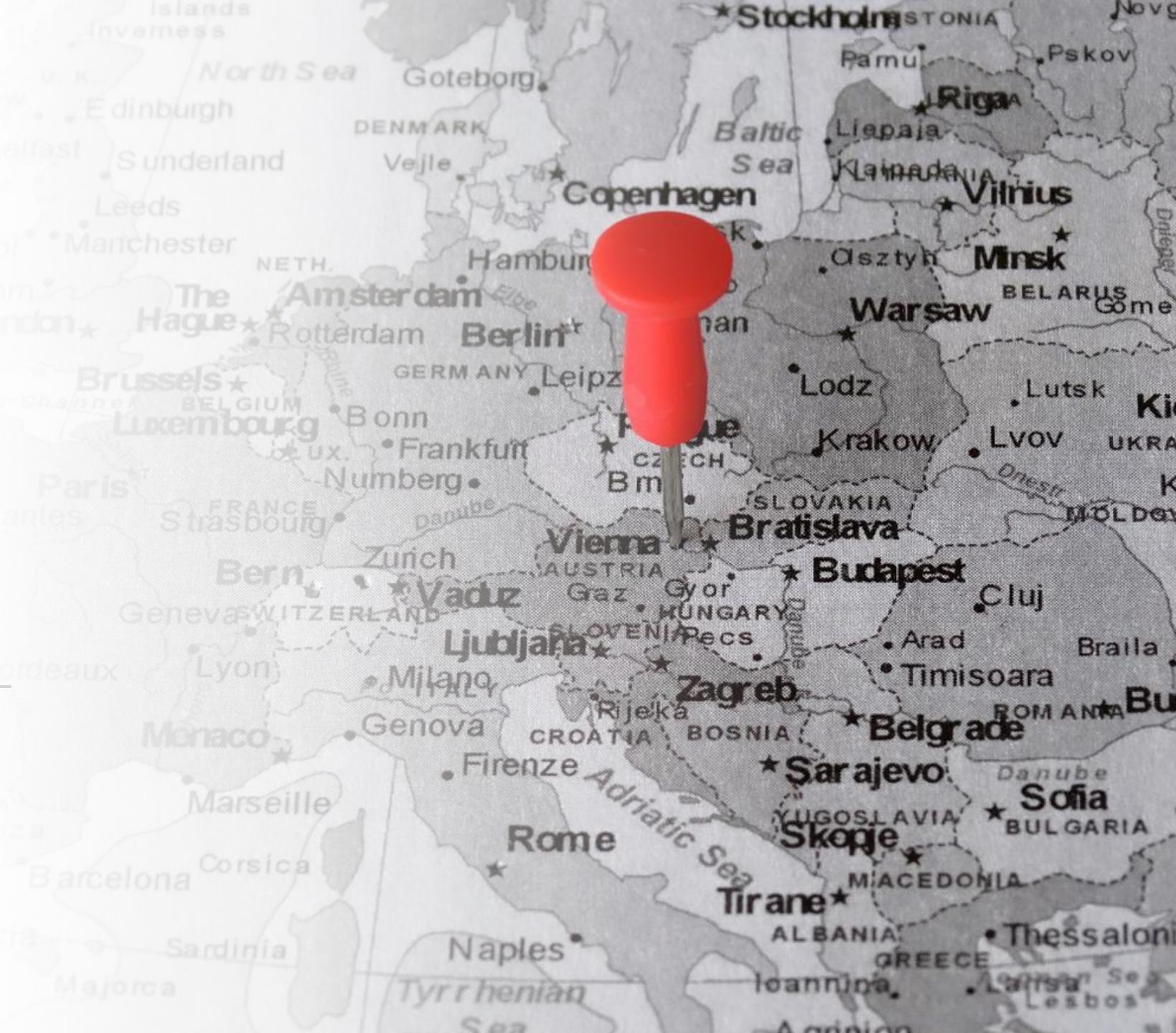


Projektpräsentation

VU LV: 716409 VU Geoinformatik: Web mapping
Kursleitung: B.Sc.. Klaus Förster, Mag. Bernhard Öggl
Sommersemester 2022
30. Juni 2022
Referenten:
Maria Heinrich
Paula Spannring



Inhalt

1. Themenfindung & Probleme
2. Startseite (index.html)
3. Basisseiten (index.html, main.js, main.css)
4. Winter-Seite
5. Sommerseite
6. Fazit

1. THEMENFINDUNG

Themenfindung



Quelle: https://pixabay.com/de/?utm_source=link-attribution&utm_medium=referral&utm_campaign=image&utm_content=712490“, abgerufen am 23.06.22

Lawinendarstellung

- Zusätzlich xml-Datei notwendig → schwierig einlesbar

Freizeitangebote des Landes Salzburg

- Daten als .json Speichern → mit Aufruf „no-core“
- Nur Header werden angezeigt, kein Inhalt
 - Kein open-source-Zugriff

