

# Map UI System - Dokumentation

## Übersicht

Dieses Verzeichnis enthält die gemeinsamen UI-Komponenten für alle Leaflet-Karten im Projekt. Das System basiert auf dem **CO.RA.PAN responsive popup design** und gewährleistet konsistentes Verhalten auf Desktop und Mobile.

## Dateien

### map\_ui.js - Zentrale Hilfsfunktionen

**Verfügbar als:** `window.MapUI` (globaler Namespace)

#### Funktionen

`isMobileViewport(): boolean`

Erkennt Mobile-Viewport (max-width: 599px).

#### Verwendung:

```
const isMobile = window.MapUI.isMobileViewport();
if (isMobile) {
  // Mobile-spezifische Logik
}
```

`popupOptions(className: string): object`

Gibt standardisierte Popup-Optionen zurück.

**Parameter:** - `className` (optional): CSS-Klassenname für das Popup (Standard: 'corapan-popup')

#### Rückgabe:

```
{
  className: 'corapan-popup',
  closeButton: true,
  autoClose: true,
  closeOnClick: false,
  autoPan: true,
  keepInView: true,
  maxWidth: mobile ? 280 : 320,
  minWidth: mobile ? 180 : 200,
  autoPanPaddingTopLeft: mobile ? [20, 80] : [50, 100],
  autoPanPaddingBottomRight: mobile ? [20, 20] : [50, 50]
}
```

```
bindClickPopup(map, marker, html, className)
```

Bindet ein Click-Popup an einen Marker mit automatischem Pan.

**Parameter:** - `map`: Leaflet Map-Instanz - `marker`: Marker-Objekt - `html`: HTML-Inhalt des Popups - `className`: CSS-Klassenname (optional, Standard: 'corapan-popup')

**Verwendung:**

```
const marker = L.circleMarker([lat, lng], {...}).addTo(map);
const popupHtml = '<div class="popup-sprachenkarte">...</div>';
window.MapUI.bindClickPopup(map, marker, popupHtml, 'corapan-popup');
```

**Verhalten:** - Click auf Marker → Pan zur Position → Öffne Popup - Event-Propagation wird gestoppt (kein Map-Click-Event)

```
enablePopupCloseUX(map)
```

Aktiviert globales Schließverhalten für Popups.

**Features:** - Click auf freie Kartenfläche schließt Popup - Esc-Taste schließt Popup (Desktop)

**Verwendung:**

```
const map = L.map('mapid').setView([lat, lng], zoom);
window.MapUI.enablePopupCloseUX(map);
```

```
enableResponsiveInvalidation(map)
```

Aktiviert robuste Resize/Orientation-Behandlung.

**Verwendung:**

```
const map = L.map('mapid').setView([lat, lng], zoom);
window.MapUI.enableResponsiveInvalidation(map);
```

**Verhalten:** - Bei `resize` → `map.invalidateSize()` nach 200ms - Bei `orientationchange` → `map.invalidateSize()` nach 200ms

`isMapInitialized(container): boolean`

Prüft, ob eine Karte bereits initialisiert wurde.

**Verwendung:**

```
const container = document.querySelector('[data-map="herkunft"]');
if (window.MapUI.isMapInitialized(container)) {
    return; // Bereits initialisiert, nichts tun
}
```

`markMapInitialized(container)`

Markiert einen Container als initialisiert.

**Verwendung:**

```
window.MapUI.markMapInitialized(container);
```

## CSS-Klassen

`.corapan-popup` - Standard-Popup-Klasse

**Desktop:** - `max-width: 340px` - `border-radius: 8px` - `box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15)`

**Mobile (max-width: 599px):** - `max-width: 90vw` - `max-width: calc(100vw - 48px)` (Content Wrapper) - `overflow-wrap: anywhere` - `max-height: 55vh` (mit Scrolling)

`.popup-sprachenkarte` - Content-Container

Standard-Styles für Popup-Inhalte:

```
<div class="popup-sprachenkarte">
  <div class="popup-title">Titel</div>
  <div class="popup-familie">Untertitel</div>
  <div class="popup-line">
    <span class="popup-label">Label:</span>
    <span class="popup-value">Wert</span>
  </div>
</div>
```

**Klassen:** - `.popup-title` - Haupttitel (fett, blau) - `.popup-familie` / `.popup-hauptstadt` - Untertitel (kursiv, grau) - `.popup-herkunft` - Zusatztext - `.popup-line` - Zeile mit Label/Value - `.popup-label` - Label (klein, grau) - `.popup-value` - Wert (fett, schwarz)

