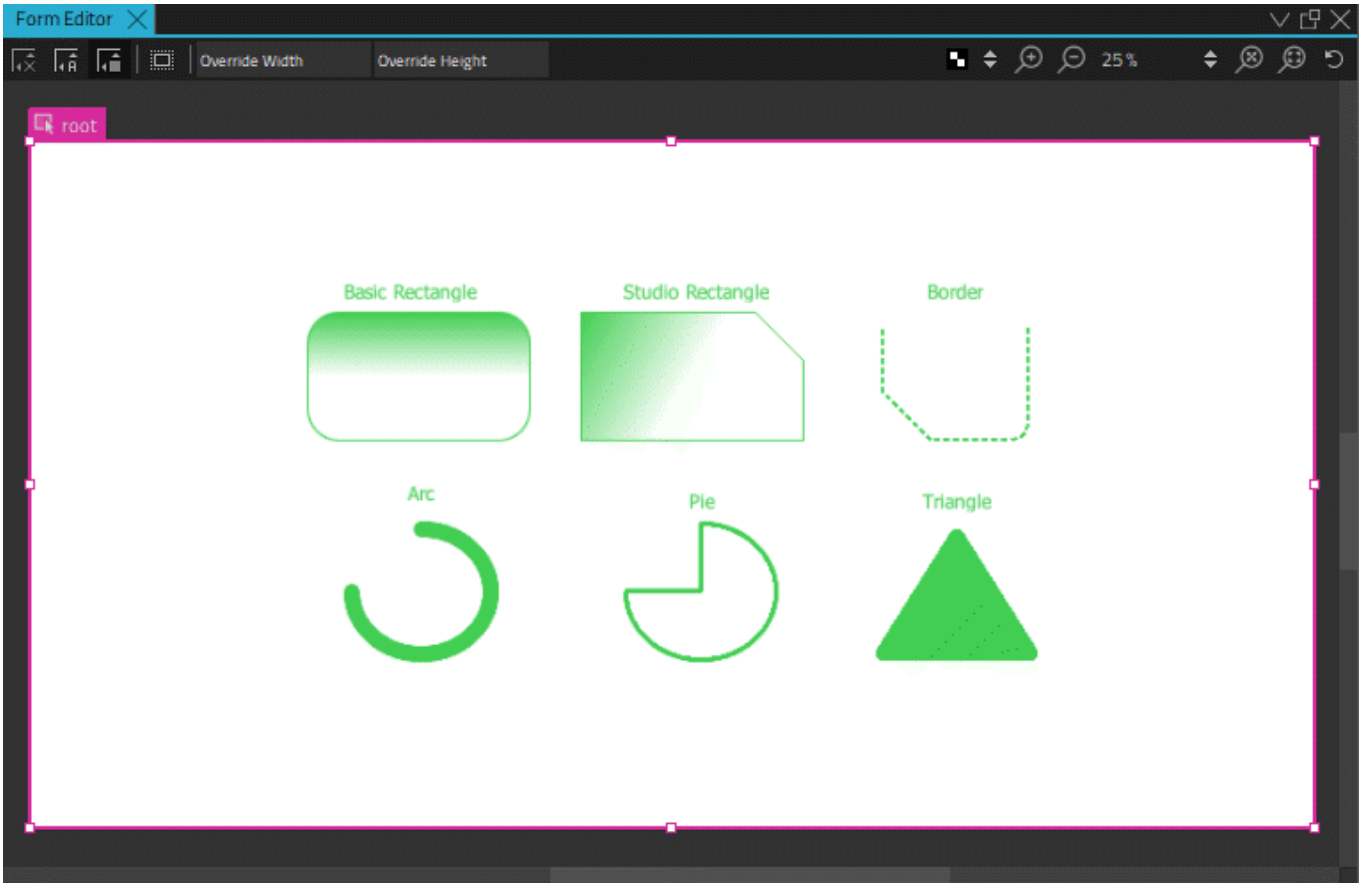
形状

Qt设计工作室是一个UI设计工具，而不是一个通用的绘图工具，因此，重点是提供现成的UI控件，您可以根据需要修改。控件的某些属性的值是使用样式指定的，因此您无法更改它们。

但是，您可以使用组件中的某些组件来绘制基本形状，如矩形。此外，Qt Design Studio还附带了一组更强大，更灵活的图形基元，允许在2D视图中创建更复杂的形状，例如边框，三角形，弧形和馅饼。

