

何时使用 {#when-to-use}

导航菜单是一个网站的灵魂，用户依赖导航在各个页面中进行跳转。一般分为顶部导航和侧边导航，顶部导航提供全局性的类目和功能，侧边导航提供多级结构来收纳和排列网站架构。

更多布局和导航的使用可以参考：[通用布局](#)。

开发者注意事项

- Menu 元素为 `ul`，因而仅支持 [li](#) 以及 [script-supporting 子元素](#)。因而你的子节点元素应该都在 `Menu.Item` 内使用。
- Menu 需要计算节点结构，因而其子元素仅支持 `Menu.*` 以及对此进行封装的 HOC 组件。

代码演示

顶部导航

```
import React, { useState } from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-
design/icons';
import type { MenuProps } from 'antd';
import { Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    label: 'Navigation One',
    key: 'mail',
    icon: <MailOutlined />,
  },
  {
    label: 'Navigation Two',
    key: 'app',
    icon: <AppstoreOutlined />,
    disabled: true,
  },
  {
    label: 'Navigation Three - Submenu',
    key: 'SubMenu',
    icon: <SettingOutlined />,
    children: [
      {
        type: 'group',
        label: 'Item 1',
        children: [
          { label: 'Option 1', key: 'setting:1' },
```

```

        { label: 'Option 2', key: 'setting:2' },
      ],
    },
    {
      type: 'group',
      label: 'Item 2',
      children: [
        { label: 'Option 3', key: 'setting:3' },
        { label: 'Option 4', key: 'setting:4' },
      ],
    },
  ],
},
{
  key: 'alipay',
  label: (
    <a href="https://ant.design" target="_blank" rel="noopener
noreferrer">
      Navigation Four - Link
    </a>
  ),
},
];

const App: React.FC = () => {
  const [current, setCurrent] = useState('mail');

  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
    setCurrent(e.key);
  };

  return <Menu onClick={onClick} selectedKeys={[current]} mode="horizontal"
items={items} />;
};

export default App;

```

顶部导航 (dark)

Debug

```

import React, { useState } from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-
design/icons';
import type { MenuProps } from 'antd';

```

```

import { Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    label: 'Navigation One',
    key: 'mail',
    icon: <MailOutlined />,
  },
  {
    label: 'Navigation Two',
    key: 'app',
    icon: <AppstoreOutlined />,
    disabled: true,
  },
  {
    label: 'Navigation Three - Submenu',
    key: 'SubMenu',
    icon: <SettingOutlined />,
    children: [
      {
        type: 'group',
        label: 'Item 1',
        children: [
          { label: 'Option 1', key: 'setting:1' },
          { label: 'Option 2', key: 'setting:2' },
        ],
      },
      {
        type: 'group',
        label: 'Item 2',
        children: [
          { label: 'Option 3', key: 'setting:3' },
          { label: 'Option 4', key: 'setting:4' },
        ],
      },
    ],
  },
  {
    key: 'alipay',
    label: (
      <a href="https://ant.design" target="_blank" rel="noopener
norereferrer">
        Navigation Four - Link
      </a>
    )
  }
];

```

```

    ),
  },
];

const App: React.FC = () => {
  const [current, setCurrent] = useState('mail');

  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
    setCurrent(e.key);
  };

  return (
    <Menu onClick={onClick} selectedKeys={[current]} mode="horizontal"
    items={items} theme="dark" />
  );
};

export default App;

```

内嵌菜单

```

import React from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-
design/icons';
import type { MenuProps } from 'antd';
import { Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: 'sub1',
    label: 'Navigation One',
    icon: <MailOutlined />,
    children: [
      {
        key: 'g1',
        label: 'Item 1',
        type: 'group',
        children: [
          { key: '1', label: 'Option 1' },
          { key: '2', label: 'Option 2' },
        ],
      },
    ],
  },
  {

