Recharts | Reading Material

Concepts in Focus

1. Third-Party Package `recharts`

NPM contains **recharts**, a third-party package to display charts in your application.

It supports different types of charts like:

- Bar Chart
- Pie Chart

- Area Chart
- Composed Chart, etc.

It supports different types of visualization methods like:

Cartesian:

- Area
- Bar
- Line, etc.

Polar:

- Pie
- Radar
- Radial Bar

Installation Command:

```
npm install recharts
```

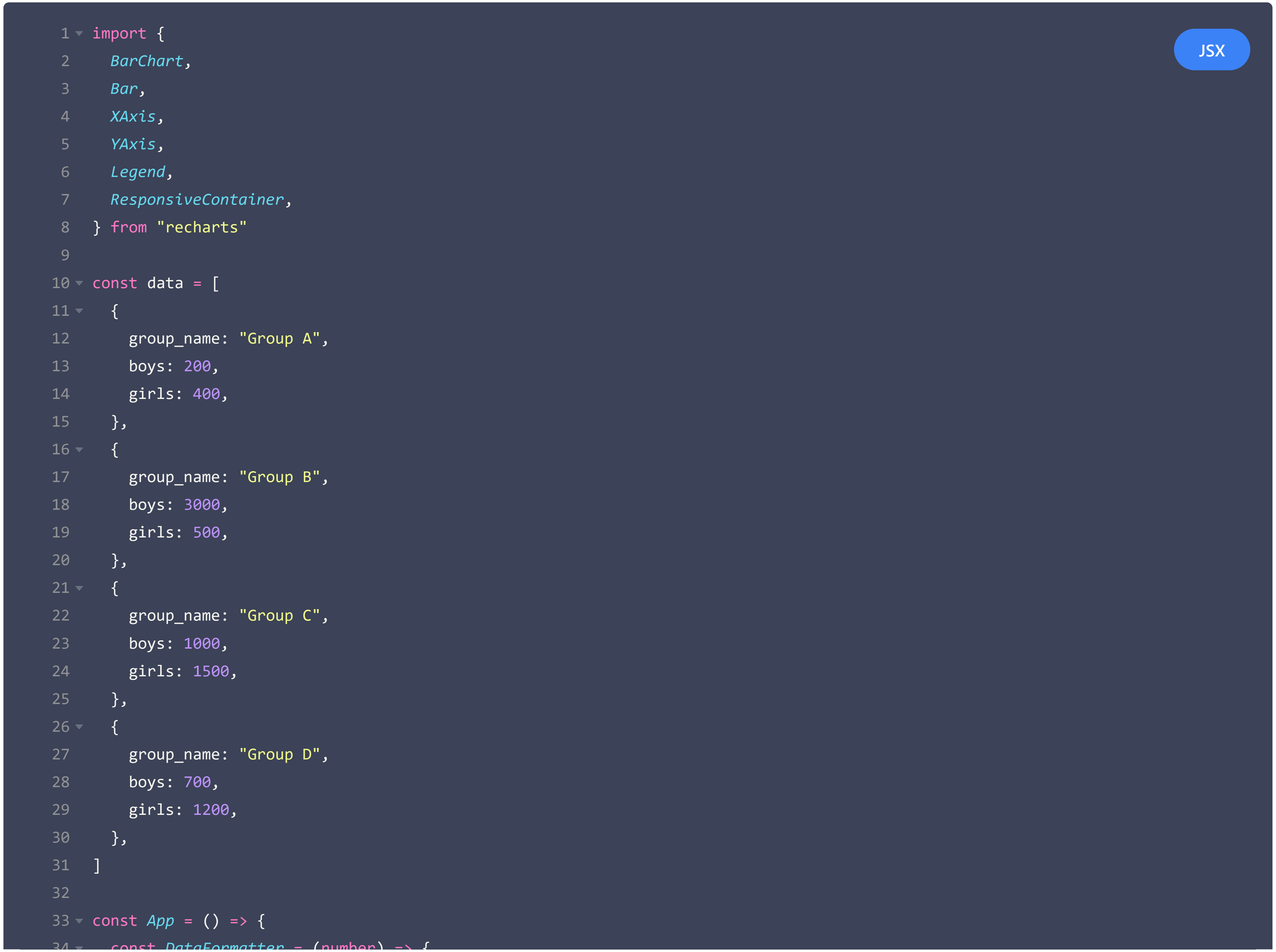
1.1 Advantages

- Responsive
- Built for React, from scratch
- Customizable

2. Bar Chart

The **BarChart** Component represents the container of the Bar Chart.