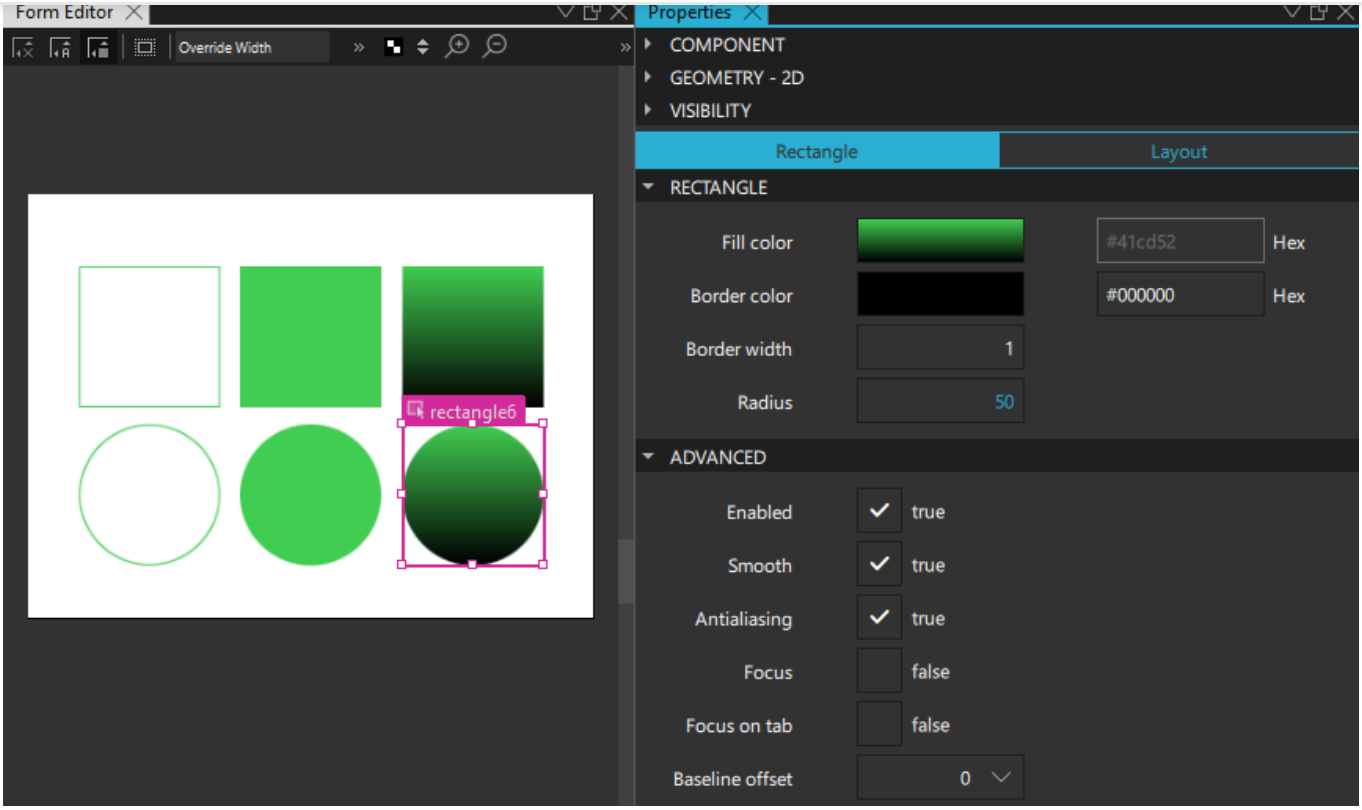


"组件"中的大多数可视组件都基于"项"组件。即使它本身没有可视外观（例如，类似于鼠标区域），它也定义了跨可视组件通用的所有属性，例如位置、大小和可见性。有关更多信息，请参见基本组件属性。

此外，每个组件都有一组属性，用于指定其可视外观和行为。您可以修改属性值以设置填充和边框颜色、描边宽度以及形状的其他特征。

以下各节更详细地介绍了可用的形状及其属性。可以在**"属性"**视图中修改属性的值。

矩形



矩形可以使用纯色填充颜色或在"填充颜色"字段中设置的线性渐变进行**填充**。还可以使用颜色选取器来选择颜色，并使用渐变选取器从一组预定义的 Web 梯度中选择渐变。

