Dropdown

A dropdown list.

When To Use

When there are more than a few options to choose from, you can wrap them in a `Dropdown`. By hovering or clicking on the trigger, a dropdown menu will appear, which allows you to choose an option and execute the relevant action.

Examples

Hover me ⌄

Basic

The most basic dropdown menu.

```
import React from 'react';
import { DownOutlined, SmileOutlined } from '@
import type { MenuProps } from 'antd';
import { Dropdown, Space } from 'antd';

const items: MenuProps['items'] = [
  {
    key: '1',
    label: (
      <a target="_blank" rel="noopener norefer
        1st menu item
      </a>
    ),
  },
  {
    key: '2',
    label: (
      <a target="_blank" rel="noopener norefer
        2nd menu item (disabled)
      </a>
    ),
    icon: <SmileOutlined />,
    disabled: true,
  },
  {
    key: '3',
    label: (
      <a target="_blank" rel="noopener norefer
        3rd menu item (disabled)
      </a>
    ),
    disabled: true,
  },
  {
    key: '4',
    danger: true,
    label: 'a danger item',
  },
];

const App: React.FC = () => (
  <Dropdown menu={{ items }}>
    <a onClick={(e) => e.preventDefault()}>
      <Space>
        Hover me
        <DownOutlined />
      </Space>
    </a>
  </Dropdown>
);

export default App;
```

bottomLeft bottom bottomRight

topLeft top topRight

Placement

Support 6 placements.

```
import React from 'react';
import type { MenuProps } from 'antd';
import { Button, Dropdown, Space } from 'antd'

const items: MenuProps['items'] = [
  {
    key: '1',
    label: (
      <a target="_blank" rel="noopener norefer
        1st menu item
      </a>
    ),
  },
  {
    key: '2',
    label: (
      <a target="_blank" rel="noopener norefer
        2nd menu item
      </a>
    ),
  },
  {
    key: '3',
    label: (
      <a target="_blank" rel="noopener norefer
        3rd menu item
      </a>
    ),
  },
];

const App: React.FC = () => (
  <Space direction="vertical">
    <Space wrap>
      <Dropdown menu={{ items }} placement="bo
        <Button>bottomLeft</Button>
      </Dropdown>
      <Dropdown menu={{ items }} placement="bo
        <Button>bottom</Button>
      </Dropdown>
      <Dropdown menu={{ items }} placement="bo
        <Button>bottomRight</Button>
      </Dropdown>
    </Space>
    <Space wrap>
      <Dropdown menu={{ items }} placement="to
        <Button>topLeft</Button>
      </Dropdown>
      <Dropdown menu={{ items }} placement="to
        <Button>top</Button>
      </Dropdown>
      <Dropdown menu={{ items }} placement="to
        <Button>topRight</Button>
      </Dropdown>
    </Space>
  </Space>
);

export default App;
```

bottomLeft bottom bottomRight

topLeft top topRight

Arrow

You could display an arrow.

```
import React from 'react';
import type { MenuProps } from 'antd';
import { Button, Dropdown } from 'antd';

const items: MenuProps['items'] = [
  {
    key: '1',
    label: (
      <a target="_blank" rel="noopener norefer
        1st menu item
      </a>
    ),
  },
  {
    key: '2',
    label: (
      <a target="_blank" rel="noopener norefer
        2nd menu item
      </a>
    ),
  },
  {
    key: '3',
    label: (
      <a target="_blank" rel="noopener norefer
        3rd menu item
      </a>
    ),
  },
];

const App: React.FC = () => (
  <>
    <Dropdown menu={{ items }} placement="bott
      <Button>bottomLeft</Button>
    </Dropdown>
    <Dropdown menu={{ items }} placement="bott
      <Button>bottom</Button>
    </Dropdown>
    <Dropdown menu={{ items }} placement="bott
      <Button>bottomRight</Button>
    </Dropdown>
    <br />
    <Dropdown menu={{ items }} placement="topL
      <Button>topLeft</Button>
    </Dropdown>
    <Dropdown menu={{ items }} placement="top'
      <Button>top</Button>
    </Dropdown>
    <Dropdown menu={{ items }} placement="topR
      <Button>topRight</Button>
    </Dropdown>
  </>
);

export default App;
```

