Card

- Usage
- Props
  - elevation: number
  - isBorderless: boolean
  - isRounded: boolean
  - size: string | object
  - Inherited props
- Sub-Components
  - Sub-Components Example
  - Logical Padding Properties
  - Context

`Card` provides a flexible and extensible content container.

Usage

`Card` also provides a convenient set of sub-components such as `CardBody`, `CardHeader`, `CardFooter`, and more (see below).

```
import {
    Card,
    CardHeader,
    CardBody,
    CardFooter,
    __experimentalText as Text,
    __experimentalHeading as Heading,
} from '@wordpress/components';

function Example() {
```

```
    return (
        <Card>
            <CardHeader>
                <Heading level={ 4 }>Card Title</Heading>
            </CardHeader>
            <CardBody>
                <Text>Card Content</Text>
            </CardBody>
            <CardFooter>
                <Text>Card Footer</Text>
            </CardFooter>
        </Card>
    );
}
```

Props

`elevation`: `number`

Size of the elevation shadow, based on the Style system's elevation system. This may be helpful in highlighting certain content. For more information, check out `Elevation`.

- Required: No
- Default: `0`

`isBorderless`: `boolean`

Renders without a border.

- Required: No
- Default: `false`

`isRounded`: `boolean`

Renders with rounded corners.

- Required: No
- Default: `true`

`size`: `string | object`

Determines the amount of padding within the component. Can be specified either as a single size token or as an object.

- Required: No
- Default: `medium`
- Allowed values:
  - Single size token: `none`, `xSmall`, `small`, `medium`, `large`
  - Object:

```
{
  blockStart: 'none' | 'xSmall' | 'small' | 'medium' | 'large';
  blockEnd: 'none' | 'xSmall' | 'small' | 'medium' | 'large';
  inlineStart: 'none' | 'xSmall' | 'small' | 'medium' | 'large';
  inlineEnd: 'none' | 'xSmall' | 'small' | 'medium' | 'large';
}
```

Inherited props

`Card` also inherits all of the `Surface` props.

Sub-Components

This component provides a collection of sub-component that can be used to compose various interfaces.

- `<CardBody />`
- `<CardDivider />`
- `<CardFooter />`
- `<CardHeader />`
- `<CardMedia />`

Sub-Components Example

```
import {
    Card,
    CardBody,
```

```
    CardDivider,
    CardFooter,
    CardHeader,
    CardMedia,
} from '@wordpress/components';

const Example = () => (
    <Card>
        <CardHeader>...</CardHeader>
        <CardBody>...</CardBody>
        <CardDivider />
        <CardBody>...</CardBody>
        <CardMedia>
            <img src="..." />
        </CardMedia>
        <CardFooter>...</CardFooter>
    </Card>
);
```

Logical Padding Properties

The `size` prop supports logical properties that adapt to different writing directions:

- `blockStart` – Maps to `top` in horizontal writing modes
- `blockEnd` – Maps to `bottom` in horizontal writing modes
- `inlineStart` – Maps to `left` in horizontal left-to-right writing modes
- `inlineEnd` – Maps to `right` in horizontal left-to-right writing modes

```
import { Card, CardBody } from '@wordpress/components';

const Example = () => (
    <Card
        size={ {
            blockStart: 'large',
            blockEnd: 'small',
            inlineStart: 'medium',
            inlineEnd: 'medium',
        } }
    >
        <CardBody
```

```
            size={ {
                blockStart: 'small',
                inlineStart: 'large',
                inlineEnd: 'large',
            } }
        >
            Content with logical padding properties
        </CardBody>
    </Card>
);
```

Context

`<Card />`'s sub-components are connected to `<Card />` using Context. Certain props like `size` and `isBorderless` are passed through to some of the sub-components.

In the following example, the `<CardBody />` will render with a size of `small`:

```
import { Card, CardBody } from '@wordpress/components';

const Example = () => (
    <Card size="small">
        <CardBody>...</CardBody>
    </Card>
);
```

These sub-components are designed to be flexible. The Context props can be overridden by the sub-component(s) as required. In the following example, the last `<CardBody />` will render its specified size:

```
import { Card, CardBody } from '@wordpress/components';

const Example = () => (
    <Card size="small">
        <CardBody>...</CardBody>
        <CardBody>...</CardBody>
        <CardBody size="large">...</CardBody>
    </Card>
);
```

□□□ #2
□ □□□ □□□ 2026-10-31 18:12:25 UTC
□ □□□ □□□ 2026-10-31 18:12:27 UTC