通过设置"**边框颜色**"和"**边框宽度**"字段的值，可以将可选的实心边框添加到具有自己的颜色和厚度的矩形中。

若要创建没有填充颜色的边框，请选择将颜色设置为*透明的*按钮。

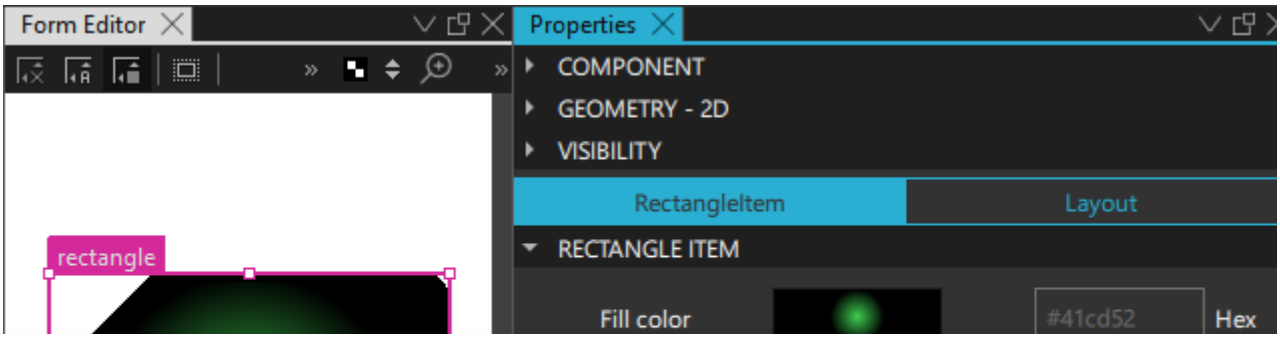
通过设置"**半径**"字段的值，可以创建具有圆角的形状。通过矩形宽度、高度和半径属性值的正确组合，可以创建椭圆形和圆形形状。例如，要绘制一个完整的圆，请首先绘制一个所有四条边相等的矩形，然后将其半径设置为边长的一半。

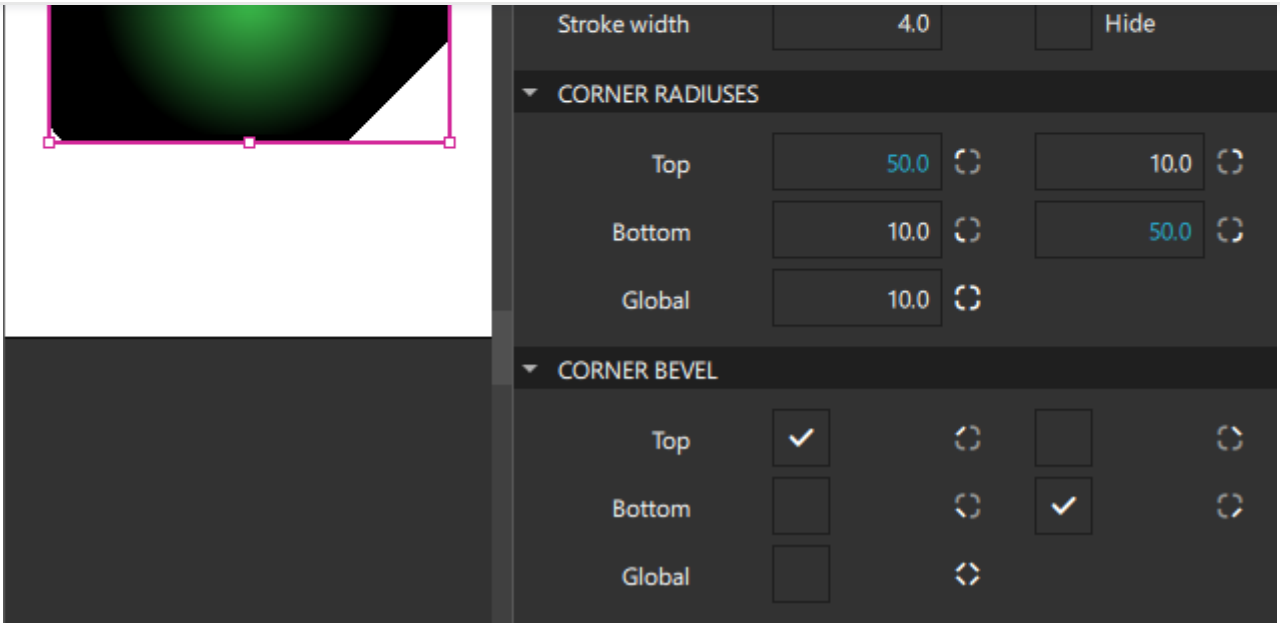
When drawing curved edges, consider enabling the **Antialiasing** check box in the **Advanced** section to improve the appearance of your shape.

Studio Rectangle

If you want to modify each corner of the rectangle independently or use a dashed border, create an instance of the **Rectangle** component available in **Components** > **Qt Quick Studio Components** instead of the basic **Rectangle** component available in **Default Components** > **Basic**.