Hover me ∨

Other elements

Divider and disabled menu item.

```
import React from 'react';
import { DownOutlined } from '@ant-design/icon
import type { MenuProps } from 'antd';
import { Dropdown, Space } from 'antd';

const items: MenuProps['items'] = [
  {
    label: (
      <a target="_blank" rel="noopener norefer
        1st menu item
      </a>
    ),
    key: '0',
  },
  {
    label: (
      <a target="_blank" rel="noopener norefer
        2nd menu item
      </a>
    ),
    key: '1',
  },
  {
    type: 'divider',
  },
  {
    label: '3rd menu item (disabled) ',
    key: '3',
    disabled: true,
  },
];

const App: React.FC = () => (
  <Dropdown menu={{ items }}>
    <a onClick={(e) => e.preventDefault()}>
      <Space>
        Hover me
        <DownOutlined />
      </Space>
    </a>
  </Dropdown>
);

export default App;
```

bottomLeft bottom bottomRight

topLeft top topRight

Arrow pointing at the center

By specifying `arrow` prop with `{ pointAtCenter: true }`, the arrow will point to the center of the target element.

```
import React from 'react';
import type { MenuProps } from 'antd';
import { Button, Dropdown } from 'antd';

const items: MenuProps['items'] = [
  {
    key: '1',
    label: (
      <a target="_blank" rel="noopener norefer
        1st menu item
      </a>
    ),
  },
  {
    key: '2',
    label: (
      <a target="_blank" rel="noopener norefer
        2nd menu item
      </a>
    ),
  },
  {
    key: '3',
    label: (
      <a target="_blank" rel="noopener norefer
        3rd menu item
      </a>
    ),
  },
];

const App: React.FC = () => (
  <>
    <Dropdown menu={{ items }} placement="bott
      <Button>bottomLeft</Button>
    </Dropdown>
    <Dropdown menu={{ items }} placement="bott
      <Button>bottom</Button>
    </Dropdown>
    <Dropdown menu={{ items }} placement="bott
      <Button>bottomRight</Button>
    </Dropdown>
    <br />
    <Dropdown menu={{ items }} placement="topL
      <Button>topLeft</Button>
    </Dropdown>
    <Dropdown menu={{ items }} placement="top'
      <Button>top</Button>
    </Dropdown>
    <Dropdown menu={{ items }} placement="topR
      <Button>topRight</Button>
    </Dropdown>
  </>
);

export default App;
```

Click me ∨

Trigger mode

The default trigger mode is `hover`, you can change it to `click`.

```
import React from 'react';
import { DownOutlined } from '@ant-design/icor
import type { MenuProps } from 'antd';
import { Dropdown, Space } from 'antd';

const items: MenuProps['items'] = [
  {
    label: <a href="https://www.antgroup.com">
    key: '0',
  },
  {
    label: <a href="https://www.aliyun.com">2r
    key: '1',
  },
  {
    type: 'divider',
  },
  {
    label: '3rd menu item',
    key: '3',
  },
];

const App: React.FC = () => (
  <Dropdown menu={{ items }} trigger={['click'
    <a onClick={(e) => e.preventDefault()}>
      <Space>
        Click me
        <DownOutlined />
      </Space>
    </a>
  </Dropdown>
);

export default App;
```

Hover me, Click menu item ∨

Click event

An event will be triggered when you click menu items, in which you can make different operations according to item's key.

