

## zh-CN

通过 `Form.useForm` 对表单数据域进行交互。

注意 `useForm` 是 [React Hooks](#) 的实现，只能用于函数组件，class 组件请查看下面的例子。

## en-US

Call form method with `Form.useForm`.

Note that `useForm` is a [React Hooks](#) that only works in functional component.

```
import { Form, Input, Button, Select } from 'antd';

const { Option } = Select;

const layout = {
  labelCol: { span: 8 },
  wrapperCol: { span: 16 },
};
const tailLayout = {
  wrapperCol: { offset: 8, span: 16 },
};

const Demo = () => {
  const [form] = Form.useForm();

  const onGenderChange = (value: string) => {
    switch (value) {
      case 'male':
        form.setFieldsValue({ note: 'Hi, man!' });
        return;
      case 'female':
        form.setFieldsValue({ note: 'Hi, lady!' });
        return;
      case 'other':
        form.setFieldsValue({ note: 'Hi there!' });
    }
  };

  const onFinish = (values: any) => {
    console.log(values);
  };

  const onReset = () => {
    form.resetFields();
  };

  const onFill = () => {
    form.setFieldsValue({
      note: 'Hello world!',
      gender: 'male',
    });
  };
};
```

```

    });
  };

  return (
    <Form {...layout} form={form} name="control-hooks" onFinish={onFinish}>
      <Form.Item name="note" label="Note" rules={[{ required: true }]}>
        <Input />
      </Form.Item>
      <Form.Item name="gender" label="Gender" rules={[{ required: true }]}>
        <Select
          placeholder="Select a option and change input text above"
          onChange={onGenderChange}
          allowClear
        >
          <Option value="male">male</Option>
          <Option value="female">female</Option>
          <Option value="other">other</Option>
        </Select>
      </Form.Item>
      <Form.Item
        noStyle
        shouldUpdate={(prevValues, currentValues) => prevValues.gender !==
currentValues.gender}
      >
        ({ getFieldValue }) =>
          getFieldValue('gender') === 'other' ? (
            <Form.Item name="customizeGender" label="Customize Gender" rules={[{
required: true }]}>
              <Input />
            </Form.Item>
          ) : null
        }
      </Form.Item>
      <Form.Item {...tailLayout}>
        <Button type="primary" htmlType="submit">
          Submit
        </Button>
        <Button htmlType="button" onClick={onReset}>
          Reset
        </Button>
        <Button type="link" htmlType="button" onClick={onFill}>
          Fill form
        </Button>
      </Form.Item>
    </Form>
  );
};

export default () => <Demo />;

```

```
#components-form-demo-control-hooks .ant-btn {  
  margin-right: 8px;  
}
```