

## Analyse und Ursachen

Das Problem lag nicht an einem falschen JavaScript, sondern an fehlenden CSS-Stilregeln für das Custom-Element und seine Unterkomponenten. Im Standard-Lovelace-Layout (flex/grid) schrumpfen Elemente ohne explizite Vorgaben auf Nullbreite, so dass Teile (z.B. Progress-Balken, Text) unsichtbar bleiben. In Release Notes wurde deshalb ausdrücklich festgehalten, dass für eingebettete Karten („nested layout compatibility“) fixiert wurde: „Added `:host { display: block; width: 100% }` to all child components“ <sup>1</sup>. Tatsächlich sehen wir in den Style-Dateien des Plugins, dass **jede Komponente** nun mit `:host { display: block; width: 100%; }` versehen ist.

Beispielsweise im **Card-Styles**:

```
:host {  
  display: block;  
  width: 100%;  
  /* ... weitere CSS-Variablen ... */  
}  
ha-card {  
  display: block;  
  width: 100%;  
  min-width: 0;  
  box-sizing: border-box;  
}
```

(Zeilen <sup>2</sup> im Stylesheet). Durch `min-width: 0;` wird zudem verhindert, dass das Ha-Card-Element in einem flexiblen Container überläuft.

Ähnlich wurden die Unterkomponenten (Timeline-Bar, Fakten-Block, Marker-Liste) definiert:

```
ytc-timeline-bar, ytc-facts-block, ytc-marker-list {  
  display: block;  
  width: 100%;  
}
```

Konkret sieht man in den Stylesheets z.B.: `:host { display: block; width: 100%; }` bei der Timeline-Bar <sup>3</sup>, im Facts-Block <sup>4</sup> und in der Marker-Liste <sup>5</sup>. Vor diesen Fixes waren die Komponenten inline oder auf „Auto“-Breite, wodurch sie in einem verschachtelten Layout (z.B. in `layout-card`-Grid oder `mod-card`-Flex) auf minimale Breite geschrumpft sind – das erklärt das abgeschnittene Rendering.

# Vergleich mit funktionierenden Karten

Andere Custom Cards definieren dieses Verhalten korrekt. So verwendet z.B. die **Weather Forecast Card** ebenfalls ein `:host { display:block; width:100% }`-Stil und setzt u.U. einen `ResizeObserver`, um sich an Containergrößen anzupassen. Ebenso setzt die **Button-Card** auf full-width-Stile (und liefert generell generische CSS-Regeln). Durch diese festen Breiten können sie in flexiblen oder grid-basierten Layouts problemlos skaliert werden. Die fehlenden Deklarationen waren also der Hauptunterschied.

## Lösungsvorschläge

- **CSS-Hinzugabe:** Wenn man die Karte bereits eingebunden hat, kann man mit [card-mod](#) o.Ä. dem Host-Element sofort `display:block; width:100%` verpassen. Beispiel:

```
type: custom:mod-card
card_mod:
  style: |
    year-timeline-card {
      display: block;
      width: 100%;
    }
card:
  type: custom:year-timeline-card
...
```

- **Plugin-Update:** Ab Version nach den Fixes (siehe [Release Notes] <sup>1</sup>) enthält das Plugin die nötigen Styles. Ein Update auf die aktuelle Version behebt das Problem dauerhaft.
- **Alternative: ResizeObserver:** Die Weather-Forecast-Card verwendet zusätzlich einen `ResizeObserver` (siehe deren Code <sup>6</sup>) um dynamisch auf Größenänderungen zu reagieren. Theoretisch könnte man einen Observer einbauen, um bei Container-Resize neu zu rendern. In diesem Fall reichte aber das obige CSS.

## Zusammenfassung der Fixes (Codebeispiele)

Die im Fix hinzugefügten CSS-Regeln lauten – wie in [Release Notes] <sup>1</sup> angegeben – z.B.:

```
/* In styles/styles.ts (MonthTimeline-Card) */
:host { display: block; width: 100%; }
```

Dadurch erhält die gesamte Karte und ihre Unterelemente volle Breite im umgebenden Layout. Die Release Notes dokumentieren:

*Fixed nested layout compatibility (layout-card, mod-card, grid containers) – Added `:host { display: block; width: 100% }` to all child components <sup>1</sup>.*

Diese Änderung sorgt dafür, dass in verschachtelten Containern nun **alle** Teile der Year-Timeline-Card korrekt angezeigt werden. Beispielsweise im Timeline-Bar-Stylesheet:

```
:host {  
  display: block;  
  width: 100%;  
}
```

(Quelle: Styles von year-timeline-card <sup>3</sup>). Ohne diese Vorgaben schrumpfte die Karte in flexiblen Layouts und zeigte nur einen Teilinhalt. Mit diesen Fixes ist das Problem behoben – die Karte kann nun in `layout-card`, `mod-card` oder `grid-layout` verwendet werden, ohne dass Inhalte verloren gehen.

**Quellen:** Offizielle [Release Notes] <sup>1</sup> und Quellcodes der Jahres-Timeline-Card mit den ergänzten Styles <sup>2</sup> <sup>3</sup> <sup>4</sup> <sup>5</sup>.

---

<sup>1</sup> RELEASE\_NOTES.md

[https://github.com/moerk-o/year-timeline-card/blob/5ed54e1e4b551f642266d4cea28fe2c38970111c/RELEASE\\_NOTES.md](https://github.com/moerk-o/year-timeline-card/blob/5ed54e1e4b551f642266d4cea28fe2c38970111c/RELEASE_NOTES.md)

<sup>2</sup> <sup>3</sup> <sup>4</sup> <sup>5</sup> styles.ts

<https://github.com/moerk-o/year-timeline-card/blob/5ed54e1e4b551f642266d4cea28fe2c38970111c/src/styles/styles.ts>

<sup>6</sup> weather-forecast-card.ts

<https://github.com/troinine/ha-weather-forecast-card/blob/d016345e354c79551ff5059247f705d8d48c7161/src/weather-forecast-card.ts>