

zh-CN

这个例子演示了一个表单中包含多个表单控件的情况。

en-US

This example demonstrates the case that a form contains multiple form controls.

```
import { Form, Input, Button, Space, Select } from 'antd';
import { MinusCircleOutlined, PlusOutlined } from '@ant-design/icons';

const { Option } = Select;

const areas = [
  { label: 'Beijing', value: 'Beijing' },
  { label: 'Shanghai', value: 'Shanghai' },
];

const sights = {
  Beijing: ['Tiananmen', 'Great Wall'],
  Shanghai: ['Oriental Pearl', 'The Bund'],
};

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

  const onFinish = values => {
    console.log('Received values of form:', values);
  };

  const handleChange = () => {
    form.setFieldsValue({ sights: [] });
  };

  return (
    <Form form={form} name="dynamic_form_nest_item" onFinish={onFinish}
    autoComplete="off">
      <Form.Item name="area" label="Area" rules={[{ required: true, message:
'Missing area' }]}>
        <Select options={areas} onChange={handleChange} />
      </Form.Item>
      <Form.List name="sights">
        {(fields, { add, remove }) => (
          <>
            {fields.map(field => (
              <Space key={field.key} align="baseline">
                <Form.Item
                  noStyle
                  shouldUpdate={({prevValues, curValues}) =>
                    prevValues.area !== curValues.area || prevValues.sights !==
curValues.sights
```

```

    }
  >
  { () => (
    <Form.Item
      {...field}
      label="Sight"
      name={[field.name, 'sight']}
      rules={[{ required: true, message: 'Missing sight' }]}
    >
      <Select disabled={!form.getFieldValue('area')} style={{ width:
130 }}>

        {(sights[form.getFieldValue('area')] || []).map(item => (
          <Option key={item} value={item}>
            {item}
          </Option>
        ))}
      </Select>
    </Form.Item>
  )}
</Form.Item>
<Form.Item
  {...field}
  label="Price"
  name={[field.name, 'price']}
  rules={[{ required: true, message: 'Missing price' }]}
>
  <Input />
</Form.Item>

  <MinusCircleOutlined onClick={() => remove(field.name)} />
</Space>
)}}
```

```

<Form.Item>
  <Button type="dashed" onClick={() => add()} block icon={<PlusOutlined
/>}>

    Add sights
  </Button>
</Form.Item>
</>
)}
</Form.List>
<Form.Item>
  <Button type="primary" htmlType="submit">
    Submit
  </Button>
</Form.Item>
</Form>
);
};

export default () => <Demo />;

```

