

Complete Waterfall Chart Tutorial for AI Agents

TABLE OF CONTENTS

1. [What is a Waterfall Chart?](#)
 2. [When to Use Waterfall Charts](#)
 3. [Data Structure Requirements](#)
 4. [Step-by-Step Code Tutorial](#)
 5. [Prompt Template for AI Agents](#)
 6. [Common Mistakes to Avoid](#)
 7. [Advanced Customizations](#)
-

WHAT IS A WATERFALL CHART?

A waterfall chart shows how an initial value is affected by a series of positive or negative changes, leading to a final value.

Visual Pattern:

Start Value: €1,000
- Cost A: -€200 → Running Total: €800
+ Revenue B: +€150 → Running Total: €950
- Cost C: -€100 → Running Total: €850
Final Value: €850

Business Use Cases:

- Revenue breakdowns (Total → Net after deductions)
 - Budget analysis (Planned → Actual with variances)
 - Risk cascades (Total Exposure → Addressable Risk)
 - Profit bridges (Gross → Net Profit)
-

WHEN TO USE WATERFALL CHARTS

GOOD USE CASES:

- **Sequential changes** to a starting value

- **Cumulative impact** of multiple factors
- **Bridge analysis** (e.g., 2023 Revenue → 2024 Revenue)
- **Risk funnels** (Total → High Priority subset)

✗ AVOID WATERFALL FOR:

- **Non-cumulative comparisons** (use bar charts)
 - **Time series trends** (use line charts)
 - **Part-to-whole relationships** (use pie/stacked bar)
-

DATA STRUCTURE REQUIREMENTS

CORE DATA MODEL:

```
javascript
```

```
const waterfallData = [
  {
    name: "Starting Value",    // Label for the bar
    value: 1000,              // Actual value (for first bar)
    change: 0,                // Change amount (0 for starting)
    type: "total",            // Type: "total", "increase", "decrease"
    color: "#0066CC"          // Bar color
  },
  {
    name: "Deduction A",
    value: 0,                 // Not used for intermediate bars
    change: -200,             // Negative = decrease
    type: "decrease",
    color: "#D32F2F"
  },
  {
    name: "Addition B",
    value: 0,
    change: +150,             // Positive = increase
    type: "increase",
    color: "#4CAF50"
  },
  {
    name: "Final Value",
    value: 0,
    change: 0,                // Final bar shows cumulative
    type: "total",
    color: "#FF5722"
  }
];
```

CALCULATED FIELDS (Required for Recharts):

javascript

// Transform data to include cumulative positions

```
const processedData = waterfallData.map((item, index) => {  
  // Calculate running total up to this point  
  let cumulative = 0;  
  for (let i = 0; i <= index; i++) {  
    cumulative += waterfallData[i].value || waterfallData[i].change;  
  }  
  
  return {  
    ...item,  
    start: index === 0 ? 0 : cumulative - (item.value || item.change),  
    end: cumulative,  
    displayValue: Math.abs(item.change || item.value),  
    cumulative: cumulative  
  };  
});
```

Key Fields Explained:

- `start`: Where the bar begins on Y-axis (cumulative from previous)
 - `end`: Where the bar ends on Y-axis
 - `displayValue`: The height of the visible bar (absolute value)
 - `cumulative`: Running total (for reference lines)
-



STEP-BY-STEP CODE TUTORIAL

STEP 1: Basic Waterfall Structure

```
jsx  
  
import React from 'react';  
import { Bar
```