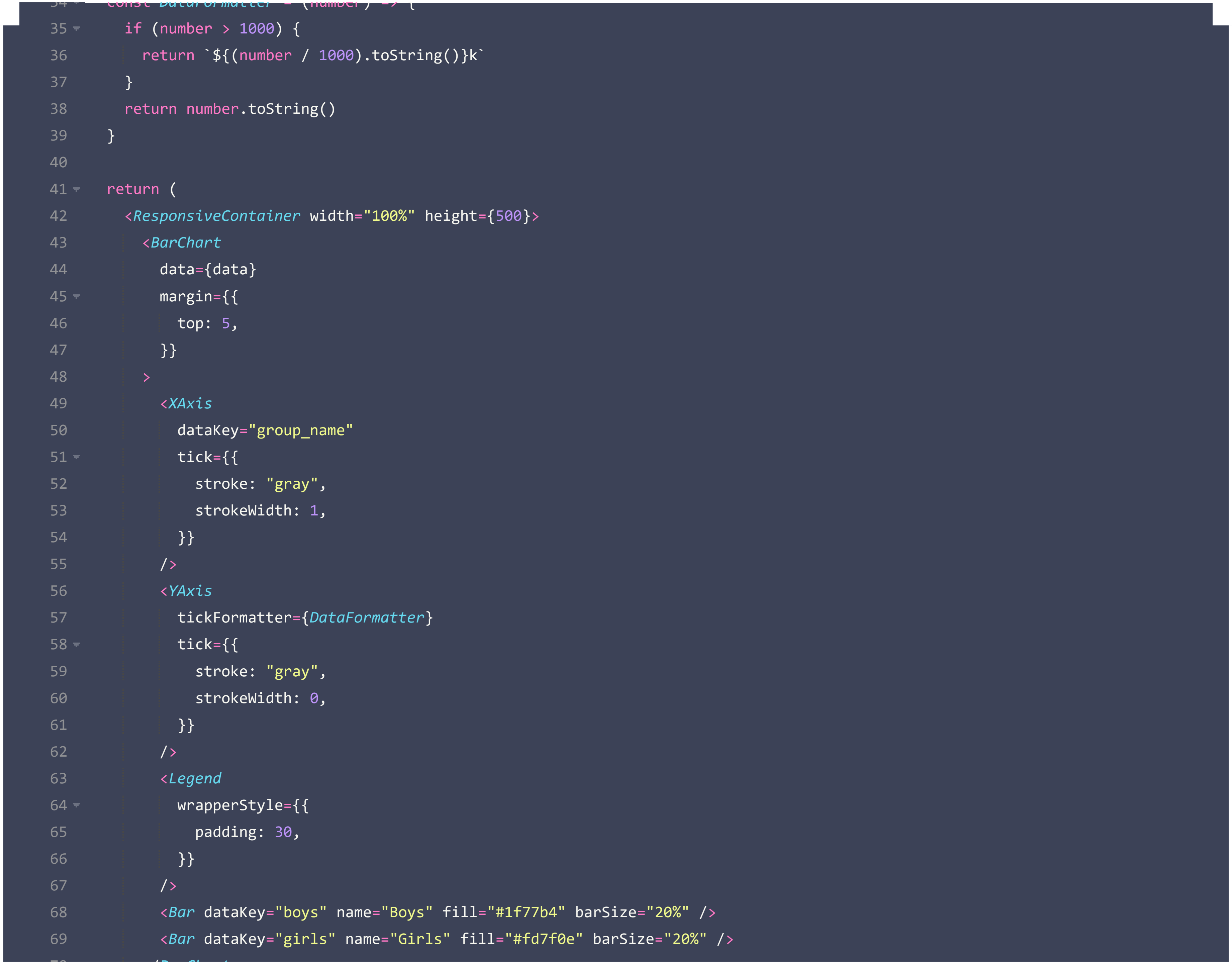
Example:


