

何时使用 {#when-to-use}

抽屉从父窗体边缘滑入，覆盖住部分父窗体内容。用户在抽屉内操作时不必离开当前任务，操作完成后，可以平滑地回到原任务。

- 当需要一个附加的面板来控制父窗体内容，这个面板在需要时呼出。比如，控制界面展示样式，往界面中添加内容。
- 当需要在当前任务流中插入临时任务，创建或预览附加内容。比如展示协议条款，创建子对象。

开发者注意事项：

自 5.17.0 版本，我们提供了 `loading` 属性，内置 `Spin` 组件作为加载状态，但是自 5.18.0 版本开始，我们修复了设计失误，将内置的 `Spin` 组件替换成了 `Skeleton` 组件，同时收窄了 `loading` api 的类型范围，只能接收 `boolean` 类型。

代码演示

基础抽屉

```
import React, { useState } from 'react';
import { Button, Drawer } from 'antd';

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

  const showDrawer = () => {
    setOpen(true);
  };

  const onClose = () => {
    setOpen(false);
  };

  return (
    <>
      <Button type="primary" onClick={showDrawer}>
        Open
      </Button>
      <Drawer title="Basic Drawer" onClose={onClose} open={open}>
        <p>Some contents...</p>
        <p>Some contents...</p>
        <p>Some contents...</p>
      </Drawer>
    </>
  );
};
```

```
export default App;
```

自定义位置

```
import React, { useState } from 'react';
import type { DrawerProps, RadioChangeEvent } from 'antd';
import { Button, Drawer, Radio, Space } from 'antd';

const App: React.FC = () => {
  const [open, setOpen] = useState(false);
  const [placement, setPlacement] = useState<DrawerProps['placement']>('left');

  const showDrawer = () => {
    setOpen(true);
  };

  const onClose = () => {
    setOpen(false);
  };

  const onChange = (e: RadioChangeEvent) => {
    setPlacement(e.target.value);
  };

  return (
    <>
      <Space>
        <Radio.Group value={placement} onChange={onChange}>
          <Radio value="top">top</Radio>
          <Radio value="right">right</Radio>
          <Radio value="bottom">bottom</Radio>
          <Radio value="left">left</Radio>
        </Radio.Group>
        <Button type="primary" onClick={showDrawer}>
          Open
        </Button>
      </Space>
      <Drawer
        title="Basic Drawer"
        placement={placement}
        closable={false}
        onClose={onClose}
        open={open}
        key={placement}
      />
    </>
  );
};
```

```

    >
    <p>Some contents...</p>
    <p>Some contents...</p>
    <p>Some contents...</p>
  </Drawer>
</>
);
};

export default App;

```

加载中

v5.17.0

```

import React from 'react';
import { Button, Drawer } from 'antd';

const App: React.FC = () => {
  const [open, setOpen] = React.useState<boolean>(false);
  const [loading, setLoading] = React.useState<boolean>(true);

  const showLoading = () => {
    setOpen(true);
    setLoading(true);

    // Simple loading mock. You should add cleanup logic in real world.
    setTimeout(() => {
      setLoading(false);
    }, 2000);
  };

  return (
    <>
      <Button type="primary" onClick={showLoading}>
        Open Drawer
      </Button>
      <Drawer
        closable
        destroyOnClose
        title=<p>Loading Drawer</p>
        placement="right"
        open={open}
        loading={loading}
        onClose={() => setOpen(false)}
      >

```

```

        <Button type="primary" style={{ marginBottom: 16 }} onClick=
{showLoading}>
          Reload
        </Button>
        <p>Some contents...</p>
        <p>Some contents...</p>
        <p>Some contents...</p>
      </Drawer>
    </>
  );
};

export default App;