```

```

        key: 'g2',
        label: 'Item 2',
        type: 'group',
        children: [
            { key: '3', label: 'Option 3' },
            { key: '4', label: 'Option 4' },
        ],
    },
],
},
{
    key: 'sub2',
    label: 'Navigation Two',
    icon: <AppstoreOutlined />,
    children: [
        { key: '5', label: 'Option 5' },
        { key: '6', label: 'Option 6' },
        {
            key: 'sub3',
            label: 'Submenu',
            children: [
                { key: '7', label: 'Option 7' },
                { key: '8', label: 'Option 8' },
            ],
        },
    ],
},
],
},
{
    type: 'divider',
},
{
    key: 'sub4',
    label: 'Navigation Three',
    icon: <SettingOutlined />,
    children: [
        { key: '9', label: 'Option 9' },
        { key: '10', label: 'Option 10' },
        { key: '11', label: 'Option 11' },
        { key: '12', label: 'Option 12' },
    ],
},
{
    key: 'grp',
    label: 'Group',
    type: 'group',
    children: [

```

```

    { key: '13', label: 'Option 13' },
    { key: '14', label: 'Option 14' },
  ],
},
];

const App: React.FC = () => {
  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
  };

  return (
    <Menu
      onClick={onClick}
      style={{ width: 256 }}
      defaultSelectedKeys={['1']}
      defaultOpenKeys={['sub1']}
      mode="inline"
      items={items}
    />
  );
};

export default App;

```

缩起内嵌菜单

```

import React, { useState } from 'react';
import {
  AppstoreOutlined,
  ContainerOutlined,
  DesktopOutlined,
  MailOutlined,
  MenuFoldOutlined,
  MenuUnfoldOutlined,
  PieChartOutlined,
} from '@ant-design/icons';
import type { MenuProps } from 'antd';
import { Button, Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  { key: '1', icon: <PieChartOutlined />, label: 'Option 1' },
  { key: '2', icon: <DesktopOutlined />, label: 'Option 2' },
  { key: '3', icon: <ContainerOutlined />, label: 'Option 3' },

```

```

{
  key: 'sub1',
  label: 'Navigation One',
  icon: <MailOutlined />,
  children: [
    { key: '5', label: 'Option 5' },
    { key: '6', label: 'Option 6' },
    { key: '7', label: 'Option 7' },
    { key: '8', label: 'Option 8' },
  ],
},
{
  key: 'sub2',
  label: 'Navigation Two',
  icon: <AppstoreOutlined />,
  children: [
    { key: '9', label: 'Option 9' },
    { key: '10', label: 'Option 10' },
    {
      key: 'sub3',
      label: 'Submenu',
      children: [
        { key: '11', label: 'Option 11' },
        { key: '12', label: 'Option 12' },
      ],
    },
  ],
},
],
},
];

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

  const toggleCollapsed = () => {
    setCollapsed(!collapsed);
  };

  return (
    <div style={{ width: 256 }}>
      <Button type="primary" onClick={toggleCollapsed} style={{
marginBottom: 16 }}>
        {collapsed ? <MenuUnfoldOutlined /> : <MenuFoldOutlined />}
      </Button>
      <Menu
        defaultSelectedKeys={['1']}
        defaultOpenKeys={['sub1']}

```

```

        mode="inline"
        theme="dark"
        inlineCollapsed={collapsed}
        items={items}
      />
    </div>
  );
};

export default App;

```

只展开当前父级菜单

```

import React, { useState } from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-
design/icons';
import type { MenuProps } from 'antd';
import { Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: '1',
    icon: <MailOutlined />,
    label: 'Navigation One',
    children: [
      { key: '11', label: 'Option 1' },
      { key: '12', label: 'Option 2' },
      { key: '13', label: 'Option 3' },
      { key: '14', label: 'Option 4' },
    ],
  },
  {
    key: '2',
    icon: <AppstoreOutlined />,
    label: 'Navigation Two',
    children: [
      { key: '21', label: 'Option 1' },
      { key: '22', label: 'Option 2' },
      {
        key: '23',
        label: 'Submenu',
        children: [
          { key: '231', label: 'Option 1' },
          { key: '232', label: 'Option 2' },