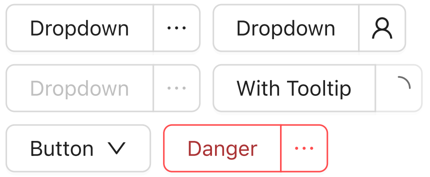
```
import React from 'react';
import { DownOutlined } from '@ant-design/icon
import type { MenuProps } from 'antd';
import { Dropdown, message, Space } from 'antd

const onClick: MenuProps['onClick'] = ({ key }
  message.info(`Click on item ${key}`);
};

const items: MenuProps['items'] = [
  {
    label: '1st menu item',
    key: '1',
  },
  {
    label: '2nd menu item',
    key: '2',
  },
  {
    label: '3rd menu item',
    key: '3',
  },
];

const App: React.FC = () => (
  <Dropdown menu={{ items, onClick }}>
    <a onClick={(e) => e.preventDefault()}>
      <Space>
        Hover me, Click menu item
        <DownOutlined />
      </Space>
    </a>
  </Dropdown>
);

export default App;
```



Button with dropdown menu

A button is on the left, and a related functional menu is on the right. You can set the icon property to modify the icon of right.

```
import React from 'react';
import { DownOutlined, UserOutlined } from '@a
import type { MenuProps } from 'antd';
import { Button, Dropdown, message, Space, Too

const handleButtonClick = (e: React.MouseEvent
  message.info('Click on left button.');
  console.log('click left button', e);
};

const handleMenuClick: MenuProps['onClick'] =
  message.info('Click on menu item.');
  console.log('click', e);
};

const items: MenuProps['items'] = [
  {
    label: '1st menu item',
    key: '1',
    icon: <UserOutlined />,
  },
  {
    label: '2nd menu item',
    key: '2',
    icon: <UserOutlined />,
  },
  {
    label: '3rd menu item',
    key: '3',
    icon: <UserOutlined />,
    danger: true,
  },
  {
    label: '4rd menu item',
    key: '4',
    icon: <UserOutlined />,
    danger: true,
    disabled: true,
  },
];

const menuProps = {
  items,
  onClick: handleMenuClick,
};

const App: React.FC = () => (
  <Space wrap>
    <Dropdown.Button menu={menuProps} onClick=
      Dropdown
    </Dropdown.Button>
    <Dropdown.Button menu={menuProps} placemen
      Dropdown
    </Dropdown.Button>
    <Dropdown.Button menu={menuProps} onClick=
      Dropdown
    </Dropdown.Button>
```

Hover me ∨

Custom dropdown

Customize the dropdown menu via `dropdownRender`. If you don't need the Menu content, use the Popover component directly.

```
import React from 'react';
import { DownOutlined } from '@ant-design/icor
import { Dropdown, Space, Divider, Button, the
import type { MenuProps } from 'antd';

const { useToken } = theme;

const items: MenuProps['items'] = [
  {
    key: '1',
    label: (
      <a target="_blank" rel="noopener norefer
        1st menu item
      </a>
    ),
  },
  {
    key: '2',
    label: (
      <a target="_blank" rel="noopener norefer
        2nd menu item (disabled)
      </a>
    ),
    disabled: true,
  },
  {
    key: '3',
    label: (
      <a target="_blank" rel="noopener norefer
        3rd menu item (disabled)
      </a>
    ),
    disabled: true,
  },
];

const App: React.FC = () => {
  const { token } = useToken();

  const contentStyle = {
    backgroundColor: token.colorBgElevated,
    borderRadius: token.borderRadiusLG,
    boxShadow: token.boxShadowSecondary,
  };

  const menuStyle = {
    boxShadow: 'none',
  };

  return (
    <Dropdown
      menu={{ items }}
      dropdownRender={(menu) => (
        <div style={contentStyle}>
          {React.cloneElement(menu as React.Re
          <Divider style={{ margin: 0 }} />
          <Space style={{ padding: 8 }}>
            <Button type="primary">Click me!</
          </Space>
        </div>
      )}
      >
```

