

## zh-CN

通过 css variable 修改全局主题色（你可以切换到组件页面查看更详细的样式展示），不支持 IE。自动生成的变量可能会根据设计调整，请勿直接依赖。详细配置请[点击查看](#)。

## en-US

Modify global theme color by css variable which IE not support. Css variable depends on the design, it may adjust so please do not directly use it. You can go to other components page for more detail style. [Check this](#) to view detail.

```
import { SketchPicker } from 'react-color';
import React, { useState } from 'react';
import {
  DownOutlined,
  MailOutlined,
  SettingOutlined,
  ClockCircleOutlined,
} from '@ant-design/icons';
import {
  ConfigProvider,
  Tag,
  Mentions,
  Steps,
  Button,
  Radio,
  Space,
  Form,
  Input,
  Row,
  Col,
  Typography,
  Menu,
  Dropdown,
  Divider,
  Pagination,
  Select,
  Checkbox,
  DatePicker,
  TimePicker,
  InputNumber,
  Slider,
  Switch,
  TreeSelect,
  Card,
  Table,
  Tabs,
  Timeline,
  Tree,
  Alert,
  Progress,
  Spin,
```

```
    Transfer,
  } from 'antd';

const SplitSpace = props => <Space split={<Divider type="vertical" />} size={4}
{...props} />;

const inputProps = {
  style: { width: 128 },
};

const selectProps = {
  ...inputProps,
  options: [
    { value: 'light', label: 'Light' },
    { value: 'bamboo', label: 'Bamboo' },
    { value: 'little', label: 'Little' },
  ],
};

const treeData = [
  {
    value: 'little',
    key: 'little',
    label: 'Little',
    title: 'Little',
    children: [
      { value: 'light', key: 'light', label: 'Light', title: 'Light' },
      { value: 'bamboo', key: 'bamboo', label: 'Bamboo', title: 'Bamboo' },
    ],
  },
];

const treeSelectProps = {
  ...inputProps,
  treeCheckable: true,
  maxTagCount: 'responsive',
  treeData,
};

const carTabListNoTitle = [
  {
    key: 'article',
    tab: 'article',
  },
  {
    key: 'app',
    tab: 'app',
  },
  {
    key: 'project',
    tab: 'project',
  },
];
```

```

];

const MyTransfer = () => {
  const mockData = [];
  for (let i = 0; i < 20; i++) {
    mockData.push({
      key: i.toString(),
      title: `content${i + 1}`,
      description: `description of content${i + 1}`,
    });
  }

  return (
    <Transfer
      dataSource={mockData}
      targetKeys={['18']}
      selectedKeys={['3']}
      render={item => item.title}
    />
  );
};

const FormSizeDemo = () => {
  const [color, setColor] = useState({
    primaryColor: '#1890ff',
    errorColor: '#ff4d4f',
    warningColor: '#faad14',
    successColor: '#52c41a',
    infoColor: '#1890ff',
  });

  function onColorChange(nextColor) {
    const mergedNextColor = {
      ...color,
      ...nextColor,
    };
    setColor(mergedNextColor);
    ConfigProvider.config({
      theme: mergedNextColor,
    });
  }

  return (
    <Row gutter={16} wrap={false}>
      <Col flex="none">
        <Space direction="vertical" align="center">
          {/* Primary Color */}
          <SketchPicker
            presetColors={['#1890ff', '#25b864', '#ff6f00']}
            color={color.primaryColor}
            onChange={({ hex }) => {
              onColorChange({

```

```

        primaryColor: hex,
    });
  }
}
/>

<span style={{ color: 'var(--ant-primary-color)' }}>var(`--ant-primary-
color`)</span>

{/* Error Color */}
<SketchPicker
  presetColors={['#ff4d4f']}
  color={color.errorColor}
  onChange={({ hex }) => {
    onColorChange({
      errorColor: hex,
    });
  }}
/>

<span style={{ color: 'var(--ant-error-color)' }}>var(`--ant-error-color`)
</span>

{/* Warning Color */}
<SketchPicker
  presetColors={['#faad14']}
  color={color.warningColor}
  onChange={({ hex }) => {
    onColorChange({
      warningColor: hex,
    });
  }}
/>

<span style={{ color: 'var(--ant-warning-color)' }}>var(`--ant-warning-
color`)</span>

{/* Success Color */}
<SketchPicker
  presetColors={['#52c41a']}
  color={color.successColor}
  onChange={({ hex }) => {
    onColorChange({
      successColor: hex,
    });
  }}
/>

<span style={{ color: 'var(--ant-success-color)' }}>var(`--ant-success-
color`)</span>

{/* Info Color */}
<SketchPicker