```



```

        { key: '233', label: 'Option 3' },
    ],
},
{
    key: '24',
    label: 'Submenu 2',
    children: [
        { key: '241', label: 'Option 1' },
        { key: '242', label: 'Option 2' },
        { key: '243', label: 'Option 3' },
    ],
},
],
},
{
    key: '3',
    icon: <SettingOutlined />,
    label: 'Navigation Three',
    children: [
        { key: '31', label: 'Option 1' },
        { key: '32', label: 'Option 2' },
        { key: '33', label: 'Option 3' },
        { key: '34', label: 'Option 4' },
    ],
},
];

interface LevelKeysProps {
    key?: string;
    children?: LevelKeysProps[];
}

const getLevelKeys = (items1: LevelKeysProps[]) => {
    const key: Record<string, number> = {};
    const func = (items2: LevelKeysProps[], level = 1) => {
        items2.forEach((item) => {
            if (item.key) {
                key[item.key] = level;
            }
            if (item.children) {
                func(item.children, level + 1);
            }
        });
    };
    func(items1);
    return key;
};

```

```

};

const levelKeys = getLevelKeys(items as LevelKeysProps[]);

const App: React.FC = () => {
  const [stateOpenKeys, setStateOpenKeys] = useState(['2', '23']);

  const onOpenChange: MenuProps['onOpenChange'] = (openKeys) => {
    const currentOpenKey = openKeys.find((key) =>
stateOpenKeys.indexOf(key) === -1);
    // open
    if (currentOpenKey !== undefined) {
      const repeatIndex = openKeys
        .filter((key) => key !== currentOpenKey)
        .findIndex((key) => levelKeys[key] === levelKeys[currentOpenKey]);

      setStateOpenKeys(
        openKeys
          // remove repeat key
          .filter((_, index) => index !== repeatIndex)
          // remove current level all child
          .filter((key) => levelKeys[key] <= levelKeys[currentOpenKey]),
      );
    } else {
      // close
      setStateOpenKeys(openKeys);
    }
  };

  return (
    <Menu
      mode="inline"
      defaultSelectedKeys={['231']}
      openKeys={stateOpenKeys}
      onOpenChange={onOpenChange}
      style={{ width: 256 }}
      items={items}
    />
  );
};

export default App;

```

垂直菜单

```
import React from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-
design/icons';
import type { MenuProps } from 'antd';
import { Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: 'sub1',
    icon: <MailOutlined />,
    label: 'Navigation One',
    children: [
      {
        key: '1-1',
        label: 'Item 1',
        type: 'group',
        children: [
          { key: '1', label: 'Option 1' },
          { key: '2', label: 'Option 2' },
        ],
      },
      {
        key: '1-2',
        label: 'Item 2',
        type: 'group',
        children: [
          { key: '3', label: 'Option 3' },
          { key: '4', label: 'Option 4' },
        ],
      },
    ],
  },
  {
    key: 'sub2',
    icon: <AppstoreOutlined />,
    label: 'Navigation Two',
    children: [
      { key: '5', label: 'Option 5' },
      { key: '6', label: 'Option 6' },
      {
        key: 'sub3',
        label: 'Submenu',
        children: [
          { key: '7', label: 'Option 7' },
        ],
      },
    ],
  },
]
```

```

        { key: '8', label: 'Option 8' },
      ],
    },
  ],
},
{
  key: 'sub4',
  label: 'Navigation Three',
  icon: <SettingOutlined />,
  children: [
    { key: '9', label: 'Option 9' },
    { key: '10', label: 'Option 10' },
    { key: '11', label: 'Option 11' },
    { key: '12', label: 'Option 12' },
  ],
},
];

const onClick: MenuProps['onClick'] = (e) => {
  console.log('click', e);
};

const App: React.FC = () => (
  <Menu onClick={onClick} style={{ width: 256 }} mode="vertical" items=
{items} />
);

export default App;

```

主题

```

import React, { useState } from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-
design/icons';
import type { MenuProps, MenuTheme } from 'antd';
import { Menu, Switch } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: 'sub1',
    label: 'Navigation One',
    icon: <MailOutlined />,
    children: [
      { key: '1', label: 'Option 1' },