JSX

```jsx
import {
  BarChart,
  Bar,
  XAxis,
  YAxis,
  Legend,
  ResponsiveContainer,
} from "recharts"

const data = [
  {
    group_name: "Group A",
    boys: 200,
    girls: 400,
  },
  {
    group_name: "Group B",
    boys: 3000,
    girls: 500,
  },
  {
    group_name: "Group C",
    boys: 1000,
    girls: 1500,
  },
  {
    group_name: "Group D",
    boys: 700,
    girls: 1200,
  },
]

const App = () => {
  const DataFormatter = (number) => {
```



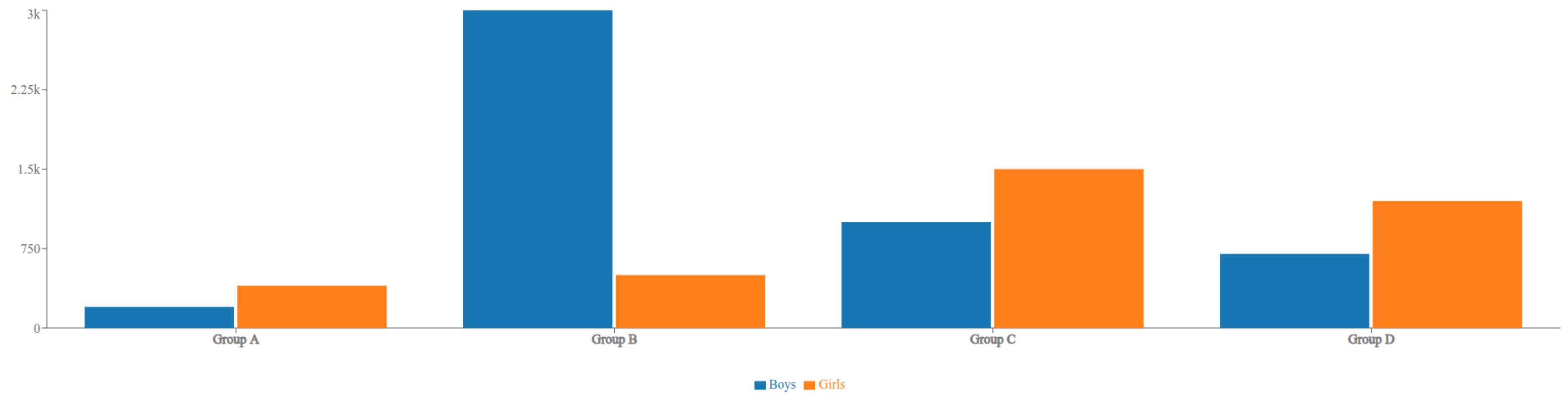
```
  const DataFormatter = (number) => {
    if (number > 1000) {
      return `${(number / 1000).toString()}k`
    }
    return number.toString()
  }

  return (
    <ResponsiveContainer width="100%" height={500}>
      <BarChart
        data={data}
        margin={{
          top: 5,
        }}
      >
        <XAxis
          dataKey="group_name"
          tick={{
            stroke: "gray",
            strokeWidth: 1,
          }}
        />
        <YAxis
          tickFormatter={DataFormatter}
          tick={{
            stroke: "gray",
            strokeWidth: 0,
          }}
        />
        <Legend
          wrapperStyle={{
            padding: 30,
          }}
        />
        <Bar dataKey="boys" name="Boys" fill="#1f77b4" barSize="20%" />
        <Bar dataKey="girls" name="Girls" fill="#fd7f0e" barSize="20%" />
      </BarChart>
    </ResponsiveContainer>
  )
}

export default App
```

Collapse

Output:



3. Components in Bar Chart

The

`recharts` supports different Components for the Bar Chart. Below are some of the most commonly used Components.