By setting the values of properties in the **Corner Radiuses** section, you can draw each corner independently. By using radius values in combination with the values in the **Corner Bevel** section, you can create shapes with cut corners.



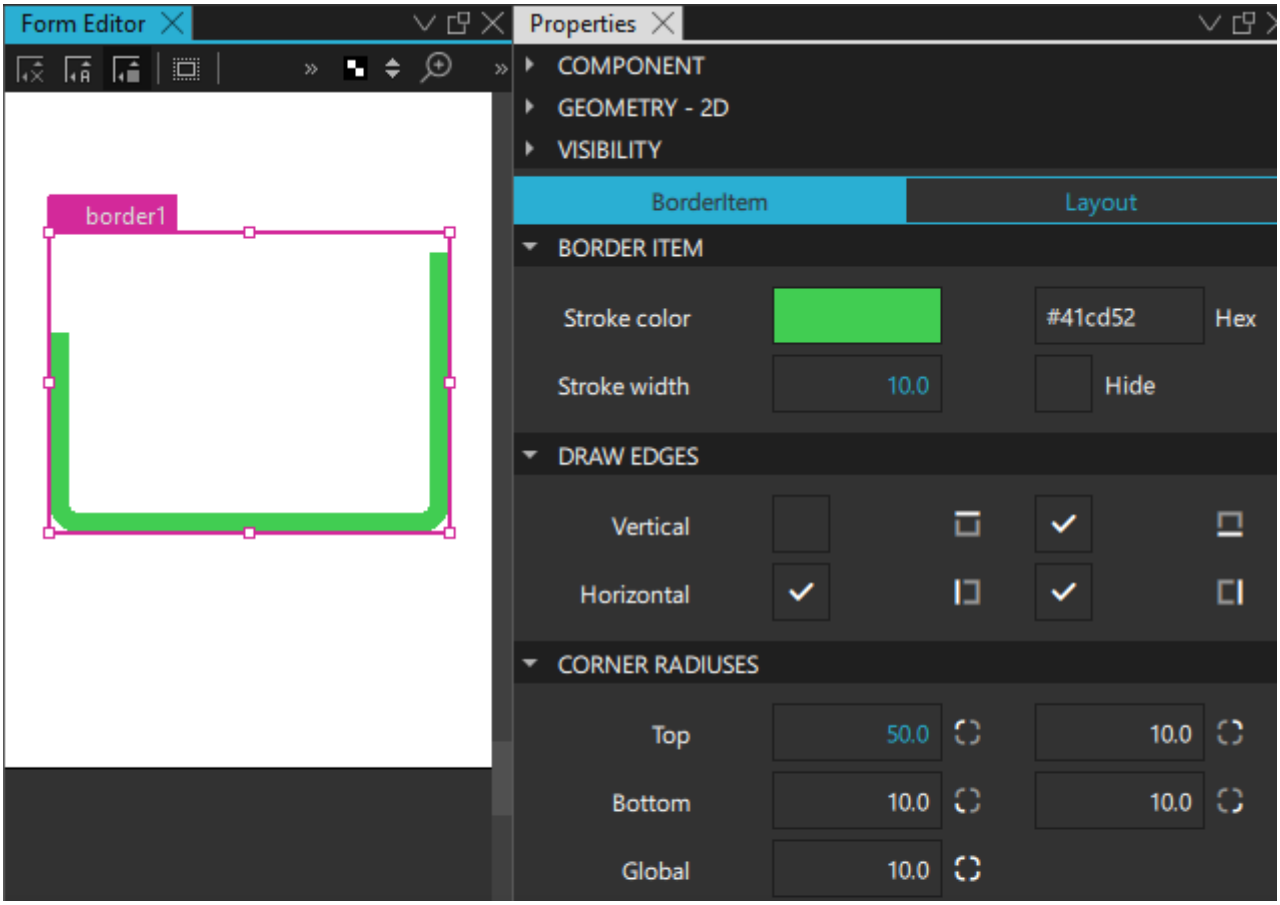


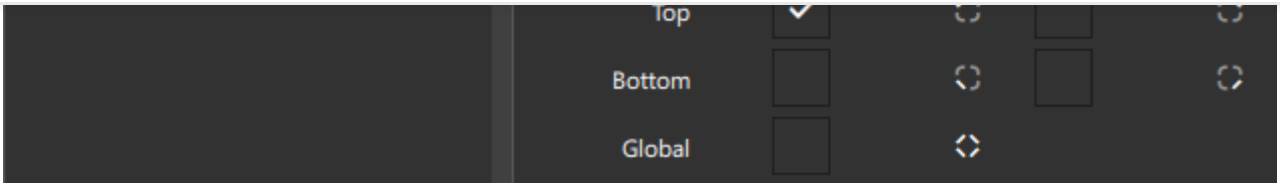
In addition to linear gradients, you can specify conical and radial gradients in the **Fill color** field.