```

```

    { key: '2', label: 'Option 2' },
    { key: '3', label: 'Option 3' },
    { key: '4', label: 'Option 4' },
  ],
},
{
  key: 'sub2',
  label: 'Navigation Two',
  icon: <AppstoreOutlined />,
  children: [
    { key: '5', label: 'Option 5' },
    { key: '6', label: 'Option 6' },
    {
      key: 'sub3',
      label: 'Submenu',
      children: [
        { key: '7', label: 'Option 7' },
        { key: '8', label: 'Option 8' },
      ],
    },
  ],
},
],
},
{
  key: 'sub4',
  label: 'Navigation Three',
  icon: <SettingOutlined />,
  children: [
    { key: '9', label: 'Option 9' },
    { key: '10', label: 'Option 10' },
    { key: '11', label: 'Option 11' },
    { key: '12', label: 'Option 12' },
  ],
},
],
};

const App: React.FC = () => {
  const [theme, setTheme] = useState<MenuTheme>('dark');
  const [current, setCurrent] = useState('1');

  const changeTheme = (value: boolean) => {
    setTheme(value ? 'dark' : 'light');
  };

  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
    setCurrent(e.key);
  };

```

```

};

return (
  <>
    <Switch
      checked={theme === 'dark'}
      onChange={changeTheme}
      checkedChildren="Dark"
      uncheckedChildren="Light"
    />
    <br />
    <br />
    <Menu
      theme={theme}
      onClick={onClick}
      style={{ width: 256 }}
      defaultOpenKeys={['sub1']}
      selectedKeys={[current]}
      mode="inline"
      items={items}
    />
  </>
);
};

export default App;

```

子菜单主题

```

import React, { useState } from 'react';
import { MailOutlined } from '@ant-design/icons';
import type { MenuProps, MenuTheme } from 'antd';
import { Menu, Switch } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const App: React.FC = () => {
  const [menuTheme, setMenuTheme] = useState<MenuTheme>('light');
  const [current, setCurrent] = useState('1');

  const changeTheme = (value: boolean) => {
    setMenuTheme(value ? 'dark' : 'light');
  };

  const onClick: MenuProps['onClick'] = (e) => {
    setCurrent(e.key);
  };

```

```

};

const items: MenuItem[] = [
  {
    key: 'sub1',
    icon: <MailOutlined />,
    label: 'Navigation One',
    theme: menuTheme,
    children: [
      { key: '1', label: 'Option 1' },
      { key: '2', label: 'Option 2' },
      { key: '3', label: 'Option 3' },
    ],
  },
  { key: '5', label: 'Option 5' },
  { key: '6', label: 'Option 6' },
];

return (
  <>
    <Switch
      checked={menuTheme === 'dark'}
      onChange={changeTheme}
      checkedChildren="Dark"
      uncheckedChildren="Light"
    />
    <br />
    <br />
    <Menu
      onClick={onClick}
      style={{ width: 256 }}
      openKeys={['sub1']}
      selectedKeys={[current]}
      mode="vertical"
      theme="dark"
      items={items}
      getPopupContainer={(node) => node.parentNode as HTMLElement}
    />
  </>
);
};

export default App;

