



React Recharts



1. Recharts란?

개념

- **Recharts**는 React 기반의 **간단하고 직관적인 차트 라이브러리**입니다.
- **D3.js** 기반으로 만들어졌으며, **컴포넌트 기반 구조**로 다양한 차트를 손쉽게 만들 수 있습니다.
- 사용성과 확장성 모두 뛰어나 실무에서 많이 활용됩니다.

특징

- React 컴포넌트 기반 구조
- 커스터마이징 및 스타일링 용이
- 반응형(responsive) 지원
- 직관적인 API와 예측 가능한 결과

설치

npm install recharts

공식 사이트

<https://recharts.org/en-US/>



2. 기본 사용 구조

예시 데이터

```
const data = [  
  { name: '월', value: 400 },  
  { name: '화', value: 300 },  
  { name: '수', value: 500 },  
];
```

LineChart 기본 예제

```
import { LineChart, Line, XAxis, YAxis, CartesianGrid, Tooltip, Legend } from 'recharts';
```

```
<LineChart width={500} height={300} data={data}>  
  <CartesianGrid strokeDasharray="3 3" />  
  <XAxis dataKey="name" />  
  <YAxis />
```

```
<Tooltip />
<Legend />
<Line type="monotone" dataKey="value" stroke="#8884d8" />
</LineChart>
```

✅ 핵심 구성요소는 차트 종류 컴포넌트(`LineChart` 등)와 내부 구성요소(`XAxis`, `Tooltip`, `Legend` 등)

3. 주요 차트 종류

차트 종류	컴포넌트 이름
선형 차트	<code>LineChart</code>
막대 차트	<code>BarChart</code>
파이 차트	<code>PieChart</code>
영역 차트	<code>AreaChart</code>
레이더 차트	<code>RadarChart</code>
반지형 차트	<code>RadialBarChart</code>

✅ 대부분의 차트는 유사한 구성(`data`, `XAxis`, `YAxis`, `Tooltip` 등)을 공유함

4. 실전 예제

`BarChart` (막대 차트)

```
import { BarChart, Bar, XAxis, YAxis, Tooltip, Legend } from 'recharts';
```

```
<BarChart width={500} height={300} data={data}>
  <XAxis dataKey="name" />
  <YAxis />
  <Tooltip />
  <Legend />
  <Bar dataKey="value" fill="#82ca9d" />
</BarChart>
```

📌 PieChart (원형 차트)

```
import { PieChart, Pie, Cell, Tooltip } from 'recharts';

const COLORS = ['#0088FE', '#00C49F', '#FFBB28'];

<PieChart width={400} height={400}>
  <Pie
    data={data}
    dataKey="value"
    nameKey="name"
    cx="50%"
    cy="50%"
    outerRadius={80}
    fill="#8884d8"
    label
  >
    {data.map((entry, index) => (
      <Cell key={`cell-${index}`} fill={COLORS[index % COLORS.length]} />
    ))}
  </PieChart>
  <Tooltip />
</PieChart>
```

📌 5. 커스터마이징

✅ Tooltip 커스터마이징

```
<Tooltip formatter={(value) => `${value}명`} />
```

✅ 범례 위치 지정

```
<Legend verticalAlign="top" height={36} />
```

✅ 반응형 차트

```
import { ResponsiveContainer } from 'recharts';

<ResponsiveContainer width="100%" height={300}>
  <LineChart data={data}>...</LineChart>
</ResponsiveContainer>
```



6. 여러 데이터 처리



복수 데이터 표현

```
const data = [
  { name: '월', A: 400, B: 240 },
  { name: '화', A: 300, B: 456 },
];
<LineChart data={data}>
  <Line dataKey="A" stroke="#8884d8" />
  <Line dataKey="B" stroke="#82ca9d" />
</LineChart>
```



하나의 차트에서 여러 데이터 값을 표현 가능



7. 주의사항

- 차트 크기를 지정하지 않으면 렌더링 되지 않음 (`width`, `height` 또는 `ResponsiveContainer` 필수)
 - 데이터 구조는 반드시 일관되게 유지 (key 누락 시 오류 발생)
 - 다이내믹 데이터의 경우 리렌더링 타이밍에 주의 (`useEffect`로 상태 관리)
-



8. 마무리



핵심 요약

- Recharts는 React 기반으로 구성된 차트 라이브러리로, 사용법이 직관적이고 확장성이 뛰어남
- 기본 차트 유형만 익혀도 다양한 실무 시각화에 응용 가능
- 반응형, 커스터마이징, 다중 데이터 대응까지 지원되어 실무에 최적화됨