```

额外操作

```

import React, { useState } from 'react';
import { Button, Drawer, Radio, Space } from 'antd';
import type { DrawerProps, RadioChangeEvent } from 'antd';

const App: React.FC = () => {
  const [open, setOpen] = useState(false);
  const [placement, setPlacement] = useState<DrawerProps['placement']>
('right');

  const showDrawer = () => {
    setOpen(true);
  };

  const onChange = (e: RadioChangeEvent) => {
    setPlacement(e.target.value);
  };

  const onClose = () => {
    setOpen(false);
  };

  return (
    <>
      <Space>
        <Radio.Group value={placement} onChange={onChange}>
          <Radio value="top">top</Radio>
          <Radio value="right">right</Radio>
          <Radio value="bottom">bottom</Radio>
          <Radio value="left">left</Radio>
        </Radio.Group>

```

```

    <Button type="primary" onClick={showDrawer}>
      Open
    </Button>
  </Space>
  <Drawer
    title="Drawer with extra actions"
    placement={placement}
    width={500}
    onClose={onClose}
    open={open}
    extra={
      <Space>
        <Button onClick={onClose}>Cancel</Button>
        <Button type="primary" onClick={onClose}>
          OK
        </Button>
      </Space>
    }
  >
    <p>Some contents...</p>
    <p>Some contents...</p>
    <p>Some contents...</p>
  </Drawer>
</>
);
};

export default App;

```

渲染在当前 DOM

```

import React, { useState } from 'react';
import { Button, Drawer, theme } from 'antd';

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

  const showDrawer = () => {
    setOpen(true);
  };

  const onClose = () => {
    setOpen(false);
  };

```

```

const containerStyle: React.CSSProperties = {
  position: 'relative',
  height: 200,
  padding: 48,
  overflow: 'hidden',
  background: token.colorFillAlter,
  border: `1px solid ${token.colorBorderSecondary}`,
  borderRadius: token.borderRadiusLG,
};

return (
  <div style={containerStyle}>
    Render in this
    <div style={{ marginTop: 16 }}>
      <Button type="primary" onClick={showDrawer}>
        Open
      </Button>
    </div>
    <Drawer
      title="Basic Drawer"
      placement="right"
      closable={false}
      onClose={onClose}
      open={open}
      getContainer={false}
    >
      <p>Some contents...</p>
    </Drawer>
  </div>
);
};

export default App;

```

抽屉表单

```

import React, { useState } from 'react';
import { PlusOutlined } from '@ant-design/icons';
import { Button, Col, DatePicker, Drawer, Form, Input, Row, Select, Space }
from 'antd';

const { Option } = Select;

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

```

```

const showDrawer = () => {
  setOpen(true);
};

const onClose = () => {
  setOpen(false);
};

return (
  <>
    <Button type="primary" onClick={showDrawer} icon={<PlusOutlined />>
      New account
    </Button>
    <Drawer
      title="Create a new account"
      width={720}
      onClose={onClose}
      open={open}
      styles={{
        body: {
          paddingBottom: 80,
        },
      }}
      extra={
        <Space>
          <Button onClick={onClose}>Cancel</Button>
          <Button onClick={onClose} type="primary">
            Submit
          </Button>
        </Space>
      }
    >
      <Form layout="vertical" hideRequiredMark>
        <Row gutter={16}>
          <Col span={12}>
            <Form.Item
              name="name"
              label="Name"
              rules={[{ required: true, message: 'Please enter user name'
}}}
              >
                <Input placeholder="Please enter user name" />
              </Form.Item>
            </Col>
            <Col span={12}>
              <Form.Item

