TELUS Component Library

Box

Multi-Platform Component | Figma UI KIT

Hello world

```
<Box variant={{ background: 'light' }} space={3}>
  <Typography>Hello world</Typography>
</Box>
```

Introduction

Box creates space (padding) around content, and has variants that give content a background.

Follow the appropriate instructions to add this component in to your app.

Guidance

Use Box to create padding within a container, around all its content. To create space after a box, between boxes or between items in a box, combine `Box` with other layout components, such as `StackView` and `Spacer`:

Some content

Some example content within the first box.

Another line of text in the first box.

More content

Some example content within the second box.

Another line of text in the second box.

```
<Box variant={{ background: 'light' }} space={4}>
  <StackView direction={{ xs: 'column', md: 'row' }} space={4}>
    <Box flex={1} variant={{ background: 'lightest' }} space={4}>
      <Typography variant={{ size: 'h3' }}>Some content</Typography>
      <Spacer space={4} />
      <Typography>Some example content within the first box.</Typography>
      <Spacer space={2} />
      <Typography>Another line of text in the first box.</Typography>
    </Box>
    <Box flex={1} variant={{ background: 'lightest' }} space={4}>
      <Typography variant={{ size: 'h3' }}>More content</Typography>
      <Spacer space={4} />
      <Typography>Some example content within the second box.</Typography>
      <Spacer space={2} />
      <Typography>Another line of text in the second box.</Typography>
    </Box>
  </StackView>
</Box>
```

`space` prop

The `space` prop sets the default padding on all sides, using the **spacing scale**.

Simple numbers may be used, or objects (see spacing scale docs for details. These examples change spacing depending on the viewport.

This box has wider padding on wider screens.

```
<Box variant={{ background: 'light' }} space={{ xs: 2, sm: 3,
lg: 5, xl: 6 }}>
  <Typography>This box has wider padding on wider screens.
</Typography>
</Box>
```

`horizontal` and `vertical` props

These props are the same as the `space` prop but apply space specifically on the left and right (`horizontal`) or top and bottom (`vertical`) sides of a box. If `space` and `horizontal` or `vertical` are provided, `space` is treated as the default and the more specific props as overrides.

This box has wider horizontal padding on wider screens, and stable vertical padding.

```
<Box variant={{ background: 'light' }} vertical={4} horizontal=
{{ xs: 2, sm: 3, lg: 5, xl: 6 }}>
  <Typography>
    This box has wider horizontal padding on wider screens, and
stable vertical padding.
  </Typography>
</Box>
```

`left`, `right`, `top` and `bottom` props

These props are the same as the `space` props above but apply to one specific side only. If one of these is provided alongside `space`, `horizontal` or `vertical`, those are treated as the defaults and the more specific prop as an override.

This box has wider bottom padding on wider screens, and stable padding on other sides.

```
<Box variant={{ background: 'light' }} space={4} bottom={{ xs: 2, sm: 3, lg: 5, xl: 6 }}>
  <Typography>
    This box has wider bottom padding on wider screens, and stable padding on other sides.
  </Typography>
</Box>
```

`flex` prop and layout

By default, the box takes its minimum width as the width of its children (flex value of `0`). `flex={1}` may be passed to make the box stretch and grow to fit a flex container.

Other values greater than `0` may be passed to distribute space proportionately. See the React Native `flex` documentation for more details. Behaviour is similar on web and in native apps.

This box has flex prop set to 1. It grows and shrinks, claiming half as much space as its flex: 2 neighbour.

This box has flex prop set to 2. It grows and shrinks, claiming twice as much space as its flex: 1 neighbour.

```
<StackView direction="row" space={3}>
  <Box space={3} variant={{ background: 'light' }} flex={1}>
    <Typography>
      This box has flex prop set to 1. It grows and shrinks, claiming half as much space as its
      flex: 2 neighbour.
    </Typography>
  </Box>
  <Box space={3} variant={{ background: 'light' }} flex={2}>
    <Typography>
      This box has flex prop set to 2. It grows and shrinks,
claiming twice as much space as its
      flex: 1 neighbour.
    </Typography>
  </Box>
</StackView>
```

Accessibility

By default, Box applies no accessibility props or values relevant to accessibility tools. It may accept React Native accessibility props (web docs, native docs), and applies them to the outer container.

`Tag` prop for semantic HTML

By default, Box renders a `<div>` on web and a `View` with no accessibility role on native apps.

