

HTML Content Tooltip Design Guide

Creating Interactive KPI Tooltips in Power BI

This comprehensive guide demonstrates how to implement sophisticated HTML content tooltips in Power BI to enhance KPI visualization and user experience. By leveraging custom HTML/CSS design within DAX measures, organizations can create dynamic, visually appealing tooltips that provide detailed insights while maintaining smooth hover interactions.

Concept Overview

What is HTML Content Tooltip Design?

HTML Content Tooltip Design is an advanced Power BI technique that enables developers to create custom, interactive tooltips using HTML and CSS embedded within DAX measures. This approach allows for



HTML Content

Daniel Marsh-Patrick

★★★★★ 4.6 (33)

Overview

[Ratings + reviews](#)

Visualise column or measure values as HTML in your Power BI reports.

If you want a certified edition of HTML Content, check out [HTML Content \(lite\)](#)

Report authors can use the visual to write their own columns and measures using DAX, to create dynamic HTML content, or render existing content from their data model that contains HTML or Markdown in their reports.

Add

Download Sample

[Sample Instructions](#)

- Enhanced Visual Appeal: Custom styling beyond Power BI's default tooltip limitations
- Dynamic Content: Real-time data updates based on user interactions
- Improved User Experience: Smooth hover effects and professional presentation
- KPI Optimization: Better display of key performance indicators with contextual information

Implementation Strategy

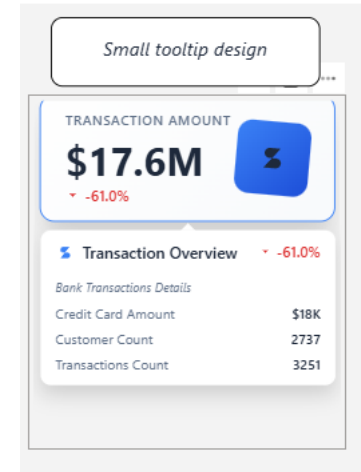
Phase 1: Planning and Design

Step 1: Define Requirements

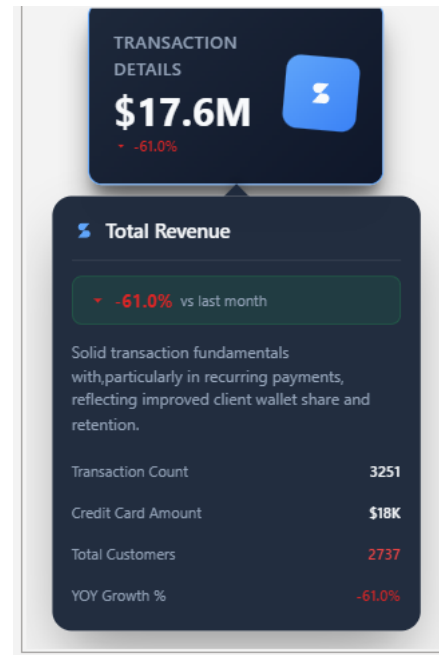
- Identify the key performance indicators (KPIs) to display
- Determine the target audience and their information needs
- Establish design guidelines and brand consistency requirements
- Plan for responsive design across different devices

Step 2: Create Visual Mockups

- Design the tooltip layout with proper spacing and margins
- Define color schemes for different themes (light/dark mode)
- Establish typography hierarchy and readability standards
- Plan for different data scenarios (high/low values, null data, etc.)



This entire height & width contains the visual to display properly.