3.1 ResponsiveContainer

It is a container Component to make charts adapt to the size of the parent container.

Props:

We can provide different props to the ReactJS **ResponsiveContainer** Component. Below are some of the most commonly used props.

Prop	Default Value
width	'100%' (value can be percentage string or number)
height	'100%' (value can be percentage string or number)

Note

One of the props `width` and `height` should be a percentage string in the `ResponsiveContainer` Component.

3.2 XAxis

The **XAxis** Component represents the X-Axis of a Chart.

Props:

We can provide different props to the ReactJS **XAxis** Component. Below are some of the most commonly used props.

Prop	Description	Default Value
dataKey	The key of the object in `data` that we want to display it's value on the axis	No default value (value can be string or number)
tick	Represents a tick	No default value. If false - No ticks will be drawn, object - Configuration of ticks, React element - Custom react element for drawing ticks (value can be boolean, object or React element)
tickFormatter	The formatter function of tick	No default value (Function)

Example - tickFormatter:

If we want to show the thousands in the form of

`k` on the tick, the formatter function would be:

JSX

```jsx
const DataFormatter = (number) => {
  if (number > 1000) {
    return `${(number / 1000).toString()}k`
  }
  return number.toString()
}
```

3.3 YAxis

The **YAxis** Component represents the Y-Axis of a Chart.

The Props of the **YAxis** Component are similar to the **XAxis** Component.

3.4 Legend

The **Legend** Component represents the legend of a Chart.

By default, the content of the legend is generated by the name of

`Line` , `Bar` , `Area` , etc. If no name has been set, the prop `dataKey` is used to generate the content of the legend.

Props:

We can provide different props to the ReactJS **Legend** Component. Below are some of the most commonly used props.

Prop	Description	Default Value
iconType	The type of icon in the legend item	No default value (value can be 'line', 'plainline', 'square', 'rect', 'circle', 'cross', 'diamond','star', 'triangle', or 'wye')

Prop	Description	Default Value
layout	The layout of legend items	'horizontal' (value can be 'horizontal' or 'vertical')
verticalAlign	The alignment of legend items in vertical direction	'middle' (value can be 'top', 'middle', or 'bottom')
align	The alignment of legend items in horizontal direction	'center' (value can be 'left', 'center', or 'right')
wrapperStyle	The style of the legend container	No default value (value can be React Inline styles)

3.5 Bar

The **Bar** Component represents a bar in the Chart.

Props:

We can provide different props to the ReactJS **Bar** Component. Below are some of the most commonly used props.

Prop	Description	Default Value
dataKey	The key of the object in `data` that we want to display it's value	No default value (value can be string or number)
name	The name of the bar	No default value (value can be string or number)
fill	The color to fill the rectangle in a bar	(value can be given color in hexCode or string format)
barSize	The width or height of the bar	No default value (value can be number)

Note

The value of the prop `name` is used in tooltip and legend to represent a bar/pie. If no value was set, the value of `dataKey` will be used alternatively.

4. PieChart

The **PieChart** Component represents the container of the Pie Chart.

Example:

JSX

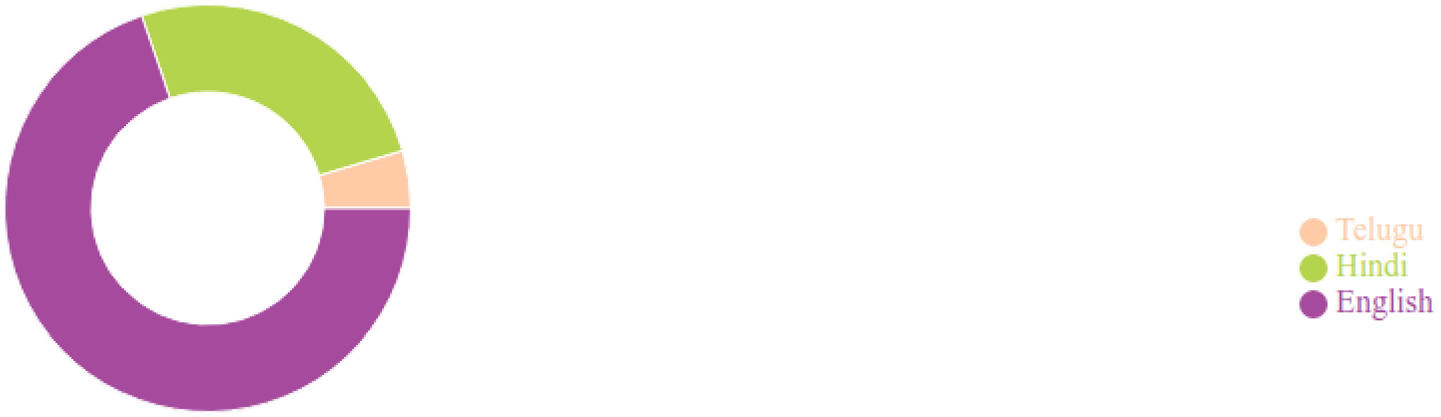
```jsx
import { PieChart, Pie, Legend, Cell, ResponsiveContainer } from "recharts"

const data = [
  {
    count: 809680,
    language: "Telugu",
  },
  {
    count: 4555697,
    language: "Hindi",
  },
  {
    count: 12345657,
    language: "English",
  },
]

const App = () => {
  return (
    <ResponsiveContainer width="100%" height={300}>
      <PieChart>
        <Pie
        <Pie
          cx="70%"
          cy="40%"
          data={data}
          startAngle={0}
          endAngle={360}
          innerRadius="40%"
          outerRadius="70%"
          dataKey="count"
        >
          <Cell name="Telugu" fill="#fecba6" />
          <Cell name="Hindi" fill="#b3d23f" />
          <Cell name="English" fill="#a44c9e" />
        </Pie>
        <Legend
          iconType="circle"
          layout="vertical"
          verticalAlign="middle"
          align="right"
        />
      </PieChart>
    </ResponsiveContainer>
  )
}

export default App
```

Output:



5. Components in Pie Chart

The

`recharts` supports different Components for the Bar Chart. Below are some of the most commonly used Components.

5.1 Pie

The **Pie** Component represents a pie in the Chart.

Props:

We can provide different props to the ReactJS **Pie** Component. Below are some of the most commonly used props.

Prop	Description	Default Value
cx	The x-axis coordinate of a center point	'50%'. If set a percentage, the final value is obtained by multiplying the percentage of container width (value can be percentage string or number)

Prop	Description	Default Value
cy	The y-axis coordinate of a center point	'50%'. If set a percentage, the final value is obtained by multiplying the percentage of container height (value can be percentage string or number)
data	The source data in which each element is an object	No default value (value can be Array)
startAngle	The start angle of the first sector	0 (value can be number)
endAngle	The end angle of the last sector, which should be unequal to startAngle	360 (value can be number)
innerRadius	The inner radius of all the sectors	0 (value can be percentage or number)
OuterRadius	The outer radius of all the sectors	0 (value can be percentage or number)
dataKey	The key of the object in `data` that we want to display it's value on the sector	No default value

Note

If a percentage is set to the props `innerRadius` or `outerRadius` , the final value is obtained by multiplying the percentage of `maxRadius` which is calculated by the `width` , `height` , `cx` , and `cy` .

5.2 Cell

The **Cell** Component represents the cell of a Chart.

It can be wrapped by a

`Pie` , `Bar` , or `RadialBar` Components to specify the attributes of each child.

Props:

We can provide different props to the ReactJS **Cell** Component. Below are some of the most commonly used props.

Prop	Description	Default Value
name	The name of the cell	No default value (can be a string. This value can be taken as the content of the legend)
fill	The color to fill the cell	(value can be any color in hexCode or string format)

Note

The `ResponsiveContainer` and `Legend` Components in Pie Chart are similar to the Components in Bar Chart.

6. Reference

To know more about the

`recharts` , you can refer to **this**.

MARKED AS COMPLETE

Submit Feedback