## zh-CN

动态增加、减少表单项。 `add` 方法参数可用于设置初始值。

## en-US

Add or remove form items dynamically. `add` function support config initial value.

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

const formItemLayout = {
  labelCol: {
    xs: { span: 24 },
    sm: { span: 4 },
  },
  wrapperCol: {
    xs: { span: 24 },
    sm: { span: 20 },
  },
};
const formItemLayoutWithOutLabel = {
  wrapperCol: {
    xs: { span: 24, offset: 0 },
    sm: { span: 20, offset: 4 },
  },
};

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

  return (
    <Form name="dynamic_form_item" {...formItemLayoutWithOutLabel} onFinish=
{onFinish}>
      <Form.List
        name="names"
        rules={[
          {
            validator: async (_, names) => {
              if (!names || names.length < 2) {
                return Promise.reject(new Error('At least 2 passengers'));
              }
            },
          },
        ]}
      >
        {(fields, { add, remove }, { errors }) => (
          <>
            {fields.map((field, index) => (
```

```jsx
              <Form.Item
                {...(index === 0 ? formItemLayout : formItemLayoutWithOutLabel)}
                label={index === 0 ? 'Passengers' : ''}
                required={false}
                key={field.key}
              >
                <Form.Item
                  {...field}
                  validateTrigger={['onChange', 'onBlur']}
                  rules={[
                    {
                      required: true,
                      whitespace: true,
                      message: "Please input passenger's name or delete this
field.",
                    },
                  ]}
                  noStyle
                >
                  <Input placeholder="passenger name" style={{ width: '60%' }} />
                </Form.Item>
                {fields.length > 1 ? (
                  <MinusCircleOutlined
                    className="dynamic-delete-button"
                    onClick={() => remove(field.name)}
                  />
                ) : null}
              </Form.Item>
            ))}
            <Form.Item>
              <Button
                type="dashed"
                onClick={() => add()}
                style={{ width: '60%' }}
                icon={<PlusOutlined />}
              >
                Add field
              </Button>
              <Button
                type="dashed"
                onClick={() => {
                  add('The head item', 0);
                }}
                style={{ width: '60%', marginTop: '20px' }}
                icon={<PlusOutlined />}
              >
                Add field at head
              </Button>
              <Form.ErrorList errors={errors} />
            </Form.Item>
          </>
        )}
```

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

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

```css
.dynamic-delete-button {
  position: relative;
  top: 4px;
  margin: 0 8px;
  color: #999;
  font-size: 24px;
  cursor: pointer;
  transition: all 0.3s;
}
.dynamic-delete-button:hover {
  color: #777;
}
.dynamic-delete-button[disabled] {
  cursor: not-allowed;
  opacity: 0.5;
}
```