In the **Stroke Details** section, you can specify the border mode, line style, and dash pattern for dashed and dotted lines. For more information, see Strokes.

Border

The Border component is used to create borders out of four segments: left, top, right, and bottom. By selecting the check boxes in the **Draw Edges** section, you can determine whether each of the segments is visible. This enables you to draw solid or dashed lines with specified dash patterns and joint and cap styles.

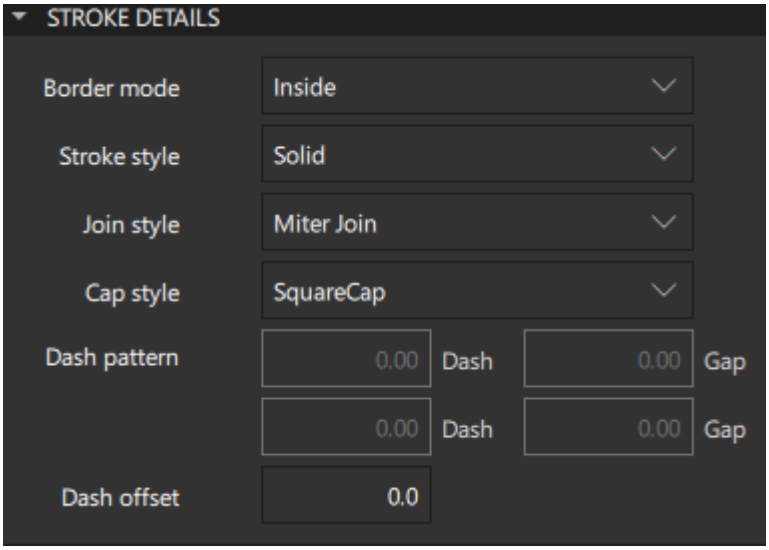




Set the color of the border in the **Stroke color** field and its thickness in the **Stroke width** field. Then specify additional properties for borders in the **Stroke Details** section.

Strokes

In the **Stroke Details** section, you can specify the border mode, line style, and dash pattern for dashed and dotted lines.



In the **Border mode** field, you can specify whether the border is drawn along the inside or outside edge of the component, or on top of the edge.

If you select a dashed or dotted pattern in the **Stroke style** field, you can specify the dash pattern as the dashes and the gaps between them in the **Dash pattern** field. The dash pattern is specified in units of the stroke width. That is, a dash with the length 5 and width 10 is 50 pixels long.

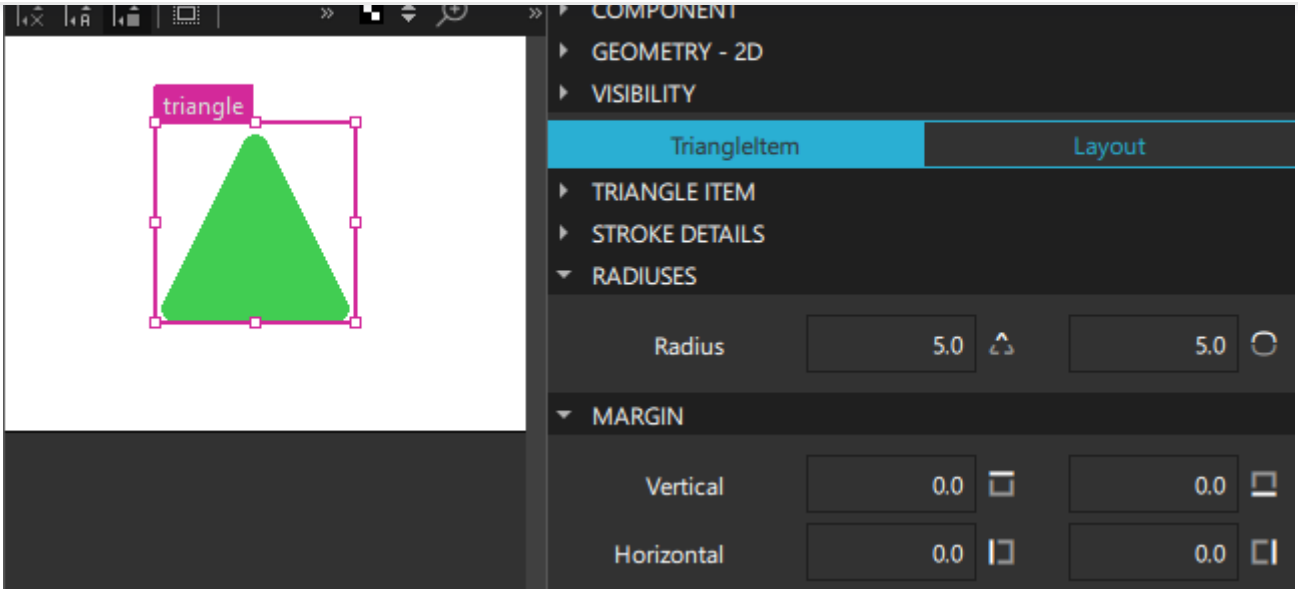
The value of the **Dash offset** field specifies the starting point of the dash pattern for a line. It is measured in terms of the units used to specify the dash pattern. For example, a pattern where each stroke is four units long, followed by a gap of two units, will begin with the stroke when drawn as a line. However, if the dash offset is set to 4.0, any line drawn will begin with the gap. Values of the offset up to 4.0 will cause part of the stroke to be drawn first, and values of the offset between 4.0 and 6.0 will cause the line to begin with part of the gap.

