zh-CN

这里演示 `Form.Item` 内有多个元素的使用方式。 `<Form.Item name="field" />` 只会对它的直接子元素绑定表单功能，例如直接包裹了 Input/Select。如果控件前后还有一些文案或样式装点，或者一个表单项内有多个控件，你可以使用内嵌的 `Form.Item` 完成。你可以给 `Form.Item` 自定义 `style` 进行内联布局，或者添加 `noStyle` 作为纯粹的无样式绑定组件（类似 3.x 中的 `getFieldDecorator` ）。

```diff
- <Form.Item label="Field" name="field">
-   <Input />
- </Form.Item>
+ <Form.Item label="Field">
+   <Form.Item name="field" noStyle><Input /></Form.Item> // 直接包裹才会绑定表单
+   <span>description</span>
+ </Form.Item>
```

这里展示了三种典型场景：

- `Username`：输入框后面有描述文案或其他组件，在 `Form.Item` 内使用 `<Form.Item name="field" noStyle />` 去绑定对应子控件。
- `Address`：有两个控件，在 `Form.Item` 内使用两个 `<Form.Item name="field" noStyle />` 分别绑定对应控件。
- `BirthDate`：有两个内联控件，错误信息展示各自控件下，使用两个 `<Form.Item name="field" />` 分别绑定对应控件，并修改 `style` 使其内联布局。

> *注意，在 label 对应的 Form.Item 上不要在指定 `name` 属性，这个 Item 只作为布局作用。*

更复杂的封装复用方式可以参考下面的 `自定义表单控件` 演示。

en-US

This demo shows how to use `Form.Item` with multiple controls. `<Form.Item name="field" />` will only bind the control(Input/Select) which is the only children of it. Imagine this case: you added some text description after the Input, then you have to wrap the Input by an extra `<Form.Item name="field">`. `style` property of `Form.Item` could be useful to modify the nested form item layout, or use `<Form.Item noStyle />` to turn it into a pure form-binded component(like `getFieldDecorator` in 3.x).

```diff
- <Form.Item label="Field" name="field">
-   <Input />
- </Form.Item>
+ <Form.Item label="Field">
+   <Form.Item name="field" noStyle><Input /></Form.Item> // that will bind input
+   <span>description</span>
+ </Form.Item>
```

This demo shows three typical usages:

- `Username`: extra elements after control, using `<Form.Item name="field" noStyle />` inside `Form.Item` to bind Input.
- `Address`: two controls in one line, using two `<Form.Item name="field" noStyle />` to bind each control.

- `BirthDate` : two controls in one line with independent error message, using two `<Form.Item name="field" noStyle />` to bind each control, make layout inline by customizing `style` property.

> *Note that, in this case, no more `name` property should be left in Form.Item with label.*

See the `Customized Form Controls` demo below for more advanced usage.

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

const { Option } = Select;

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

  return (
    <Form name="complex-form" onFinish={onFinish} labelCol={{ span: 8 }} wrapperCol=
{{ span: 16 }}>
      <Form.Item label="Username">
        <Space>
          <Form.Item
            name="username"
            noStyle
            rules={[{ required: true, message: 'Username is required' }]}
          >
            <Input style={{ width: 160 }} placeholder="Please input" />
          </Form.Item>
          <Tooltip title="Useful information">
            <Typography.Link href="#API">Need Help?</Typography.Link>
          </Tooltip>
        </Space>
      </Form.Item>
      <Form.Item label="Address">
        <Input.Group compact>
          <Form.Item
            name={['address', 'province']}
            noStyle
            rules={[{ required: true, message: 'Province is required' }]}
          >
            <Select placeholder="Select province">
              <Option value="Zhejiang">Zhejiang</Option>
              <Option value="Jiangsu">Jiangsu</Option>
            </Select>
          </Form.Item>
          <Form.Item
            name={['address', 'street']}
            noStyle
            rules={[{ required: true, message: 'Street is required' }]}
          >
            <Input style={{ width: '50%' }} placeholder="Input street" />
          </Form.Item>
        </Input.Group>
      </Form.Item>
      <Form.Item label="BirthDate" style={{ marginBottom: 0 }}>
        <Form.Item
          name="year"
          rules={[{ required: true }]}
          style={{ display: 'inline-block', width: 'calc(50% - 8px)' }}
        >
          <Input placeholder="Input birth year" />
        </Form.Item>
        <Form.Item
          name="month"
          rules={[{ required: true }]}
          style={{ display: 'inline-block', width: 'calc(50% - 8px)', margin: '0
8px' }}
        >
          <Input placeholder="Input birth month" />
        </Form.Item>
      </Form.Item>
      <Form.Item label=" " colon={false}>
        <Button type="primary" htmlType="submit">
          Submit
        </Button>
      </Form.Item>
    </Form>
  );
};

export default () => <Demo />;
```