Cascading menu ∨

Cascading menu

The menu has multiple levels.

```
import React from 'react';
import { DownOutlined } from '@ant-design/icor
import type { MenuProps } from 'antd';
import { Dropdown, Space } from 'antd';

const items: MenuProps['items'] = [
  {
    key: '1',
    type: 'group',
    label: 'Group title',
    children: [
      {
        key: '1-1',
        label: '1st menu item',
      },
      {
        key: '1-2',
        label: '2nd menu item',
      },
    ],
  },
  {
    key: '2',
    label: 'sub menu',
    children: [
      {
        key: '2-1',
        label: '3rd menu item',
      },
      {
        key: '2-2',
        label: '4th menu item',
      },
    ],
  },
  {
    key: '3',
    label: 'disabled sub menu',
    disabled: true,
    children: [
      {
        key: '3-1',
        label: '5d menu item',
      },
      {
        key: '3-2',
        label: '6th menu item',
      },
    ],
  },
];

const App: React.FC = () => (
  <Dropdown menu={{ items }}>
    <a onClick={(e) => e.preventDefault()}>
      <Space>
        Cascading menu
        <DownOutlined />
      </Space>
    </a>
  </Dropdown>
);

export default App;
```

Hover me ∨

The way of hiding menu.

The default is to close the menu when you click on menu items, this feature can be turned off.

```
import React, { useState } from 'react';
import { DownOutlined } from '@ant-design/icon
import type { MenuProps } from 'antd';
import { Dropdown, Space } from 'antd';

const App: React.FC = () => {
  const [open, setOpen] = useState(false);

  const handleMenuClick: MenuProps['onClick']
    if (e.key === '3') {
      setOpen(false);
    }
  };

  const handleOpenChange = (flag: boolean) =>
    setOpen(flag);
  };

  const items: MenuProps['items'] = [
    {
      label: 'Clicking me will not close the m
      key: '1',
    },
    {
      label: 'Clicking me will not close the m
      key: '2',
    },
    {
      label: 'Clicking me will close the menu.
      key: '3',
    },
  ];

  return (
    <Dropdown
      menu={{
        items,
        onClick: handleMenuClick,
      }}
      onOpenChange={handleOpenChange}
      open={open}
    >
      <a onClick={(e) => e.preventDefault()}>
        <Space>
          Hover me
          <DownOutlined />
        </Space>
      </a>
    </Dropdown>
  );
};

export default App;
```

Right Click on here

Context Menu

The default trigger mode is `hover`, you can change it to `contextMenu`.

```
import React from 'react';
import type { MenuProps } from 'antd';
import { Dropdown, theme } from 'antd';

const items: MenuProps['items'] = [
  {
    label: '1st menu item',
    key: '1',
  },
  {
    label: '2nd menu item',
    key: '2',
  },
  {
    label: '3rd menu item',
    key: '3',
  },
];

const App: React.FC = () => {
  const {
    token: { colorBgLayout, colorTextTertiary
  } = theme.useToken();

  return (
    <Dropdown menu={{ items }} trigger={['cont
      <div
        style={{
          color: colorTextTertiary,
          background: colorBgLayout,
          height: 200,
          textAlign: 'center',
          lineHeight: '200px',
        }}
      >
        Right Click on here
      </div>
    </Dropdown>
  );
};

export default App;
```

Submit

Submit

Submit

Submit ⌄

Loading

A loading indicator can be added to a button by setting the `loading` property on the `Dropdown.Button`.

```
import React, { useState } from 'react';
import { DownOutlined } from '@ant-design/icor
import type { MenuProps } from 'antd';
import { Dropdown, Space } from 'antd';

const items: MenuProps['items'] = [
  {
    label: 'Submit and continue',
    key: '1',
  },
];

const App: React.FC = () => {
  const [loadings, setLoadings] = useState<boc

  const enterLoading = (index: number) => {
    setLoadings((state) => {
      const newLoadings = [...state];
      newLoadings[index] = true;
      return newLoadings;
    });

    setTimeout(() => {
      setLoadings((state) => {
        const newLoadings = [...state];
        newLoadings[index] = false;
        return newLoadings;
      });
    }, 6000);
  };

  return (
    <Space direction="vertical">
      <Dropdown.Button type="primary" loading
        Submit
      </Dropdown.Button>
      <Dropdown.Button type="primary" size="sn
        Submit
      </Dropdown.Button>
      <Dropdown.Button
        type="primary"
        loading={loadings[0]}
        menu={{ items }}
        onClick={() => enterLoading(0)}
      >
        Submit
      </Dropdown.Button>
      <Dropdown.Button
        icon={<DownOutlined />}
        loading={loadings[1]}
        menu={{ items }}
        onClick={() => enterLoading(1)}
      >
        Submit
      </Dropdown.Button>
    </S
```