In the **Join style** field, select **Miter Join** to extend the outer edges of the lines to meet at an angle and to fill the area between them. Select **Bevel Join** to fill the triangular notch between the two lines. Select **Round Join** to fill a circular arc between the two lines.

The value of the **Cap style** property specifies whether line ends are square or rounded.

Triangle

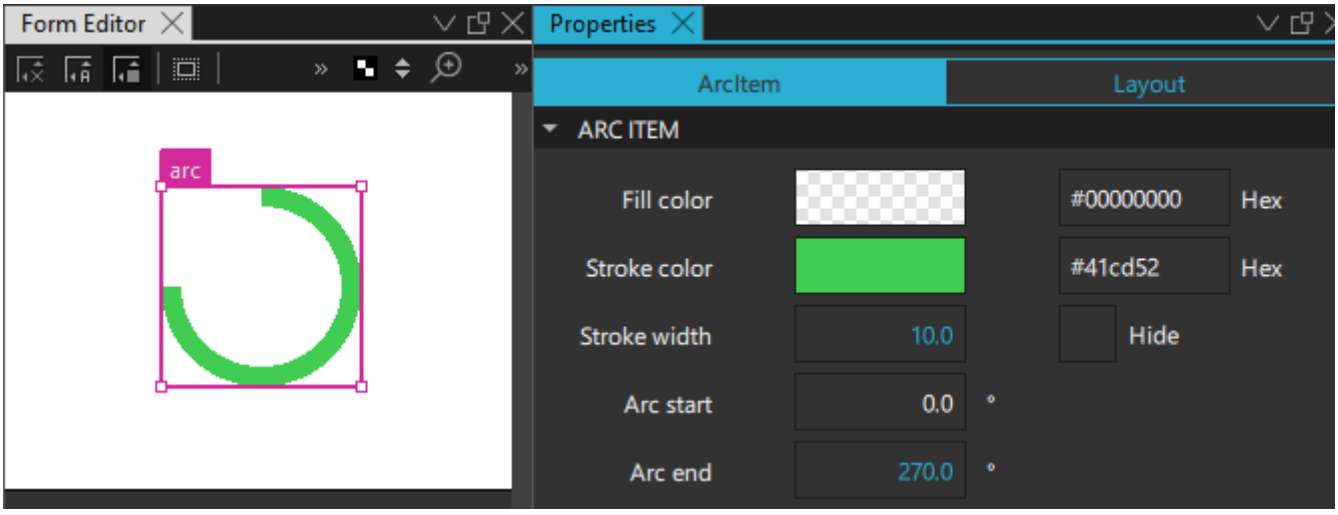
The Triangle component can be used to draw triangles with different dimensions and shapes. The component is enclosed in an invisible rectangle that determines its size. The dimensions of the triangle can be changed to make it elongated or squatter with space around it by setting the top, bottom, left, and right margins in the **Margin** section. The margins are set between the triangle and the edges of the parent rectangle.



The fill and stroke color property values are set in the same way as for a rectangle. The border property values are described in Strokes.

Arc

An arc is specified by setting values in degrees in the **Arc start** and **Arc end** fields. The arc can be just a line or a filled outline. The properties of the line or outline are described in Strokes.