```

```

        name="url"
        label="Url"
        rules={[{ required: true, message: 'Please enter url' }]}
      >
        <Input
          style={{ width: '100%' }}
          addonBefore="http://"
          addonAfter=".com"
          placeholder="Please enter url"
        />
      </Form.Item>
    </Col>
  </Row>
  <Row gutter={16}>
    <Col span={12}>
      <Form.Item
        name="owner"
        label="Owner"
        rules={[{ required: true, message: 'Please select an owner'
}}}
      >
        <Select placeholder="Please select an owner">
          <Option value="xiao">Xiaoxiao Fu</Option>
          <Option value="mao">Maomao Zhou</Option>
        </Select>
      </Form.Item>
    </Col>
    <Col span={12}>
      <Form.Item
        name="type"
        label="Type"
        rules={[{ required: true, message: 'Please choose the type'
}}}
      >
        <Select placeholder="Please choose the type">
          <Option value="private">Private</Option>
          <Option value="public">Public</Option>
        </Select>
      </Form.Item>
    </Col>
  </Row>
  <Row gutter={16}>
    <Col span={12}>
      <Form.Item
        name="approver"
        label="Approver"

```



```

        rules={[{ required: true, message: 'Please choose the
approver' }]}
      >
        <Select placeholder="Please choose the approver">
          <Option value="jack">Jack Ma</Option>
          <Option value="tom">Tom Liu</Option>
        </Select>
      </Form.Item>
    </Col>
    <Col span={12}>
      <Form.Item
        name="dateTime"
        label="DateTime"
        rules={[{ required: true, message: 'Please choose the
dateTime' }]}
      >
        <DatePicker.RangePicker
          style={{ width: '100%' }}
          getPopupContainer={(trigger) => trigger.parentElement!}
        />
      </Form.Item>
    </Col>
  </Row>
  <Row gutter={16}>
    <Col span={24}>
      <Form.Item
        name="description"
        label="Description"
        rules={[
          {
            required: true,
            message: 'please enter url description',
          },
        ]}
      >
        <Input.TextArea rows={4} placeholder="please enter url
description" />
      </Form.Item>
    </Col>
  </Row>
</Form>
</Drawer>
</>
);
};

```

```
export default App;
```

信息预览抽屉

```
import React, { useState } from 'react';
import { Avatar, Col, Divider, Drawer, List, Row } from 'antd';

interface DescriptionItemProps {
  title: string;
  content: React.ReactNode;
}

const DescriptionItem = ({ title, content }: DescriptionItemProps) => (
  <div className="site-description-item-profile-wrapper">
    <p className="site-description-item-profile-p-label">{title}</p>
    {content}
  </div>
);

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

  const showDrawer = () => {
    setOpen(true);
  };

  const onClose = () => {
    setOpen(false);
  };

  return (
    <>
      <List
        dataSource={[
          {
            id: 1,
            name: 'Lily',
          },
          {
            id: 2,
            name: 'Lily',
          },
        ]}
        bordered
        renderItem={(item) => (
```

```

<List.Item
  key={item.id}
  actions={[
    <a onClick={showDrawer} key={`a-${item.id}`}>
      View Profile
    </a>,
  ]}
>
  <List.Item.Meta
    avatar={
      <Avatar
src="https://gw.alipayobjects.com/zos/rmsportal/BiazfanxmamNRoxxVxka.png"
/>
    }
    title={<a href="https://ant.design/index-cn">{item.name}</a>}
    description="Progresser XTech"
  </List.Item>
)}
/>
<Drawer width={640} placement="right" closable={false} onClose=
{onClose} open={open}>
  <p className="site-description-item-profile-p" style={{
marginBottom: 24 }}>
    User Profile
  </p>
  <p className="site-description-item-profile-p">Personal</p>
  <Row>
    <Col span={12}>
      <DescriptionItem title="Full Name" content="Lily" />
    </Col>
    <Col span={12}>
      <DescriptionItem title="Account"
content="AntDesign@example.com" />
    </Col>
  </Row>
  <Row>
    <Col span={12}>
      <DescriptionItem title="City" content="HangZhou" />
    </Col>
    <Col span={12}>
      <DescriptionItem title="Country" content="China🇨🇳" />
    </Col>
  </Row>
  <Row>
    <Col span={12}>