Selectable ⌄

Selectable Menu

Configure the `selectable` property in `menu` to enable selectable ability.

```
import React from 'react';
import { DownOutlined } from '@ant-design/icor
import type { MenuProps } from 'antd';
import { Dropdown, Space, Typography } from 'c

const items: MenuProps['items'] = [
  {
    key: '1',
    label: 'Item 1',
  },
  {
    key: '2',
    label: 'Item 2',
  },
  {
    key: '3',
    label: 'Item 3',
  },
];

const App: React.FC = () => (
  <Dropdown
    menu={{
      items,
      selectable: true,
      defaultSelectedKeys: ['3'],
    }}
  >
    <Typography.Link>
      <Space>
        Selectable
        <DownOutlined />
      </Space>
    </Typography.Link>
  </Dropdown>
);

export default App;
```

```
    </Space>
  );
};

export default App;
```

Dropdown

Property	Description	Type	Defau
arrow	Whether the dropdown arrow should be visible	boolean | { pointAtCenter: boolean }	false
autoAdjustOverflow	Whether to adjust dropdown placement automatically when dropdown is off screen	boolean	true
autoFocus	Focus element in `overlay` when opened	boolean	false
disabled	Whether the dropdown menu is disabled	boolean	-
destroyPopupOnHide	Whether destroy dropdown when hidden	boolean	false
dropdownRender	Customize dropdown content	(menus: ReactNode) => ReactNode	-
getPopupContainer	To set the container of the dropdown menu. The default is to create a div element in body, but you can reset it to the scrolling area and make a relative reposition. [Example on CodePen]()	(triggerNode: HTMLElement) => HTMLElement	() => docum
menu	The menu props	[MenuProps]()	-
overlayClassName	The class name of the dropdown root element	string	-
overlayStyle	The style of the dropdown root element	CSSProperties	-
placement	Placement of popup menu: `bottom` `bottomLeft` `bottomRight` `top` `topLeft` `topRight`	string	`botto`
trigger	The trigger mode which executes the dropdown action. Note that hover	Array<`click` | `hover` | `contextMenu`>	[`hove`

Property	Description	Type	Defau
	can't be used on touchscreens		
open	Whether the dropdown menu is currently open. Use `visible` under 4.23.0 (why?)	boolean	-
onOpenChange	Called when the open state is changed. Not trigger when hidden by click item. Use `onVisibleChange` under 4.23.0 (why?)	(open: boolean) => void	-

Dropdown.Button

Same props from Dropdown. And includes additional props:

Property	Description	Type	Default	Version
buttonsRender	Custom buttons inside Dropdown.Button	(buttons: ReactNode[]) => ReactNode[]	-	
loading	Set the loading status of button	boolean | { delay: number }	false	
danger	Set the danger status of button	boolean	-	4.23.0
icon	Icon (appears on the right)	ReactNode	-	
size	Size of the button, the same as Button	string	`default`	
type	Type of the button, the same as Button	string	`default`	
onClick	The same as Button: called when you click the button on the left	(event) => void	-	

Note

Please ensure that the child node of `Dropdown` accepts `onMouseEnter`, `onMouseLeave`, `onFocus`, `onClick` events.

