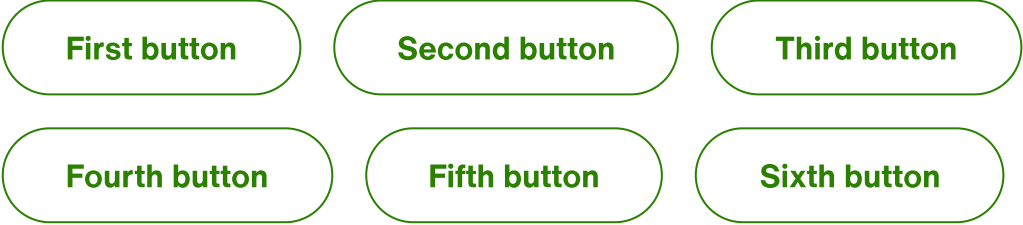


TELUS Component Library

StackWrap

Multi-Platform Component



```
<StackWrap space={3}>  
  <Button onPress={() => {}}>First button</Button>  
  <Button onPress={() => {}}>Second button</Button>  
  <Button onPress={() => {}}>Third button</Button>  
  <Button onPress={() => {}}>Fourth button</Button>  
  <Button onPress={() => {}}>Fifth button</Button>  
  <Button onPress={() => {}}>Sixth button</Button>  
</StackWrap>
```

Introduction

StackWrap is an alternative to [StackView](#) where the spaced items are allowed to wrap.

[Follow the appropriate instructions](#) to add this component in to your app.

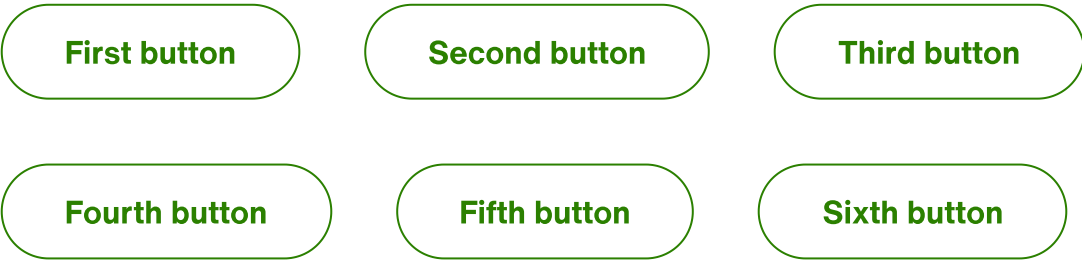
Guidance

Use StackWrap where elements should be evenly spaced, and wrap if they exceed the available space.

`space` prop

The `space` prop sets the size of spacing between items, using the [spacing scale](#).

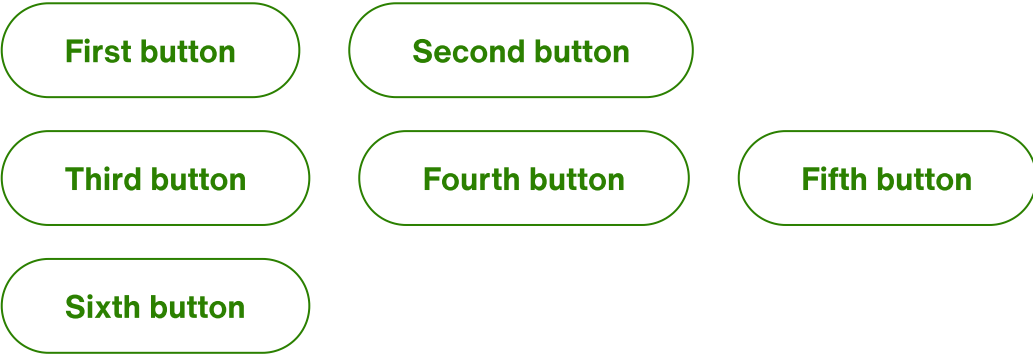
Simple numbers may be used, or objects (see [spacing scale docs](#) for details). These examples change spacing depending on the viewport.



```
<StackWrap space={{ xs: 1, md: 3, xl: 5 }}>
  <Button onPress={() => {}}>First button</Button>
  <Button onPress={() => {}}>Second button</Button>
  <Button onPress={() => {}}>Third button</Button>
  <Button onPress={() => {}}>Fourth button</Button>
  <Button onPress={() => {}}>Fifth button</Button>
  <Button onPress={() => {}}>Sixth button</Button>
</StackWrap>
```

gap prop

By default, the `space` value is applied to the gap between rows (or columns) of wrapped content as well as between individual items. If the gap between lines of content should have different spacing to the space between items, set the `gap` prop, which accepts the same spacing scale values as `space`.