```

```

        <DescriptionItem title="Birthday" content="February 2,1900" />
    </Col>
    <Col span={12}>
        <DescriptionItem title="Website" content="-" />
    </Col>
</Row>
<Row>
    <Col span={24}>
        <DescriptionItem
            title="Message"
            content="Make things as simple as possible but no simpler."
        />
    </Col>
</Row>
<Divider />
<p className="site-description-item-profile-p">Company</p>
<Row>
    <Col span={12}>
        <DescriptionItem title="Position" content="Programmer" />
    </Col>
    <Col span={12}>
        <DescriptionItem title="Responsibilities" content="Coding" />
    </Col>
</Row>
<Row>
    <Col span={12}>
        <DescriptionItem title="Department" content="XTech" />
    </Col>
    <Col span={12}>
        <DescriptionItem title="Supervisor" content={<a>Lin</a>} />
    </Col>
</Row>
<Row>
    <Col span={24}>
        <DescriptionItem
            title="Skills"
            content="C / C + +, data structures, software engineering,
operating systems, computer networks, databases, compiler theory, computer
architecture, Microcomputer Principle and Interface Technology, Computer
English, Java, ASP, etc."
        />
    </Col>
</Row>
<Divider />
<p className="site-description-item-profile-p">Contacts</p>
<Row>

```

```

        <Col span={12}>
          <DescriptionItem title="Email" content="AntDesign@example.com"
/>

        </Col>
        <Col span={12}>
          <DescriptionItem title="Phone Number" content="+86 181 0000
0000" />
        </Col>
      </Row>
      <Row>
        <Col span={24}>
          <DescriptionItem
            title="Github"
            content={
              <a href="http://github.com/ant-design/ant-design/"
target="_blank" rel="noreferrer">
                github.com/ant-design/ant-design/
              </a>
            }
          />
        </Col>
      </Row>
    </Drawer>
  </>
);
};

export default App;

```

多层抽屉

```

import React, { useState } from 'react';
import { Button, Drawer } from 'antd';

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

  const showDrawer = () => {
    setOpen(true);
  };

  const onClose = () => {
    setOpen(false);
  };

```

```

const showChildrenDrawer = () => {
  setChildrenDrawer(true);
};

const onChildrenDrawerClose = () => {
  setChildrenDrawer(false);
};

return (
  <>
    <Button type="primary" onClick={showDrawer}>
      Open drawer
    </Button>
    <Drawer title="Multi-level drawer" width={520} closable={false}
onClose={onClose} open={open}>
      <Button type="primary" onClick={showChildrenDrawer}>
        Two-level drawer
      </Button>
      <Drawer
        title="Two-level Drawer"
        width={320}
        closable={false}
        onClose={onChildrenDrawerClose}
        open={childrenDrawer}
      >
        This is two-level drawer
      </Drawer>
    </Drawer>
  </>
);
};

export default App;

```

预设宽度

```

import React, { useState } from 'react';
import { Button, Drawer, Space } from 'antd';
import type { DrawerProps } from 'antd';

const App: React.FC = () => {
  const [open, setOpen] = useState(false);
  const [size, setSize] = useState<DrawerProps['size']>();

  const showDefaultDrawer = () => {
    setSize('default');
  };

```

```

    setOpen(true);
  };

  const showLargeDrawer = () => {
    setSize('large');
    setOpen(true);
  };

  const onClose = () => {
    setOpen(false);
  };

  return (
    <>
      <Space>
        <Button type="primary" onClick={showDefaultDrawer}>
          Open Default Size (378px)
        </Button>
        <Button type="primary" onClick={showLargeDrawer}>
          Open Large Size (736px)
        </Button>
      </Space>
      <Drawer
        title={` ${size} Drawer`}
        placement="right"
        size={size}
        onClose={onClose}
        open={open}
        extra={
          <Space>
            <Button onClick={onClose}>Cancel</Button>
            <Button type="primary" onClick={onClose}>
              OK
            </Button>
          </Space>
        }
      >
        <p>Some contents...</p>
        <p>Some contents...</p>
        <p>Some contents...</p>
      </Drawer>
    </>
  );
};

export default App;