```

切换菜单类型

```
import React, { useState } from 'react';
import {
  AppstoreOutlined,
  CalendarOutlined,
  LinkOutlined,
  MailOutlined,
  SettingOutlined,
} from '@ant-design/icons';
import { Divider, Menu, Switch } from 'antd';
import type { GetProp, MenuProps } from 'antd';

type MenuTheme = GetProp<MenuProps, 'theme'>;

type MenuItem = GetProp<MenuProps, 'items'>[number];

const items: MenuItem[] = [
  {
    key: '1',
    icon: <MailOutlined />,
    label: 'Navigation One',
  },
  {
    key: '2',
    icon: <CalendarOutlined />,
    label: 'Navigation Two',
  },
  {
    key: 'sub1',
    label: 'Navigation Two',
    icon: <AppstoreOutlined />,
    children: [
      { key: '3', label: 'Option 3' },
      { key: '4', label: 'Option 4' },
      {
        key: 'sub1-2',
        label: 'Submenu',
        children: [
          { key: '5', label: 'Option 5' },
          { key: '6', label: 'Option 6' },
        ],
      },
    ],
  },
  {
    key: 'sub2',
    label: 'Navigation Three',
  },
]
```



```

        icon: <SettingOutlined />,
        children: [
          { key: '7', label: 'Option 7' },
          { key: '8', label: 'Option 8' },
          { key: '9', label: 'Option 9' },
          { key: '10', label: 'Option 10' },
        ],
      },
      {
        key: 'link',
        icon: <LinkOutlined />,
        label: (
          <a href="https://ant.design" target="_blank" rel="noopener
noreferrer">
            Ant Design
          </a>
        ),
      },
    ],
  ];

const App: React.FC = () => {
  const [mode, setMode] = useState<'vertical' | 'inline'>('inline');
  const [theme, setTheme] = useState<MenuTheme>('light');

  const changeMode = (value: boolean) => {
    setMode(value ? 'vertical' : 'inline');
  };

  const changeTheme = (value: boolean) => {
    setTheme(value ? 'dark' : 'light');
  };

  return (
    <>
      <Switch onChange={changeMode} /> Change Mode
      <Divider type="vertical" />
      <Switch onChange={changeTheme} /> Change Style
      <br />
      <br />
      <Menu
        style={{ width: 256 }}
        defaultSelectedKeys={['1']}
        defaultOpenKeys={['sub1']}
        mode={mode}
        theme={theme}
        items={items}
      />
    </>
  );
};

```

```

        />
      </>
    );
  };

  export default App;

```

Style debug

Debug

```

import React, { useState } from 'react';
import { AppstoreOutlined, MailOutlined } from '@ant-design/icons';
import type { MenuProps, MenuTheme } from 'antd';
import { Menu, Switch } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: 'sub1',
    label: 'Navigation One Long Long Long Long',
    icon: <MailOutlined />,
    children: [
      { key: '1', label: 'Option 1' },
      { key: '2', label: 'Option 2' },
      { key: '3', label: 'Option 3' },
      { key: '4', label: 'Option 4' },
    ],
  },
  {
    key: 'sub2',
    label: 'Navigation Two',
    icon: <AppstoreOutlined />,
    children: [
      { key: '5', label: 'Option 5' },
      { key: '6', label: 'Option 6' },
      {
        key: 'sub3',
        label: 'Submenu',
        children: [
          { key: '7', label: 'Option 7' },
          { key: '8', label: 'Option 8' },
        ],
      },
    ],
  },
],

```

```

    },
    { key: '11', label: 'Option 11' },
    { key: '12', label: 'Option 12' },
  ];

const App: React.FC = () => {
  const [menuTheme, setMenuTheme] = useState<MenuTheme>('dark');
  const [current, setCurrent] = useState('1');

  const changeTheme = (value: boolean) => {
    setMenuTheme(value ? 'dark' : 'light');
  };

  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
    setCurrent(e.key);
  };

  return (
    <>
      <Switch
        checked={menuTheme === 'dark'}
        onChange={changeTheme}
        checkedChildren="Dark"
        uncheckedChildren="Light"
      />
      <br />
      <br />
      <Menu
        theme={menuTheme}
        onClick={onClick}
        selectedKeys={[current]}
        mode="inline"
        items={items}
        inlineCollapsed
        // Test only. Remove in future.
        _internalRenderMenuItem={(node) =>
          React.cloneElement<any>(node, {
            style: {
              ...(node as any).props.style,
              textDecoration: 'underline',
            },
          })
        }
        // Test only. Remove in future.
        _internalRenderSubMenuItem={(node) =>

```

```

      React.cloneElement<any>(node, {
        style: {
          ...(node as any).props.style,
          background: 'rgba(255, 255, 255, 0.3)',
        },
      })
    }
    // Test only. Remove in future.
    _internalDisableMenuItemTitleTooltip
  />
</>
);
};

export default App;

