# BlockControl

A block control does not accept anything else on the same horizontal with it. They are displayed vertically one after the other.

## Margins

* The top and bottom margins are displayed as empty lines.
* The left and right margins are white spaces displayed using the Console's background color.

Note: For the left and right margins to be displayed the inherited controls must always use the ControlDisplay instance provided as parameters on the DoDisplayContent method.

## Paddings

* All the Paddings (left, right, top, bottom) are white spaces displayed using the control's background color.

Note: For the left and right padding to be displayed the inherited controls must always use the ControlDisplay instance provided as parameters on the DoDisplayContent method.

## Content

* The content is displayed using the control's background and foreground colors.

Note: For the content to be displayed using the control's foreground and background colors, the inherited controls must always use the ControlDisplay instance provided as parameters on the DoDisplayContent method.

* The ControlDisplay instance will detect when the line is full and will wrap

# How To: Create a custom block control

## Step 1) BlockControl class

Create a new class and inherit from BlockControl base class.

public class CustomControl : BlockControl

{

// ...

}

## Step 2) DoDisplayContent method

Override the DoDisplayContent method:

protected abstract void DoDisplayContent(ControlDisplay display);

Note: The derived control must always write to the console using the provided ControlDisplay instance. It will count the displayed lines, this information being used by other functionalities like EraseAfterClose.

## Step 3) DesiredContentWidth property

Override the DesiredContentWidth property:

protected virtual int DesiredContentWidth { get; }

Note: This property is useful when calculating the horizontal alignment of the content.