This is based on the what you need to achieve

Technical Implementation

Measures

```
1 KPI Card HTML Dark & Light SVG Bar =
2 VAR Total_Transactions_Amount = [Total Transaction Amount]
3 VAR YOY_Transaction_Growth = [YOY Transaction Growth]
4 VAR Credit_Card_Amount = [_02_Total Credit Card Fees Converted]
5 VAR Customers = [Total Customers]
6 VAR Total_Transactions = [Total Transactions]
7
8 -- Get theme colors from Theme table
9 VAR SelectedThemeMode = [_01_SelectedThemeMode]
10
11 -- Theme Colors
12 VAR CardBackground = [_02_CardBackgroundColor]
13 VAR CardBackgroundGradient = [_03_CardBackgroundGradientColor]
14 VAR TextPrimary = [_04_TextPrimaryColor]
15 VAR TextSecondary = [_05_TextSecondaryColor]
16 VAR CardBorder = [_06_CardBorder]
17 VAR IconBackground = [_07_IconBackground]
18 VAR IconBackgroundGradient = [_08_IconBackgroundGradient]
19 VAR AccentColor = [_09_AccentColor]
20 VAR AccentColorDark = [_10_AccentColorDark]
21 VAR TooltipBackground = [_11_TooltipBackground]
22 VAR TooltipText = [_12_TooltipText]
23 VAR TooltipTextSecondary = [_13_TooltipTextSecondary]
24
25 -- Format values for display
26 VAR Total_Transaction_Amount_Formatted = FORMAT([Total Transaction Amount] / 1000000, "$#,##0.0") & "M"
27 VAR YOY_Transaction_Growth_Formatted =
28     IF(
29         YOY_Transaction_Growth >= 0,
30         "+" & FORMAT(YOY_Transaction_Growth, "0.0%"),
31         FORMAT(YOY_Transaction_Growth, "0.0%")
32     )
33
34 -- Determine growth color
35 VAR GrowthColor = IF(YOY_Transaction_Growth >= 0, "#059669", "#dc2626")
36 VAR GrowthIcon = IF(YOY_Transaction_Growth >= 0, "M7 1415-5 5 5", "M17 101-5 5-5-5")
37
38 -- Shadow colors based on theme
39 VAR ShadowColor = IF(SelectedThemeMode = "Dark", "rgba(0, 0, 0, 0.3)", "rgba(0, 0, 0, 0.05)")
40 VAR ShadowColorHover = IF(SelectedThemeMode = "Dark", "rgba(0, 0, 0, 0.5)", "rgba(0, 0, 0, 0.1)")
```

Step 3: Develop DAX Measures

Create comprehensive measures that will populate your tooltips with dynamic data:

Step 4: Build HTML Structure

Create semantic HTML markup for accessibility

Implement proper container hierarchy

Add appropriate CSS classes for styling hooks

Ensure cross-browser compatibility

Step 5: Implement CSS Styling

Define responsive layout properties

Create theme variables for easy customization

Implement hover effects and transitions

Optimize for performance and smooth rendering

Data Layer

Source Data: Bank transaction data (as referenced in the example)

Measures: Custom DAX calculations for dynamic values

Filters: Context-aware filtering for relevant data display

Presentation Layer

HTML Structure: Semantic markup for content organization

CSS Styling: Modern design with theme support

JavaScript (if needed): Enhanced interactivity and animations

Integration Layer

Power BI Integration: Seamless embedding within Power BI reports

Theme Management: Automatic adaptation to Power BI's light/dark modes

Performance Optimization: Efficient rendering and memory usage



Implementation Example

HTML Block

```
88 <head>
92   <style>
361 </style>
362 </head>
363 <body>
364   <div class='kpi-container'>
365     <div class='kpi-card'>
366       <div class='card-content'>
367         <div class='card-info'>
368           <div class='card-title'>Transaction Amount</div>
369           <div class='card-value'>" & Total_Trasnaction_Amount_Formatted & "</div>
370           <div class='card-growth'>
371             <svg class='growth-icon' viewBox='0 0 24 24'>
372               <path d='" & GrowthIcon & "'/>
373             </svg>
374             <span>" & YOY_Transaction_Growth_Formatted & "</span>
375           </div>
376         </div>
377         <div class='card-icon'>
378           <svg viewBox='0 0 24 24'>
379             <path d='M12 2v20M17 5H9.5a3.5 3.5 0 0 0 7h5a3.5 3.5 0 0 1 0 7H6'/'>
380           </svg>
381         </div>
382       </div>
383     </div>
384     <div class='tooltip'>
385       <div class='tooltip-header'>
386         <div class='tooltip-title'>
387           <svg width='16' height='16' viewBox='0 0 24 24' fill='" & AccentColor & "'>
388             <path d='M12 2v20M17 5H9.5a3.5 3.5 0 0 0 7h5a3.5 3.5 0 0 1 0 7H6'/'>
389           </svg>
390           Transaction Overview
391         </div>
392         <div class='tooltip-growth'>
393           <svg width='10' height='10' viewBox='0 0 24 24' fill='" & GrowthColor & "'>
394             <path d='" & GrowthIcon & "'/>
395           </svg>
396           <span>" & YOY_Transaction_Growth_Formatted & "</span>
397         </div>