To render a HTML attribute with a semantic meaning on web, pass a string naming a HTML semantic layout tag to the `tag` prop. An `accessibilityRole` may also be passed if a role is needed in native apps. Heading tags (`h1`, `h2`, `h3`, `h4`, `h5`, `h6`) are by default mapped to "heading" role in native apps.

See the `tag` entry in the props table for the full list of supported tags.

"h2" tag on web

This subheading is part of the semantic h2 heading

This is inside an "Article" tag on web.
In native apps, it is treated like any other content.
The h2 above it is treated as accessibilityRole="heading".

All text in this box is treated as a "Label" on web.

In native apps, it is set to behave as a "header".

This is inside a "Section" tag on web.
On web, the label box's content is this section box's accessible name.
This helps users of accessibility tools understand a page's structure.
In native apps, headers may be the closest equivalent to named sections.

```
<Box tag="main">
  <Box tag="h2" space={2}>
    <Typography variant={{ size: 'h1' }}>"h2" tag on web</Typography>
    <Spacer space={2} />
    <Typography variant={{ size: 'h4' }}>
      This subheading is part of the semantic h2 heading
    </Typography>
  </Box>
  <Box tag="article" space={2}>
    <Typography>This is inside an "Article" tag on web.</Typography>
    <Typography>In native apps, it is treated like any other content.</Typography>
    <Typography>The h2 above it is treated as accessibilityRole="heading".</Typography>
  </Box>
  <Box tag="label" space={2} nativeID="label-box" accessibilityRole="header">
    <Typography>All text in this box is treated as a "Label" on web.</Typography>
    <Typography>In native apps, it is set to behave as a "header".</Typography>
  </Box>
  <Box tag="section" space={2} accessibilityLabelledBy="label-box">
    <Typography>This is inside a "Section" tag on web.</Typography>
    <Typography>On web, the label box's content is this section box's accessible name.</Typography>
    <Typography>This helps users of accessibility tools understand a page's structure.</Typography>
    <Typography>
      In native apps, headers may be the closest equivalent to named sections.
    </Typography>
  </Box>
</Box>
```

Platform considerations

The component is available on both native platforms and web.

Scrollable boxes

The prop `scroll` may be used to make a box scrollable. This changes the box from rendering a `View` to rendering a `ScrollView`. If an object is passed to the `scroll` prop, its properties are passed to the `ScrollView` as props.

On native apps, where screens do not scroll by default, a `Box` with `scroll` set and appropriate spacing and background colour is therefore a good first container for an app screen.

Do not use `scroll` to make a box within a page scrollable, unless it is as part of a component with a design that is accessible and makes it visually apparent that additional content can be reached. See `Tabs` for an example of such a component.

Props

Name	Type	Platform	Default	Description
horizontal	union	standard	space	Sets left and right p theme's spacing sc SpacingValue}
vertical	union	standard	space	Sets top and botton theme's spacing sc SpacingValue}
top	union	standard	vertical	Sets top padding us spacing scale. @se SpacingValue}

Name	Type	Platform	Default	Description
bottom	union	standard	vertical	Sets bottom paddin spacing scale. @se SpacingValue}
left	union	standard	horizontal	Sets left padding us spacing scale. @se SpacingValue}
right	union	standard	horizontal	Sets right padding u spacing scale. @se SpacingValue}
variant	variant	standard		System variant prop more details
tokens	tokens	standard		System tokens prop more details
space	union	standard		Sets default paddin box using the theme @see {@link Spacin
flex	number	standard		Sets the \`flex\` style \`flexGrow\`, \`flexSh styles. Set as 1 for t and grow to fit in th another number to s disproportionately w https://reactnative.c for details. With the takes its minimum w content.
scroll	union	standard		Renders a scrollable of an unscrollable V

Name	Type	Platform	Default	Description
				object of ScrollView passed to the rende
tag	'h1' | 'h2' | 'h3' | 'h4' | 'h5' | 'h6' | 'article' | 'aside' | 'blockquote' | 'footer' | 'figure' | 'form' | 'header' | 'ul' | 'li' | 'main' | 'nav' | 'section' | 'label'	standard		Optional semantic H the container on we styling. In native ap provided ("h1", "h2' accessibilityRole pr accessibilityRole wi "heading".
testID	string	standard		Use in tests if the b targetted and no ot the box is available targetted using an a `accessibilityRole`)
children*	node	standard		Box accepts any co

Tokens

In exceptional circumstances, the following tokens can be passed to this component to override its default styles. **Do not do this unless absolutely necessary.** Read more about overriding styles.