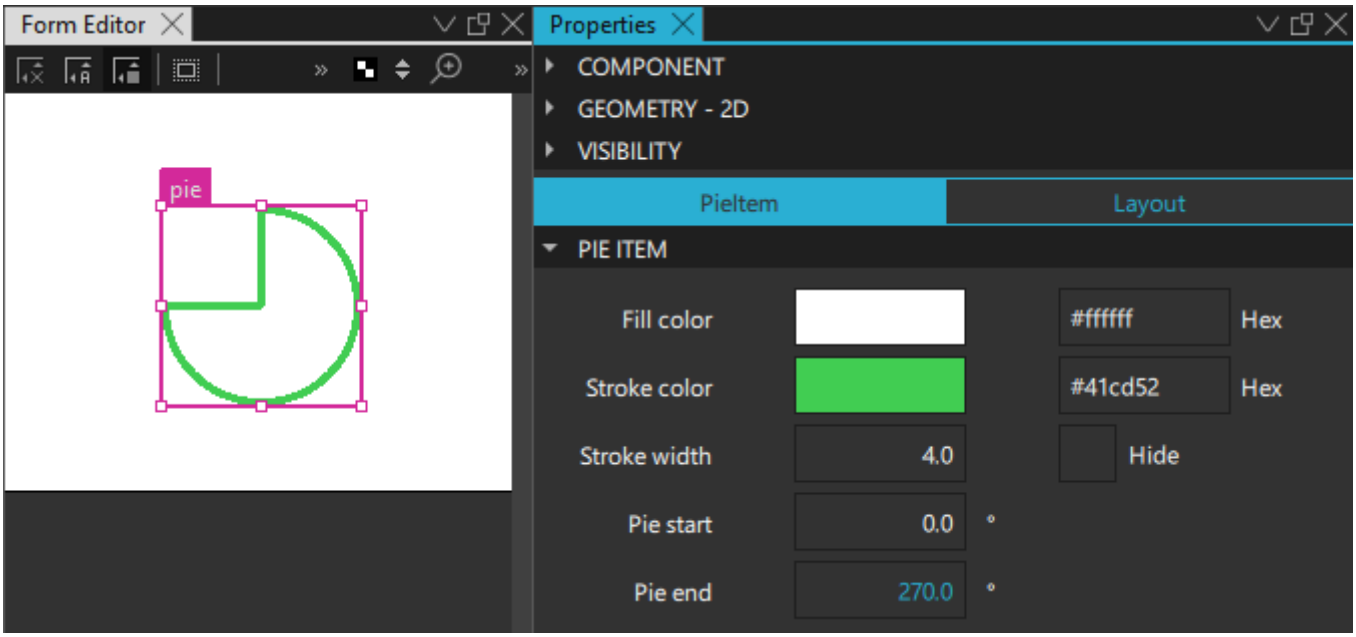


The area between the arc's start and end points or the area inside the outline are painted using either a solid fill color or a gradient.

The **Round outline**, **Round start**, and **Round end** properties specify whether the end points of the arc outline have rounded caps. For an arc that does not have an outline, the **Cap style** field in the **Stroke Details** section specifies whether the line ends are square or rounded.

Pie

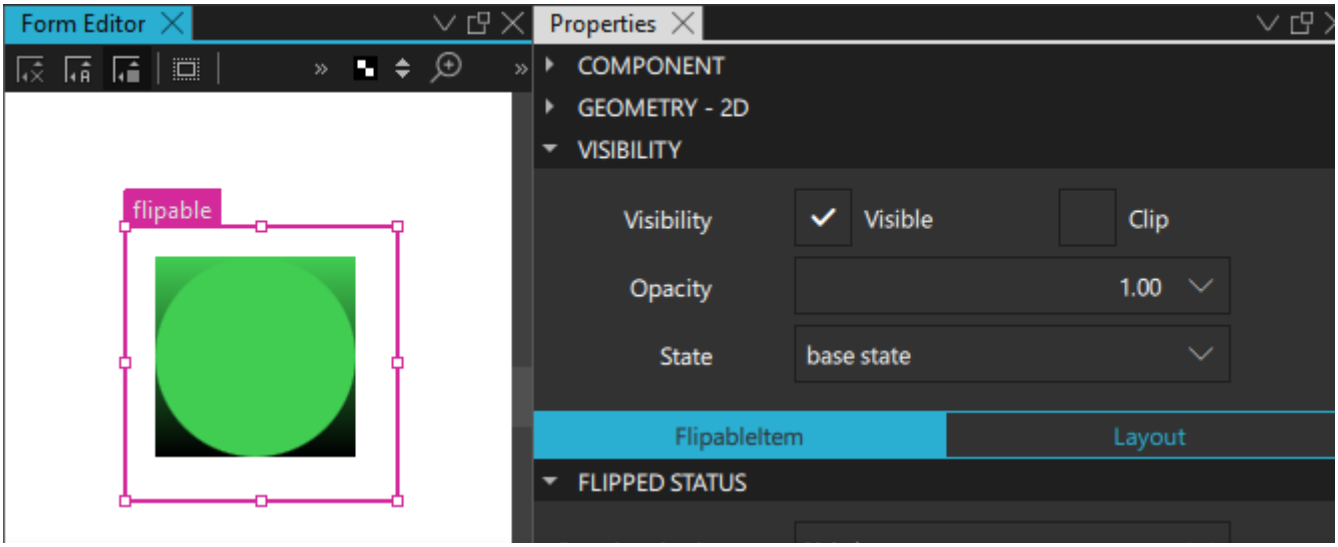
The Pie component is used to create a pie slice, a pie that is missing slices, or just the pie rind (similar to an arc), depending on the values of the **Pie start** and **Pie end** fields and the **Hide** check box.