```

v4 版本 Menu

Debug

```

import React, { useState } from 'react';
import {
  AppstoreOutlined,
  CalendarOutlined,
  LinkOutlined,
  MailOutlined,
  SettingOutlined,
} from '@ant-design/icons';
import { ConfigProvider, Menu, Switch, Typography } from 'antd';
import type { MenuProps } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: '1',
    icon: <MailOutlined />,
    label: 'Navigation One',
  },
  {
    key: '2',
    icon: <CalendarOutlined />,
    label: 'Navigation Two',
  },
  {
    key: 'sub1',

```

```

icon: <AppstoreOutlined />,
label: 'Navigation Two',
children: [
  {
    key: '3',
    label: (
      <Typography.Text ellipsis>
        Ant Design, a design language for background applications, is
refined by Ant UED Team
      </Typography.Text>
    ),
  },
  {
    key: '4',
    label: 'Option 4',
  },
  {
    key: 'sub1-2',
    label: 'Submenu',
    children: [
      { key: '5', label: 'Option 5' },
      { key: '6', label: 'Option 6' },
    ],
  },
],
},
{
  key: 'sub2',
  label: 'Navigation Three',
  icon: <SettingOutlined />,
  children: [
    { label: 'Option 7', key: '7' },
    { label: 'Option 8', key: '8' },
    { label: 'Option 9', key: '9' },
    { label: 'Option 10', key: '10' },
  ],
},
{
  key: 'link',
  icon: <LinkOutlined />,
  label: (
    <a href="https://ant.design" target="_blank" rel="noopener
norereferrer">
      Ant Design
    </a>
  ),

```

```

    },
  ];

const App: React.FC = () => {
  const [mode, setMode] = useState<'vertical' | 'inline'>('inline');

  const changeMode = (value: boolean) => {
    setMode(value ? 'vertical' : 'inline');
  };

  return (
    <>
      <Switch onChange={changeMode} /> Change Mode
      <br />
      <br />
      <ConfigProvider
        theme={{
          components: {
            Menu: {
              itemBorderRadius: 0,
              subMenuItemBorderRadius: 0,
              itemHoverColor: '#1890ff',
              itemSelectedColor: '#1890ff',
              itemSelectedBg: '#e6f7ff',
              activeBarWidth: 3,
              itemMarginInline: 0,
              itemHoverBg: 'transparent',
            },
          },
        }}
      >
        <Menu
          style={{ width: 256 }}
          defaultSelectedKeys={['1']}
          defaultOpenKeys={['sub1']}
          mode={mode}
          items={items}
        />
      </ConfigProvider>
    </>
  );
};

export default App;

```

组件 Token

Debug

```
import React, { useState } from 'react';
import {
  AppstoreOutlined,
  ContainerOutlined,
  DesktopOutlined,
  MailOutlined,
  PieChartOutlined,
  SettingOutlined,
} from '@ant-design/icons';
import type { MenuProps } from 'antd';
import { ConfigProvider, Menu, Space, theme } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    label: 'Navigation One',
    key: 'mail',
    icon: <MailOutlined />,
  },
  {
    label: 'Navigation Two',
    key: 'app',
    icon: <AppstoreOutlined />,
    disabled: true,
  },
  {
    label: 'Navigation Three - Submenu',
    key: 'SubMenu',
    icon: <SettingOutlined />,
    children: [
      {
        type: 'group',
        label: 'Item 1',
        children: [
          { label: 'Option 1', key: 'setting:1' },
          { label: 'Option 2', key: 'setting:2' },
        ],
      },
      {
        type: 'group',
        label: 'Item 2',
        children: [
          { label: 'Option 3', key: 'setting:3' },
        ],
      },
    ],
  },
];
```

```

        { label: 'Option 4', key: 'setting:4' },
      ],
    },
  ],
},
{
  key: 'alipay',
  label: (
    <a href="https://ant.design" target="_blank" rel="noopener
norereferrer">
      Navigation Four - Link
    </a>
  ),
},
];

const items2: MenuItem[] = [
  {
    key: '1',
    icon: <PieChartOutlined />,
    label: 'Option 1',
  },
  {
    key: '2',
    icon: <DesktopOutlined />,
    label: 'Option 2',
  },
  {
    key: '3',
    icon: <ContainerOutlined />,
    label: 'Option 3',
  },
  {
    key: 'sub1',
    label: 'Navigation One',
    icon: <MailOutlined />,
    children: [
      { key: '5', label: 'Option 5' },
      { key: '6', label: 'Option 6' },
      { key: '7', label: 'Option 7' },
      { key: '8', label: 'Option 8' },
    ],
  },
  {
    key: 'sub2',
    label: 'Navigation Two',
  },
];