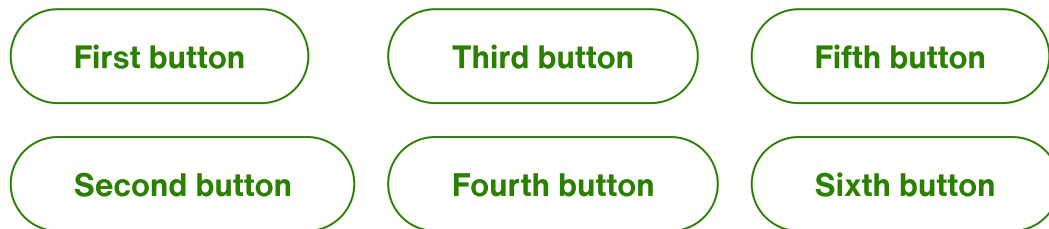
```
<StackWrap space={4} gap={{ xs: 1, md: 2, xl: 3 }}>
  <Button onPress={() => {}}>First button</Button>
  <Button onPress={() => {}}>Second button</Button>
  <Button onPress={() => {}}>Third button</Button>
  <Button onPress={() => {}}>Fourth button</Button>
  <Button onPress={() => {}}>Fifth button</Button>
</StackWrap>
```

```
<Button onPress={() => {}}>Sixth button</Button>  
</StackWrap>
```

direction prop

By default, StackWrap stacks items horizontally, with `flexDirection` of `'row'`.

In rare cases where wrapping columns are needed, this may be changed by passing `direction` of `'column'`.



```
<div style={{ display: 'flex', height: 150 }}>  
  <StackWrap space={3} direction="column">  
    <Button onPress={() => {}}>First button</Button>  
    <Button onPress={() => {}}>Second button</Button>  
    <Button onPress={() => {}}>Third button</Button>  
    <Button onPress={() => {}}>Fourth button</Button>  
    <Button onPress={() => {}}>Fifth button</Button>  
    <Button onPress={() => {}}>Sixth button</Button>  
  </StackWrap>  
</div>
```

Directions of `row-reverse` and `column-reverse` may be used to reverse the visual display of the stacked children.

The `direction` prop may also be passed a responsive object keyed by viewports.

Composition notes

Like StackView, StackWrap only spaces rendered, direct children, allowing common React composition patterns to be used without creating spurious space.

It uses the same insertion logic as `StackView`; see the [StackView docs](#) for more details and an example.

Accessibility

By default, `StackWrap` 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, `StackView` 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.

CONTENT

Nav tag

The container uses the nav tag.

CONTENT

Accessibi lity

Accessibility tools will interpret this section as a nav bar.

CONTENT

Semantic s

Crawlers like search engine bots may also interpret this content differently.

```
<StackWrap tag="nav" space={2}>
```

```
<StoryCard title="Nav tag" description="The container uses the
nav tag." href="/" tag="Content" />
<StoryCard
  title="Accessibility"
  description="Accessibility tools will interpret this section
as a nav bar."
  href="/"
  tag="Content"
/>
<StoryCard
  title="Semantics"
  description="Crawlers like search engine bots may also
interpret this content differently."
  href="/"
  tag="Content"
/>
</StackWrap>
```

Platform considerations

The component is available on both native platforms and web.

Props

Name	Type	Platform	Default	Description
tokens	tokens	standard		System tokens prop, see tokens for more details
direction	custom	standard		Sets the `flexDirection` of the container
space	union	standard		The size of the space between items according to the theme's spacing scale. Either a

Name	Type	Platform	Default	Description
				number corresponding to a position on the theme's spacing scale (1 is smallest, 2 is second smallest, etc), or, a SpacingObject with viewport keys and options (see <code>`useSpacingScale`</code> for details).
gap	union	standard		The size of the space between lines of content, according to the theme's spacing scale. By default, matches <code>`space`</code> . Either a number corresponding to a position on the theme's spacing scale (1 is smallest, 2 is second smallest, etc), or, a SpacingObject with viewport keys and options (see <code>`useSpacingScale`</code> for details).
tag	'h1' 'h2' 'h3' 'h4' 'h5' 'h6' 'article'	standard		Optional semantic HTML tag, to apply to the container on web. Does not

Name	Type	Platform	Default	Description
	'aside' 'blockquote' 'footer' 'figure' 'form' 'header' 'ul' 'li' 'main' 'nav' 'section' 'label'			change styling. In native apps, if a header tag is provided ("h1", "h2" etc) and accessibilityRole prop is not defined, the accessibilityRole will default to "heading".
children	node	standard		A StackWrap may take any children, but will have no effect if it is only passed one child or is passed children wrapped in a component. If necessary, children may be wrapped in one React Fragment.

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

This component does not have any stylistic variants.

To control flex alignment and positioning, the `tokens` prop may be passed an object to apply the flex layout styles `'alignItems'`, `'justifyContent'`, `'flexShrink'` and `'flexGrow'`.

As always with the `tokens` prop, this should be used only where strictly necessary to avoid creating off-brand layouts.

First item

Second item

Third item

Fourth item

Fifth item

Sixth item

```
<StackWrap
  space={6}
  tokens={{
    alignItems: 'baseline',
    justifyContent: 'flex-end',
    flexGrow: 1
  }}
>
  <Typography variant={{ size: 'large' }}>First
item</Typography>
  <Typography>Second item</Typography>
  <Typography variant={{ size: 'micro' }}>Third
item</Typography>
  <Typography variant={{ size: 'large' }}>Fourth
item</Typography>
  <Typography>Fifth item</Typography>
  <Typography variant={{ size: 'micro' }}>Sixth
item</Typography>
</StackWrap>
```

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](#)