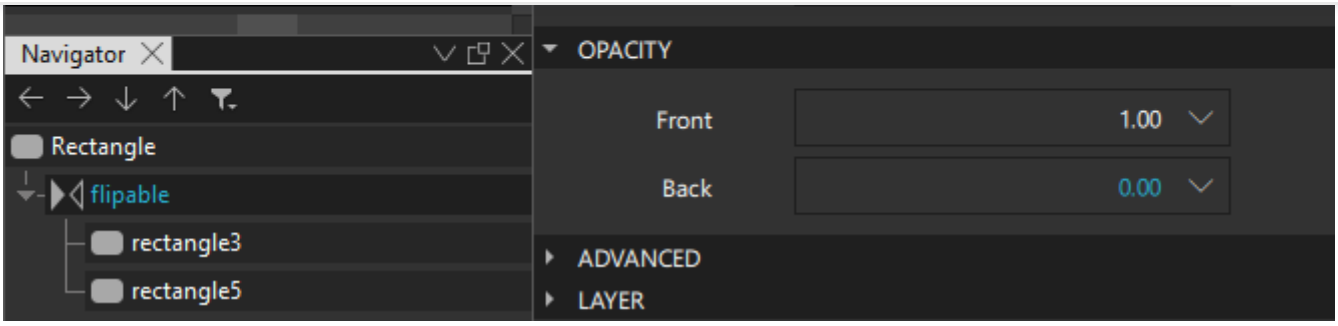


The filling of the pie is painted using either a solid fill color or a gradient. The fill and stroke color property values are set in the same way as for a rectangle. The border property values are described in Strokes.

Flipable

A Flipable component can be visibly *flipped* between its front and back sides, like a card. The front and back sides are specified by using any two components inside the Flipable component. The component with the higher z-order is the front side. The **Front** and **Back** fields in the **Opacity** section are used to hide and show either the front or back side of the item at a time.





The **Flip angle** property is used to animate the angle of the component to produce the flipping effect. The value of the **Rotational axis** property determine whether the component is rotated around the x-axis or the y-axis.

Summary of Shapes

The following table lists the components that you can use to draw shapes. The *Location* column indicates the location of the component in **Components**. The *MCU* column indicates which components are supported on MCUs.

Icon	Name	Location	MCU	Purpose
	Arc	Qt Quick Studio Components		An arc that begins and ends at given positions.
	Border	Qt Quick Studio Components		A line with four segments that you can show and shape individually.
	Pie	Qt Quick Studio Components		A pie slice or a pie with a slice missing from it.
	Flipable	Qt Quick Studio Components		A component that can be visibly *flipped* between its front and back sides, like a card.
	Rectangle	Default Components - Basic	✓	A rectangle that is painted with a solid fill color or linear gradient and an optional border. You can use the radius property to draw circles.
	Rectangle	Qt Quick Studio Components		An extended rectangle that is painted with a solid fill color or linear, conical, or radial gradients, and corners that you can shape independently of each other.
	Triangle	Qt Quick Studio Components		A triangle with different dimensions and shapes that is enclosed in an invisible rectangle.

‹ Preset Components

Text ›

Qt DOCUMENTATION

Qt Company

Contact Us

Company

About Us
Investors
Newsroom
Careers
Office Locations

Licensing

Terms & Conditions
Open Source
FAQ

Support

Support Services
Professional Services
Partners
Training

For Customers

Support Center
Downloads
Qt Login
Contact Us
Customer Success

Community

Contribute to Qt
Forum
Wiki
Downloads
Marketplace

Feedback Sign In