```

```

        presetColors={['#1890ff']}
        color={color.infoColor}
        onChange={({ hex }) => {
            onColorChange({
                infoColor: hex,
            });
        }}
    />

    <span style={{ color: 'var(--ant-info-color)' }}>var(`--ant-info-color`)
</span>

    </Space>
</Col>

<Col flex="auto">
    <Space direction="vertical" split={<Divider />} style={{ width: '100%' }}
size={0}>
    {/* Primary Button */}
    <SplitSpace>
        <Button type="primary">Primary</Button>
        <Button>Default</Button>
        <Button type="dashed">Dashed</Button>
        <Button type="text">Text</Button>
        <Button type="link">Link</Button>
    </SplitSpace>

    {/* Danger Button */}
    <SplitSpace>
        <Button danger type="primary">
            Primary
        </Button>
        <Button danger>Default</Button>
        <Button danger type="dashed">
            Dashed
        </Button>
        <Button danger type="text">
            Text
        </Button>
        <Button danger type="link">
            Link
        </Button>
    </SplitSpace>

    {/* Ghost Button */}
    <SplitSpace style={{ background: 'rgb(190, 200, 200)' }}>
        <Button type="primary" ghost>
            Primary
        </Button>
        <Button ghost>Default</Button>
        <Button type="dashed" ghost>
            Dashed
        </Button>
    </SplitSpace>

```

```

<Button type="primary" ghost danger>
  Primary
</Button>
<Button ghost danger>
  Default
</Button>
<Button type="dashed" ghost danger>
  Dashed
</Button>
</SplitSpace>

{/* Typography */}
<SplitSpace>
  <Typography.Text type="success">Text (success)</Typography.Text>
  <Typography.Text type="warning">Text (warning)</Typography.Text>
  <Typography.Text type="danger">Text (danger)</Typography.Text>
  <Typography.Link href="https://ant.design" target="_blank">
    Link
  </Typography.Link>
  <Typography.Text copyable>Text</Typography.Text>

{/* Dropdown */}
<Dropdown
  overlay={
    <Menu>
      <Menu.Item>1st menu item</Menu.Item>
      <Menu.Item danger>a danger item</Menu.Item>
    </Menu>
  }
>
  <a className="ant-dropdown-link" onClick={e => e.preventDefault()}>
    Hover me <DownOutlined />
  </a>
</Dropdown>

{/* Spin */}
<Spin />
</SplitSpace>

{/* Menu - horizontal */}
<Row gutter={16}>
  <Col span={12}>
    <Menu mode="horizontal" defaultSelectedKeys={['mail']}>
      <Menu.Item key="mail" icon={<MailOutlined />}>
        Mail
      </Menu.Item>
      <Menu.SubMenu key="SubMenu" icon={<SettingOutlined />}
title="SubMenu">
        <Menu.ItemGroup title="Item 1">
          <Menu.Item key="setting:1">Option 1</Menu.Item>
          <Menu.Item key="setting:2">Option 2</Menu.Item>
        </Menu.ItemGroup>