```

自定义内部样式

```
import React, { useState } from 'react';
import { Button, ConfigProvider, Drawer, Space } from 'antd';
import { createStyles, useTheme } from 'antd-style';
import type { DrawerClassNames, DrawerStyles } from
'antd/es/drawer/DrawerPanel';

const useStyle = createStyles(({ token }) => ({
  'my-drawer-body': {
    background: token.blue1,
  },
  'my-drawer-mask': {
    boxShadow: `inset 0 0 15px #fff`,
  },
  'my-drawer-header': {
    background: token.green1,
  },
  'my-drawer-footer': {
    color: token.colorPrimary,
  },
  'my-drawer-content': {
    borderLeft: '2px dotted #333',
  },
}));

const App: React.FC = () => {
  const [open, setOpen] = useState([false, false]);
  const { styles } = useStyle();
  const token = useTheme();

  const toggleDrawer = (idx: number, target: boolean) => {
    setOpen((p) => {
      p[idx] = target;
      return [...p];
    });
  };

  const classNames: DrawerClassNames = {
    body: styles['my-drawer-body'],
    mask: styles['my-drawer-mask'],
    header: styles['my-drawer-header'],
    footer: styles['my-drawer-footer'],
    content: styles['my-drawer-content'],
  };
};
```



```

const drawerStyles: DrawerStyles = {
  mask: {
    backdropFilter: 'blur(10px)',
  },
  content: {
    boxShadow: '-10px 0 10px #666',
  },
  header: {
    borderBottom: `1px solid ${token.colorPrimary}`,
  },
  body: {
    fontSize: token.fontSizeLG,
  },
  footer: {
    borderTop: `1px solid ${token.colorBorder}`,
  },
};

return (
  <>
    <Space>
      <Button type="primary" onClick={() => toggleDrawer(0, true)}>
        Open
      </Button>
      <Button type="primary" onClick={() => toggleDrawer(1, true)}>
        ConfigProvider
      </Button>
    </Space>
    <Drawer
      title="Basic Drawer"
      placement="right"
      footer="Footer"
      onClose={() => toggleDrawer(0, false)}
      open={open[0]}
      classNames={classNames}
      styles={drawerStyles}
    >
      <p>Some contents...</p>
      <p>Some contents...</p>
      <p>Some contents...</p>
    </Drawer>
    <ConfigProvider
      drawer={{
        classNames,
        styles: drawerStyles,
      }}
    >

```

```

    >
    <Drawer
      title="Basic Drawer"
      placement="right"
      footer="Footer"
      onClose={() => toggleDrawer(1, false)}
      open={open[1]}
    >
      <p>Some contents...</p>
      <p>Some contents...</p>
      <p>Some contents...</p>
    </Drawer>
  </ConfigProvider>
</>
);
};

export default App;

```

ConfigProvider

Debug

```

import React, { useRef, useState } from 'react';
import { Button, ConfigProvider, Drawer } from 'antd';

const App: React.FC = () => {
  const domRef = useRef<HTMLDivElement>(null);
  const [open, setOpen] = useState(false);

  const showDrawer = () => {
    setOpen(true);
  };

  const onClose = () => {
    setOpen(false);
  };

  return (
    <ConfigProvider getPopupContainer={() => domRef.current!}>
      <div ref={domRef} className="site-drawer-render-in-current-wrapper">
        <Button type="primary" onClick={showDrawer}>
          Open
        </Button>
        <Drawer
          rootStyle={{ position: 'absolute' }}

```

```

        title="ConfigProvider"
        placement="right"
        onClose={onClose}
        open={open}
      >
        <p>Some contents...</p>
        <p>Some contents...</p>
        <p>Some contents...</p>
      </Drawer>
    </div>
  </ConfigProvider>
);
};

export default App;