```



```

    icon: <AppstoreOutlined />,
    children: [
      { key: '9', label: 'Option 9' },
      { key: '10', label: 'Option 10' },
      {
        key: 'sub3',
        label: 'Submenu',
        children: [
          { key: '11', label: 'Option 11' },
          { key: '12', label: 'Option 12' },
        ],
      },
    ],
  },
],
};

const App: React.FC = () => {
  const [current, setCurrent] = useState('mail');

  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
    setCurrent(e.key);
  };

  return (
    <Space direction="vertical">
      <ConfigProvider
        theme={{
          algorithm: [theme.darkAlgorithm],
          components: {
            Menu: {
              popupBg: 'yellow',
              darkPopupBg: 'red',
            },
          },
        }}
      >
        <Menu onClick={onClick} selectedKeys={[current]} mode="horizontal"
items={items} />
        <Menu
          defaultSelectedKeys={['1']}
          defaultOpenKeys={['sub1']}
          mode="inline"
          theme="dark"
          inlineCollapsed
          items={items2}

```

```

        style={{
            width: 56,
        }}
    />
</ConfigProvider>
<ConfigProvider
    theme={{
        components: {
            Menu: {
                horizontalItemBorderRadius: 6,
                popupBg: 'red',
                horizontalItemHoverBg: '#f5f5f5',
            },
        },
    }}
>
    <Menu onClick={onClick} selectedKeys={[current]} mode="horizontal"
items={items} />
</ConfigProvider>
<ConfigProvider
    theme={{
        components: {
            Menu: {
                darkItemColor: '#91daff',
                darkItemBg: '#d48806',
                darkSubMenuItemBg: '#faad14',
                darkItemSelectedColor: '#ffccc7',
                darkItemSelectedBg: '#52c41a',
            },
        },
    }}
>
    <Menu
        defaultSelectedKeys={['1']}
        defaultOpenKeys={['sub1']}
        mode="inline"
        theme="dark"
        items={items2}
        style={{
            width: 256,
        }}
    />
</ConfigProvider>
</Space>
);
};

```

```
export default App;
```

Extra Style debug

Debug

```
import React from 'react';
import { DownOutlined, MailOutlined } from '@ant-design/icons';
import type { MenuProps } from 'antd';
import { Flex, Menu, Space } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items1: MenuItem[] = [
  {
    key: 'sub1',
    icon: <MailOutlined />,
    label: 'Navigation One',
    children: [
      {
        key: '1',
        label: (
          <Flex justify="space-between">
            <span>Option 1</span>
            <DownOutlined />
          </Flex>
        ),
      },
      {
        key: '2',
        label: 'Option 2',
        extra: '%P',
      },
      {
        key: '3',
        label: <a href="https://www.baidu.com">Link Option</a>,
        disabled: true,
      },
    ],
  },
];

const items2: MenuItem[] = [
  { key: '1', label: 'Users', extra: '%U' },
  { key: '2', label: 'Profile', extra: '%P' },
];
```

```
];

const App: React.FC = () => (
  <Space direction="vertical">
    <Menu
      mode="inline"
      defaultOpenKeys={['sub1']}
      defaultSelectedKeys={['1']}
      style={{ width: 256 }}
      items={items1}
    />
    <Menu theme="dark" style={{ width: 256 }} items={items2} />
  </Space>
);

export default App;
```

API

通用属性参考: [通用属性](#)