```

```

        </Menu.SubMenu>
    </Menu>
</Col>
<Col span={12}>
    <Menu mode="horizontal" theme="dark" defaultSelectedKeys={['mail']}>
        <Menu.Item key="mail" icon={<MailOutlined />}>
            Mail
        </Menu.Item>
        <Menu.SubMenu key="SubMenu" icon={<SettingOutlined />}
title="SubMenu">
            <Menu.ItemGroup title="Item 1">
                <Menu.Item key="setting:1">Option 1</Menu.Item>
                <Menu.Item key="setting:2">Option 2</Menu.Item>
            </Menu.ItemGroup>
        </Menu.SubMenu>
    </Menu>
</Col>
</Row>

{/* Menu - vertical */}
<Row gutter={16}>
    <Col span={12}>
        <Menu mode="inline" defaultSelectedKeys={['mail']}>
            <Menu.Item key="mail" icon={<MailOutlined />}>
                Mail
            </Menu.Item>
            <Menu.SubMenu key="SubMenu" icon={<SettingOutlined />}
title="SubMenu">
                <Menu.ItemGroup title="Item 1">
                    <Menu.Item key="setting:1">Option 1</Menu.Item>
                    <Menu.Item key="setting:2">Option 2</Menu.Item>
                </Menu.ItemGroup>
            </Menu.SubMenu>
        </Menu>
    </Col>
    <Col span={12}>
        <Menu mode="vertical" theme="dark" defaultSelectedKeys={['mail']}>
            <Menu.Item key="mail" icon={<MailOutlined />}>
                Mail
            </Menu.Item>
            <Menu.SubMenu key="SubMenu" icon={<SettingOutlined />}
title="SubMenu">
                <Menu.ItemGroup title="Item 1">
                    <Menu.Item key="setting:1">Option 1</Menu.Item>
                    <Menu.Item key="setting:2">Option 2</Menu.Item>
                </Menu.ItemGroup>
            </Menu.SubMenu>
        </Menu>
    </Col>
</Row>

{/* Pagination */}

```

```

<Pagination showQuickJumper defaultCurrent={2} total={500} />

{/* Steps */}
<Steps current={1} percent={60}>
  <Steps.Step title="Finished" description="This is a description." />
  <Steps.Step
    title="In Progress"
    subTitle="Left 00:00:08"
    description="This is a description."
  />
  <Steps.Step title="Waiting" description="This is a description." />
</Steps>

{/* Steps - dot */}
<Steps current={2} status="error" progressDot>
  <Steps.Step title="Finished" description="You can hover on the dot." />
  <Steps.Step title="In Progress" description="You can hover on the dot."

/>

  <Steps.Step title="Error" description="You can hover on the dot." />
  <Steps.Step title="Waiting" description="You can hover on the dot." />
</Steps>

{/* Form - Input */}
<Form>
  <SplitSpace>
    <Form.Item>
      <Input {...inputProps} />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="success">
      <Input {...inputProps} />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="warning">
      <Input {...inputProps} />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="error">
      <Input {...inputProps} />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="validating">
      <Input {...inputProps} />
    </Form.Item>
  </SplitSpace>
</Form>

{/* Form - Select */}
<Form>
  <SplitSpace>
    <Form.Item>
      <Select {...selectProps} />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="success">
      <Select {...selectProps} />
    </Form.Item>
  </SplitSpace>
</Form>

```



```

    <Form.Item hasFeedback validateStatus="warning">
      <Select {...selectProps} />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="error">
      <Select {...selectProps} />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="validating">
      <Select {...selectProps} />
    </Form.Item>
  </SplitSpace>
</Form>

{/* Form - TreeSelect */}
<Form>
  <SplitSpace>
    <Form.Item>
      <TreeSelect {...treeSelectProps} />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="success">
      <TreeSelect {...treeSelectProps} />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="warning">
      <TreeSelect {...treeSelectProps} />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="error">
      <TreeSelect {...treeSelectProps} />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="validating">
      <TreeSelect {...treeSelectProps} />
    </Form.Item>
  </SplitSpace>
</Form>

{/* Form - InputNumber */}
<Form>
  <SplitSpace>
    <Form.Item>
      <InputNumber />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="success">
      <InputNumber />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="warning">
      <InputNumber />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="error">
      <InputNumber />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="validating">
      <InputNumber />
    </Form.Item>
  </SplitSpace>