```

无遮罩

Debug

```

import React, { useState } from 'react';
import { Button, Drawer } from 'antd';

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

  const showDrawer = () => {
    setOpen(true);
  };

  const onClose = () => {
    setOpen(false);
  };

  return (
    <>
      <Button type="primary" onClick={showDrawer}>
        Open
      </Button>
      <Drawer
        title="Drawer without mask"
        placement="right"
        mask={false}
        onClose={onClose}
        open={open}
        styles={{

```

```

        mask: {
          width: 333,
          background: 'red',
          borderRadius: 20,
          boxShadow: '-5px 0 5px green',
          overflow: 'hidden',
        },
      }},
    >
    <p>Some contents...</p>
    <p>Some contents...</p>
    <p>Some contents...</p>
  </Drawer>
</>
);
};

export default App;

```

_InternalPanelDoNotUseOrYouWillBeFired

Debug

```

import React from 'react';
import { Drawer } from 'antd';

/** Test usage. Do not use in your production. */
const { _InternalPanelDoNotUseOrYouWillBeFired: InternalDrawer } = Drawer;

export default () => (
  <div style={{ padding: 32, background: '#e6e6e6' }}>
    <InternalDrawer title="Hello Title" style={{ height: 300 }}
      footer="Footer!">
      Hello Content
    </InternalDrawer>
  </div>
);

```

滚动锁定调试

Debug

```

import React, { useState } from 'react';
import { Drawer, Modal, Space, Switch } from 'antd';

const App: React.FC = () => {

```

```

const [drawer, setDrawer] = useState(false);
const [drawer2, setDrawer2] = useState(false);
const [modal, setModal] = useState(false);
const [modal2, setModal2] = useState(false);

return (
  <div style={{ position: 'relative', zIndex: 999999 }}>
    <Space>
      <Switch
        checkedChildren="Drawer"
        uncheckedChildren="Drawer"
        checked={drawer}
        onChange={() => setDrawer(!drawer)}
      />
      <Switch
        checkedChildren="Drawer2"
        uncheckedChildren="Drawer2"
        checked={drawer2}
        onChange={() => setDrawer2(!drawer2)}
      />
      <Switch
        checkedChildren="Modal"
        uncheckedChildren="Modal"
        checked={modal}
        onChange={() => setModal(!modal)}
      />
      <Switch
        checkedChildren="Modal2"
        uncheckedChildren="Modal2"
        checked={modal2}
        onChange={() => setModal2(!modal2)}
      />
    </Space>
    <Drawer title="Drawer" open={drawer}>
      Some contents...
      <Drawer title="Drawer Sub" open={drawer}>
        Sub contents...
      </Drawer>
    </Drawer>
    <Drawer title="Drawer2" open={drawer2}>
      Some contents...
    </Drawer>
    <Modal title="Modal" open={modal}>
      Some contents...
    </Modal>
    <Modal title="Modal2" open={modal2}>

```

```
        Some contents...
      </Modal>
    </div>
  );
};

export default App;
```

组件 Token

Debug

```
import React from 'react';
import { ConfigProvider, Drawer } from 'antd';

/** Test usage. Do not use in your production. */
const { _InternalPanelDoNotUseOrYouWillBeFired: InternalDrawer } = Drawer;

export default () => (
  <ConfigProvider
    theme={{ components: { Drawer: { footerPaddingBlock: 0,
    footerPaddingInline: 0 } } }}
  >
    <div style={{ padding: 32, background: '#e6e6e6' }}>
      <InternalDrawer title="Hello Title" style={{ height: 300 }}
        footer="Footer!">
        Hello Content
      </InternalDrawer>
    </div>
  </ConfigProvider>
);
```

API

通用属性参考：[通用属性](#)

:::warning{title=注意} v5 使用 `rootClassName` 与 `rootStyle` 来配置最外层元素样式。原 v4 `className` 与 `style` 改至配置 Drawer 窗体样式以和 Modal 对齐。:::