```

CSS Block

```
<head>
  <style>
    * {
      box-sizing: border-box;
    }

    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      background: transparent;
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
      padding: 20px;
      position: relative;
    }

    .kpi-container {
      display: flex;
      justify-content: center;
      align-items: center;
      width: 100%;
      position: relative;
    }

    .kpi-card {
      position: relative;
      width: 241px;
      height: 150px;
      background: linear-gradient(135deg, " & CardBackground & " 0%, " & CardBackgroundGradient & " 100%);
      border-radius: 12px;
      box-shadow: 0 4px 6px " & ShadowColor & ", 0 1px 3px " & ShadowColor & ";
      border: 1px solid " & CardBorder & ";
      padding: 20px;
      cursor: pointer;
      transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
      overflow: visible;
    }

```

Best Practices & Guidelines

Performance Optimization

Minimize HTML Complexity: Keep markup lean and efficient

Optimize CSS: Use efficient selectors and avoid unnecessary calculations

Cache Measures: Implement proper DAX optimization techniques

Test Rendering: Ensure smooth performance across different data volumes

Accessibility Standards

Color Contrast: Maintain WCAG AA compliance (4.5:1 ratio minimum)

Semantic HTML: Use appropriate HTML elements for screen readers

Alternative Text: Provide context for visual elements

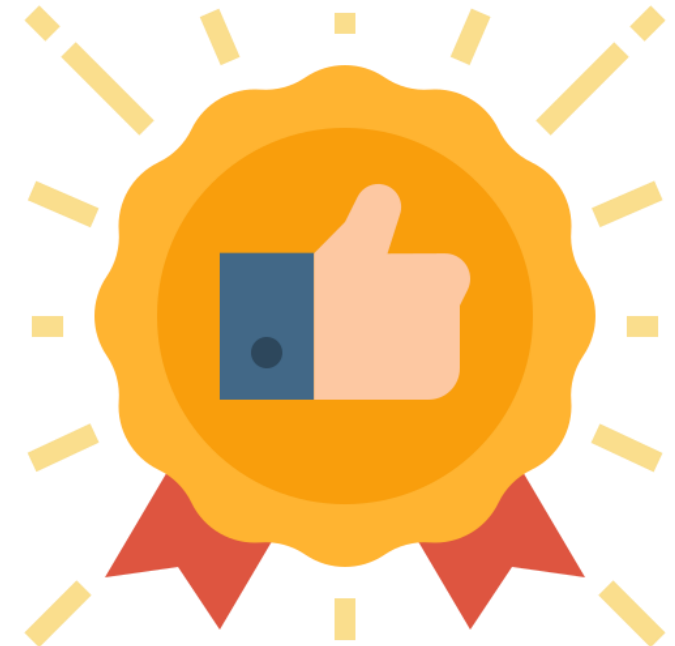
Maintenance and Updates

Version Control: Track changes to HTML/CSS implementations

Documentation: Maintain clear documentation for future modifications

Testing: Regular testing across different browsers and devices

Feedback Loop: Collect user feedback for continuous improvement



Troubleshooting Common Issues

Rendering Problems

Issue: Tooltips not displaying correctly

Solution: Verify HTML syntax and CSS property support

Prevention: Use cross-browser compatible CSS properties

Performance Issues

Issue: Slow tooltip rendering

Solution: Optimize DAX measures and reduce HTML complexity

Prevention: Implement efficient caching strategies

Theme Inconsistencies

Issue: Colors not matching Power BI theme

Solution: Use CSS custom properties linked to Power BI theme variables

Prevention: Test thoroughly in both light and dark modes



Conclusion

HTML Content Tooltip Design represents a powerful approach to enhancing Power BI reporting capabilities. By combining the flexibility of web technologies with Power BI's robust data platform, organizations can create professional, engaging, and informative data visualizations that drive better decision-making.

The implementation of custom HTML tooltips requires careful planning, attention to design principles, and thorough testing. However, the investment in creating these enhanced user experiences pays dividends in improved user adoption, better data comprehension, and more effective business intelligence outcomes.