Menu

参数	说明	类型	默认值	版本
defaultOpenKeys	初始展开的 SubMenu 菜单项 key 数组	string[]	-	
defaultSelectedKeys	初始选中的菜单项 key 数组	string[]	-	
expandIcon	自定义展开图标	ReactNode (props: SubMenuProps & { isSubMenu: boolean }) => ReactNode	-	4.9.0
forceSubMenuRender	在子菜单展示之前就渲染进 DOM	boolean	false	

inlineCollapsed	inline 时菜单是否收起状态	boolean	-	
inlineIndent	inline 模式的菜单缩进宽度	number	24	
items	菜单内容	ItemType[]	-	4.20.0
mode	菜单类型，现在支持垂直、水平、和内嵌模式三种	vertical horizontal inline	vertical	
multiple	是否允许多选	boolean	false	
openKeys	当前展开的 SubMenu 菜单项 key 数组	string[]	-	
overflowedIndicator	用于自定义 Menu 水平空间不足时的省略收缩的图标	ReactNode	<EllipsisOutlined />	
selectable	是否允许选中	boolean	true	
selectedKeys	当前选中的菜单项 key 数组	string[]	-	
style	根节点样式	CSSProperties	-	
submenuCloseDelay	用户鼠标离开子菜单后关闭延时，单位：秒	number	0.1	

subMenuOpenDelay	用户鼠标进入子菜单后开启延时，单位：秒	number	0	
theme	主题颜色	light dark	light	
triggerSubMenuAction	SubMenu 展开/关闭的触发行为	hover click	hover	
onClick	点击 MenuItem 调用此函数	function({ item, key, keyPath, domEvent })	-	
onDeselect	取消选中时调用，仅在 multiple 生效	function({ item, key, keyPath, selectedKeys, domEvent })	-	
onOpenChange	SubMenu 展开/关闭的回调	function(openKeys: string[])	-	
onSelect	被选中时调用	function({ item, key, keyPath, selectedKeys, domEvent })	-	

更多属性查看 [rc-menu](#)

ItemType

type ItemType = [MenuItemType](#) | [SubMenuType](#) | [MenuItemGroupType](#) | [MenuDividerType](#);

MenuItemType

参数	说明	类型	默认值	版本
danger	展示错误状态样式	boolean	false	
disabled	是否禁用	boolean	false	
extra	额外节点	ReactNode	-	5.21.0
icon	菜单图标	ReactNode	-	

key	item 的唯一标志	string	-	
label	菜单项标题	ReactNode	-	
title	设置收缩时展示的悬浮标题	string	-	

SubMenuType

参数	说明	类型	默认值	版本
children	子菜单的菜单项	ItemType[]	-	
disabled	是否禁用	boolean	false	
icon	菜单图标	ReactNode	-	
key	唯一标志	string	-	
label	菜单项标题	ReactNode	-	
popupClassName	子菜单样式, mode="inline" 时无效	string	-	
popupOffset	子菜单偏移量, mode="inline" 时无效	[number, number]	-	
onTitleClick	点击子菜单标题	function({ key, domEvent })	-	
theme	设置子菜单的主题, 默认从 Menu 上继承	light dark	-	

MenuItemGroupType

定义类型为 `group` 时, 会作为分组处理:

```
const groupItem = {
  type: 'group', // Must have
  label: 'My Group',
  children: [],
};
```

参数	说明	类型	默认值	版本
children	分组的菜单项	MenuItemType[]	-	
label	分组标题	ReactNode	-	

MenuDividerType

菜单项分割线，只用在弹出菜单内，需要定义类型为 `divider`：

```
const dividerItem = {
  type: 'divider', // Must have
};
```

参数	说明	类型	默认值	版本
dashed	是否虚线	boolean	false	

FAQ

为何 Menu 的子元素会渲染两次？

Menu 通过 [二次渲染](#) 收集嵌套结构信息以支持 HOC 的结构。合并成一个推导结构会使得逻辑变得十分复杂，欢迎 PR 以协助改进该设计。

在 Flex 布局中，Menu 没有按照预期响应式省略菜单？

Menu 初始化时会先全部渲染，然后根据宽度裁剪内容。当处于 Flex 布局中，你需要告知其预期宽度为响应式宽度 ([在线 Demo](#))：

```
<div style={{ flex }}>
  <div style={{ ... }}>Some Content</div>
  <Menu style={{ minWidth: 0, flex: "auto" }} />
</div>
```

主题变量（Design Token）