## Verwendungsbeispiel

### Komplette Karten-Initialisierung

```

function initMyMap() {
  const container = document.querySelector('[data-map="mymap"]');

  if (!container) return;

  // Doppel-Init verhindern
  if (window.MapUI && window.MapUI.isMapInitialized(container)) {
    return;
  }

  const mapEl = container.querySelector('#mapid');
  if (!mapEl) return;

  // Map erstellen
  const map = L.map(mapEl).setView([45, 15], 3);

  // Tiles hinzufügen
  L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    attribution: 'Map data © <a
href="https://openstreetmap.org">OpenStreetMap</a>'
  }).addTo(map);

  // UX-Funktionen aktivieren
  if (window.MapUI) {
    window.MapUI.enablePopupCloseUX(map);
    window.MapUI.enableResponsiveInvalidation(map);
  }

  // Marker hinzufügen
  const marker = L.circleMarker([48, 11], {
    color: '#1976d2',
    radius: 8
  }).addTo(map);

  const popupHtml = `
    <div class="popup-sprachenkarte">
      <div class="popup-title">München</div>
      <div class="popup-line">
        <span class="popup-label">Land:</span>
        <span class="popup-value">Deutschland</span>
      </div>
    </div>`;

  // Popup binden
  if (window.MapUI) {
    window.MapUI.bindClickPopup(map, marker, popupHtml, 'corapan-
popup');
  } else {

```

```

    marker.bindPopup(popupHtml); // Fallback } // Als initialisiert
markieren if (window.MapUI) {
window.MapUI.markMapInitialized(container); } // Init bei
DOMContentLoadedif (document.readyState === 'loading') {
document.addEventListener('DOMContentLoaded', initMyMap);} else {
initMyMap();} // Material for MkDocs Instant Navigation Supportif (typeof
document$ !== 'undefined') { document$.subscribe(() => initMyMap());}

```

## Z-Index-Hierarchie

### Leaflet-Layer (innerhalb der Karte)

```

200 - Leaflet Tile Pane (Kartenkacheln)
400 - Leaflet Control Container (Zoom, Attribution)
600 - Leaflet Marker Pane (Marker)
800 - Leaflet Popup Pane (Popup-Container)
850 - Leaflet Popup (Popup selbst)

```

### Page-Level Z-Index (Material for MkDocs)

```

1000 - Header (.md-header)
1200 - Fullscreen Map Container
2000 - Navigation Drawer + Overlay (.md-sidebar, .md-nav, .md-overlay)

```

⚠ **KRITISCH:** 1. Material for MkDocs Navigation-Drawer muss `z-index: 2000` haben 2. Fullscreen-Container dürfen maximal `z-index: 1200` haben 3. Karten-Container dürfen **keine** Stacking-Context-Fallen haben: - Kein `transform` (außer auf inneren Controls) - Kein `filter` - Kein `perspective`

**Wichtig:** `overflow: visible` auf Map-Containern, damit Popups nicht abgeschnitten werden!

## Dark Mode Support

Alle Popup-Styles sind Dark-Mode-kompatibel:

```

body[data-md-color-scheme="slate"] .corapan-popup {
  /* Dark Mode Styles */
}

```

## Checkliste für neue Karten

- [] Container mit `data-map="unique-id"` kennzeichnen
- [] Init-Funktion schreiben
- [] `window.MapUI.bindClickPopup()` für alle Marker nutzen
- [] `window.MapUI.enablePopupCloseUX(map)` aufrufen
- [] `window.MapUI.enableResponsiveInvalidation(map)` aufrufen
- [] Doppel-Init-Schutz implementieren
- [] Material Instant Navigation Support hinzufügen
- [] JS-Datei in `mkdocs.yml` eintragen
- [] Testen: Click, Esc, Map-Click, Resize, Mobile

## Debugging

### MapUI nicht verfügbar?

```
console.log(window.MapUI); // Sollte object sein
```

### Popup öffnet nicht bei Click?

```
marker.on('click', () => console.log('Click!'));
```

### Popup wird abgeschnitten?

```
/* Container muss overflow: visible haben */
#map-container {
  overflow: visible !important;
}
```

## Migration von alten Hover-Tooltips

### Alt (Hover):

```
marker.bindTooltip(html);  
marker.on('mouseover', function() { this.openTooltip(); });  
marker.on('mouseout', function() { this.closeTooltip(); });
```

#### **Neu (Click):**

```
window.MapUI.bindClickPopup(map, marker, html, 'corapan-popup');
```

---

**Erstellt:** Dezember 2025

**Projekt:** Spanische Linguistik @ School

**Referenz:** CO.RA.PAN responsive popup design