```

```

</Form>

{/* Form - DatePicker */}
<Form>
  <SplitSpace>
    <Form.Item>
      <DatePicker />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="success">
      <DatePicker />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="warning">
      <DatePicker />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="error">
      <DatePicker />
    </Form.Item>
    <Form.Item hasFeedback validateStatus="validating">
      <DatePicker />
    </Form.Item>
  </SplitSpace>
</Form>

<SplitSpace>
  <Checkbox>Checkbox</Checkbox>

  <Radio.Group defaultValue="bamboo">
    <Radio value="bamboo">Bamboo</Radio>
    <Radio value="light">Light</Radio>
    <Radio value="little">Little</Radio>
  </Radio.Group>

  <Mentions placeholder="Mention by @">
    <Mentions.Option value="afc163">afc163</Mentions.Option>
    <Mentions.Option value="zombieJ">zombieJ</Mentions.Option>
    <Mentions.Option value="yesmeck">yesmeck</Mentions.Option>
  </Mentions>

  <Slider defaultValue={30} style={{ width: 100 }} />

  <Switch defaultChecked />
</SplitSpace>

<SplitSpace>
  <DatePicker.RangePicker />
  <TimePicker.RangePicker />
</SplitSpace>

<Row gutter={16}>
  <Col span={8}>
    {/* Card */}
    <Card

```

```

        style={{ width: '100%' }}
        tabList={carTabListNoTitle}
        tabBarExtraContent={(<a href="#">More</a>)}
      />
    </Col>
    <Col span={8}>
      { /* Table */ }
      <Table
        size="small"
        bordered
        rowSelection={{}}
        columns={[
          {
            title: 'Key',
            dataIndex: 'key',
            filters: [
              {
                text: 'Little',
                value: 'little',
              },
            ],
            sorter: (a, b) => a.key.length - b.key.length,
          },
        ]}
        dataSource={[
          {
            key: 'Bamboo',
          },
          {
            key: 'Light',
          },
          {
            key: 'Little',
          },
        ]}
      />
    </Col>
    <Col span={8}>
      { /* Table */ }
      <Tabs defaultActiveKey="1">
        <Tabs.TabPane tab="Tab 1" key="1">
          Content of Tab Pane 1
        </Tabs.TabPane>
        <Tabs.TabPane tab="Tab 2" key="2">
          Content of Tab Pane 2
        </Tabs.TabPane>
        <Tabs.TabPane tab="Tab 3" key="3">
          Content of Tab Pane 3
        </Tabs.TabPane>
      </Tabs>
    </Col>
  </Row>

```

```

<SplitSpace>
  <Tag color="success">success</Tag>
  <Tag color="processing">processing</Tag>
  <Tag color="error">error</Tag>
  <Tag color="warning">warning</Tag>
  <Tag color="default">default</Tag>
  <Tag.CheckableTag checked>CheckableTag</Tag.CheckableTag>
</SplitSpace>

<Row gutter={16}>
  <Col span={16}>
    <Timeline mode="alternate">
      <Timeline.Item>Create a services site 2015-09-01</Timeline.Item>
      <Timeline.Item color="gray">
        Solve initial network problems 2015-09-01
      </Timeline.Item>
      <Timeline.Item dot={<ClockCircleOutlined style={{ fontSize: '16px'
}} />} />>
        Sed ut perspiciatis unde omnis iste natus error sit voluptatem
accusantium
        doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo
inventore
        veritatis et quasi architecto beatae vitae dicta sunt explicabo.
      </Timeline.Item>
    </Timeline>
  </Col>

  <Col span={8}>
    <Tree treeData={treeData} height={200} defaultExpandAll checkable />
  </Col>
</Row>

{/* Alert */}
<Row gutter={16}>
  <Col span={6}>
    <Alert showIcon message="Success Text" type="success" />
  </Col>
  <Col span={6}>
    <Alert showIcon message="Info Text" type="info" />
  </Col>
  <Col span={6}>
    <Alert showIcon message="Warning Text" type="warning" />
  </Col>
  <Col span={6}>
    <Alert showIcon message="Error Text" type="error" />
  </Col>
</Row>

{/* Progress */}
<Row gutter={16}>
  <Col flex="auto">

```

```
        <Progress percent={30} />
        <Progress percent={70} status="exception" />
        <Progress percent={100} />
      </Col>
      <Col flex="none">
        <Progress type="circle" percent={75} />
        <Progress type="circle" percent={70} status="exception" />
        <Progress type="circle" percent={100} />
      </Col>
    </Row>

    <MyTransfer />
  </Space>
</Col>
</Row>
);
};
export default () => <FormSizeDemo />;
```