▶ View Tokens

Variants

Please note that these variants support multiple brands, and must be used if you are building an application that is interoperable.

Background

Controls the background colour of the box. If not set, the box's background is transparent.

{ background: 'lightest' }

A pure white background.

background: 'lightest'

```
<Box variant={{ background: 'lightest' }} space={4}>
  <Typography>background: 'lightest'</Typography>
</Box>
```

{ background: 'lighter' }

A light grey background.

background: 'lighter'

```
<Box variant={{ background: 'lighter' }} space={4}>
  <Typography>background: 'lighter'</Typography>
</Box>
```

{ background: 'light' }

A subtle grey background.

background: 'light'

```
<Box variant={{ background: 'light' }} space={4}>
  <Typography>background: 'light'</Typography>
</Box>
```

{ background: 'dark' }

A heavy grey background. Content within this box should use the `inverse` variant where it is available.

background: 'dark'

```
<Box variant={{ background: 'dark' }} space={4}>
  <Typography variant={{ inverse: true }}>background: 'dark'</Typography>
</Box>
```

{ background: 'darker' }

A darker grey background.

background: 'darker'

```
<Box variant={{ background: 'darker' }} space={4}>
  <Typography variant={{ inverse: true }}>background: 'darker'</Typography>
</Box>
```

{ background: 'darkest' }

A very dark background. Content within this box should use the `inverse` variant where it is available.

background: 'darkest'

```
<Box variant={{ background: 'darkest' }} space={4}>
  <Typography variant={{ inverse: true }}>background:
'darkest'</Typography>
</Box>
```

{ background: 'critical' }

A dark red background. Use to alert users to a problem that requires immediate attention. Content within this box should use the `inverse` variant where it is available.

background: 'critical'

```
<Box variant={{ background: 'critical' }} space={4}>
  <Typography variant={{ inverse: true }}>background:
'critical'</Typography>
</Box>
```

{ background: 'danger' }

A pink / soft red background. Use carefully to alert users of a problem or warn that an action may have a cost such as costing money or losing data. Content within this box should use `danger` variants where they are available.

background: 'danger'

```
<Box variant={{ background: 'danger' }} space={4}>
  <Button variant={{ danger: true }} onPress={() => {}}>
    background: 'danger'
  </Button>
```

```
</Box>
```

{ background: 'warning' }

A yellow background. Use carefully to alert users to something requiring careful attention.

background: 'warning'

```
<Box variant={{ background: 'warning' }} space={4}>
  <Typography variant={{ bold: true }}>background:
'warning'</Typography>
</Box>
```

{ background: 'positive' }

A green background. Use when notifying users of a successful event, such as an action having completed successfully.

background: 'positive'

```
<Box variant={{ background: 'positive' }} space={4}>
  <Typography variant={{ bold: true }}>background:
'positive'</Typography>
</Box>
```

{ background: 'black' }

A black background. Limit use to align with marketing campaigns such as Black Friday.

background: 'black'

```
<Box variant={{ background: 'black' }} space={4}>
  <Typography variant={{ inverse: true }}>background:
'black'</Typography>
</Box>
```

{ backgroundColor: 'featurePrimary' }

Gradient: 'featurePrimary'

```
<Box variant={{ background: 'featurePrimary' }} space={4}>
  <Typography variant={{ inverse: true }}>Gradient:
'featurePrimary'</Typography>
</Box>
```

{ backgroundColor: 'featureSecondary' }

Gradient: 'featureSecondary'

```
<Box variant={{ background: 'featureSecondary' }} space={4}>
  <Typography variant={{ inverse: true }}>Gradient:
'featureSecondary'</Typography>
</Box>
```

{ backgroundColor: 'featureNeutral' }

Gradient: 'featureNeutral'

```
<Box space={15} variant={{ background: 'featureNeutral' }}
space={4}>
  <Typography>Gradient: 'featureNeutral'</Typography>
</Box>
```

{ backgroundColor: 'featureBrand' }

A brand background. Use sparingly as part of approved on-brand designs that create a bold visual impression. Content within this box should use the `inverse` variant where it is available.

background: 'featutureBrand'

```
<Box variant={{ background: 'featureBrand' }} space={4}>
  <Typography variant={{ inverse: true }}>background:
'featutureBrand'</Typography>
</Box>
```

Additional Variants

Please note that if you are building an application that is brand specific you can use these variants.

No brand specific variants

Feedback

- Spotted a problem with this component? Raise an issue on GitHub
- See any existing issues for this component
- Contact the team on slack in #ds-support