参数	说明	类型	默认值	版本
autoFocus	抽屉展开后是否将焦点切换至其 DOM 节点	boolean	true	4.17.0
afterOpenChange	切换抽屉时动画结束后的回调	function(open)	-	

className	Drawer 容器外层 className 设置，如果需要设置最外层，请使用 rootClassName	string	-	
classNames	语义化结构 className	Record<SemanticDOM, string>	-	5.10.0
closelcon	自定义关闭图标。5.7.0: 设置为 null 或 false 时隐藏关闭按钮	ReactNode	<CloseOutlined />	
destroyOnClose	关闭时销毁 Drawer 里的子元素	boolean	false	
extra	抽屉右上角的操作区域	ReactNode	-	4.17.0
footer	抽屉的页脚	ReactNode	-	
forceRender	预渲染 Drawer 内元素	boolean	false	
getContainer	指定 Drawer 挂载的节点，并在容器内展现，false 为挂载在当前位置	HTMLElement () => HTMLElement Selectors false	body	
height	高度，在 placement 为 top 或 bottom 时使用	string number	378	
keyboard	是否支持键盘 esc 关闭	boolean	true	
mask	是否展示遮罩	boolean	true	
maskClosable	点击蒙层是否允许关闭	boolean	true	
placement	抽屉的方向	top right bottom left	right	

push	用于设置多层 Drawer 的推动行为	boolean { distance: string number }	{ distance: 180 }	4.5.0+
rootStyle	可用于设置 Drawer 最外层容器的样式，和 style 的区别是作用节点包括 mask	CSSProperties	-	
size	预设抽屉宽度（或高度），default 378px 和 large 736px	'default' 'large'	'default'	4.17.0
style	设计 Drawer 容器样式，如果你只需要设置内容部分请使用 styles.body	CSSProperties	-	
styles	语义化结构 style	Record<SemanticDOM, CSSProperties>	-	5.10.0
title	标题	ReactNode	-	
loading	显示骨架屏	boolean	false	5.17.0
open	Drawer 是否可见	boolean	-	
width	宽度	string number	378	
zIndex	设置 Drawer 的 z-index	number	1000	
onClose	点击遮罩层或左上角叉或取消按钮的回调	function(e)	-	
drawerRender	自定义渲染抽屉	(node: ReactNode) => ReactNode	-	5.18.0

Semantic DOM

演示

```
import React from 'react';
import { Drawer, Typography } from 'antd';
```



```
import SemanticPreview from '../../../.dumi/components/SemanticPreview';
import useLocale from '../../../.dumi/hooks/useLocale';

const locales = {
  cn: {
    mask: '遮罩层元素',
    content: 'Drawer 容器元素',
    header: '头部元素',
    body: '内容元素',
    footer: '底部元素',
  },
  en: {
    mask: 'Mask element',
    content: 'Drawer container element',
    header: 'Header element',
    body: 'Body element',
    footer: 'Footer element',
  },
};

const App: React.FC = () => {
  const [locale] = useLocale(locales);
  return (
    <SemanticPreview
      componentName="Drawer"
      semantics={[
        { name: 'mask', desc: locale.mask, version: '5.13.0' },
        { name: 'content', desc: locale.content, version: '5.13.0' },
        { name: 'header', desc: locale.header, version: '5.13.0' },
        { name: 'body', desc: locale.body, version: '5.13.0' },
        { name: 'footer', desc: locale.footer, version: '5.13.0' },
      ]}
      height={300}
    >
      <Drawer
        title="Title"
        placement="right"
        footer={<Typography.Link>Footer</Typography.Link>}
        closable={false}
        open
        getContainer={false}
      >
        <p>Some contents...</p>
      </Drawer>
    </SemanticPreview>
  )
}
```

```
);  
};  
  
export default App;
```

主题变量 (Design Token)