Design Token

▼ Global Token

Token Name	Description	Type	Default Value
colorBgElevated	Container background color of the popup layer, in dark mode the color value of this token will be a little brighter than `colorBgContainer`. E.g: modal, pop-up, menu, etc.	`string`	□ `#ffffff`
colorError	Used to represent the visual elements of the operation failure, such as the error Button, error Result component, etc.	`string`	□ `#ff4d4f`
colorPrimary	Brand color is one of the most direct visual elements to reflect the characteristics and communication of the product. After you have selected the brand color, we will automatically generate a complete color palette and assign it effective design semantics.	`string`	□ `#1677ff`
colorPrimaryBorder	The stroke color under the main color gradient, used on the stroke of components such as Slider.	`string`	□ `#91caff`
colorSplit	Used as the color of separator, this color is the same as colorBorderSecondary but with transparency.	`string`	□ `rgba(5, 5, 5, 0.06)`
colorText	Default text color which comply with W3C standards, and this color is also the darkest neutral color.	`string`	□ `rgba(0, 0, 0, 0.88)`
colorTextDescription	Control the font color of text description.	`string`	□ `rgba(0, 0, 0, 0.45)`
colorTextDisabled	Control the color of text in disabled state.	`string`	□ `rgba(0, 0, 0, 0.25)`
colorTextLightSolid	Control the highlight color of text with background color, such as the text in Primary Button components.	`string`	□ `#fff`
borderRadiusLG	LG size border radius, used in some large border radius components, such as Card, Modal and other components.	`number`	8
borderRadiusOuter	Outer border radius	`number`	4
borderRadiusSM	SM size border radius, used in small size components, such as Button, Input, Select and other input components in small size	`number`	4
borderRadiusXS	XS size border radius, used in some small border radius components, such as Segmented, Arrow and other components with small border radius.	`number`	2
boxShadowSecondary	Control the secondary box shadow style of an element.	`string`	0 6px 16px 0 rgba(0, 0, 0, 0.08), 0 3px 6px -4px rgba(0, 0, 0, 0.12), 0 9px 28px 8px rgba(0, 0, 0, 0.05)

Token Name	Description	Type	Default Value
controlHeight	The height of the basic controls such as buttons and input boxes in Ant Design	`number`	32
controlItemBgActive	Control the background color of control component item when active.	`string`	□ `#e6f4ff`
controlItemBgActiveHover	Control the background color of control component item when hovering and active.	`string`	□ `#bae0ff`
controlItemBgHover	Control the background color of control component item when hovering.	`string`	□ `rgba(0, 0, 0, 0.04)`
controlPaddingHorizontal	Control the horizontal padding of an element.	`number`	12
fontFamily	The font family of Ant Design prioritizes the default interface font of the system, and provides a set of alternative font libraries that are suitable for screen display to maintain the readability and readability of the font under different platforms and browsers, reflecting the friendly, stable and professional characteristics.	`string`	-apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue', Arial, 'Noto Sans', sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji'
fontSize	The most widely used font size in the design system, from which the text gradient will be derived.	`number`	14
fontSizeIcon	Control the font size of operation icon in Select, Cascader, etc. Normally same as fontSizeSM.	`number`	12
fontSizeSM	Small font size	`number`	12
lineHeight	Line height of text.	`number`	1.5714285714285714
lineWidthFocus	Control the width of the line when the component is in focus state.	`number`	4
marginXS	Control the margin of an element, with a small size.	`number`	8
marginXXS	Control the margin of an element, with the smallest size.	`number`	4
motionDurationMid	Motion speed, medium speed. Used for medium element animation interaction.	`string`	0.2s
motionEaseInOutCirc	Preset motion curve.	`string`	cubic-bezier(0.78, 0.14, 0.15, 0.86)
motionEaseInQuint	Preset motion curve.	`string`	cubic-bezier(0.755, 0.05, 0.855, 0.06)
motionEaseOutCirc	Preset motion curve.	`string`	cubic-bezier(0.08, 0.82, 0.17, 1)
motionEaseOutQuint	Preset motion curve.	`string`	cubic-bezier(0.23, 1, 0.32, 1)

Token Name	Description	Type	Default Value
paddingXS	Control the extra small padding of the element.	`number`	8
paddingXXS	Control the extra extra small padding of the element.	`number`	4
sizePopupArrow	The size of the component arrow	`number`	16
zIndexPopupBase	Base zIndex of component like FloatButton, Affix which can be cover by large popup	`number`	1000