Freizeitangebote des der Stadt Wien

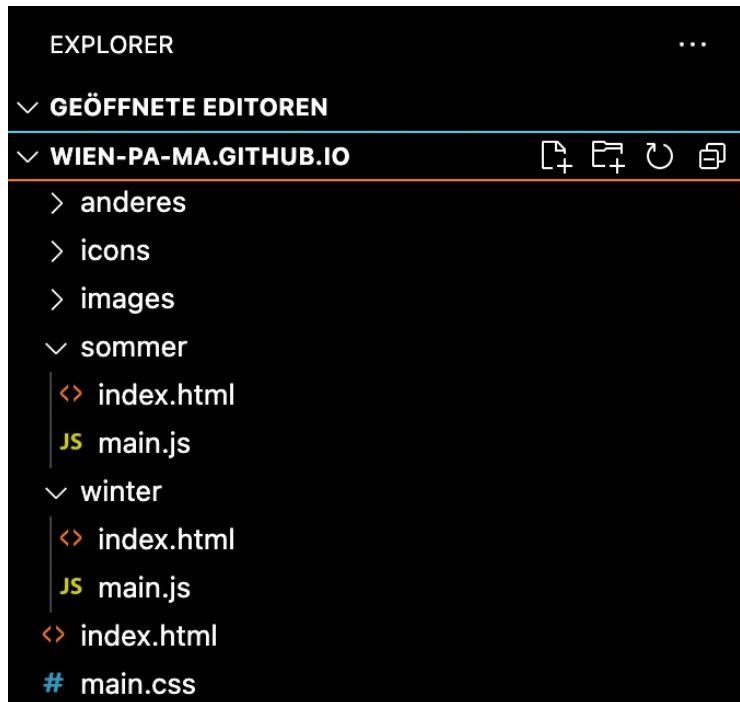
- Sommerangebote
- Winterangebote



Quelle: <https://pixabay.com/de/photos/atlas-landkarte-wien-pin-642023/>, abgerufen am 18.06.22

AUFBAU, STARTSEITE & STYLESHEETS

Aufbau



- Startseite → index.html
- Sommer → index.html, main.js
- Winter → index.html, main.js
- Übergreifende Dateien & Ordner
- → main.css, images, icons

Quelle: Screenshot / eigene Darstellung.

Startseite → index.html



Github
[wien-
pa-ma](#)

Skyline von Wien in der Nacht [Julius_Silver](#)

Sommer- und Winteraktivitäten in Wien

Starttext

Willkommen in Wien. Auf folgenden Seiten können verschiedene Freizeitangebote für ein aktives Leben in und um Wien erkundet werden. Dafür kann je nach Jahreszeit in Winter oder Sommer unterschieden werden. Viel Spaß beim erkunden!

Links zu Sommer- und Winteraktivitäten

- [Winteraktivitäten in und um Wien](#)
- [Sommeraktivitäten in und um Wien](#)

```
22 23 <body>
24 25   <main>
26 27     <header>
28 29       <div>...
30 31       </div><br>
32 33       <nav>...
34 35       </nav>
36 37     </header>
38 39
39 40   <article>
40 41     <h1>Sommer- und Winter
41 42     <h2>Starttext</h2>
42 43     <p>...
43 44     </p>
44 45     <h2>Links zu Sommer- u
45 46     <nav>...
46 47     </nav>
47 48     <figure>...
48 49     </figure>
49 50   </article>
50 51
51 52   <footer>
52 53     </footer>
53 54   </main>
54 55 </body>
55 56
```

Stylesheets & main.css

The diagram illustrates the inheritance of CSS styles from a single `main.css` file into two separate HTML files: `index.html` and `index.html sommer`.

File Structure:

- index.html**: Contains a `<link rel="stylesheet" href="main.css">` in the `<head>` section.
- index.html sommer**: Contains a `<link rel="stylesheet" href="../main.css">` in the `<head>` section.
- main.css**: A separate file containing global styles for the entire application.

Style Inheritance:

- Font Awesome:** Both `index.html` and `index.html sommer` include a `<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css"` to import the Font Awesome library.
- Font Styles:** The `main.css` file imports Google Fonts for various font families: Montserrat, Open Sans, and Roboto.
- Body Styling:** The `body` element is styled with a background color of #f0f0f0 and a font family of sans-serif.
- Header Styling:** The `header` element is styled with a background color of #e0e0e0 and a font weight of bold.
- Section Styling:** The `h1`, `h2`, and `h3` elements are styled with a font weight of bold and a font size of 1.2em.
- Text Styling:** The `p` elements within the `main` section are styled with a font weight of bold and a font size of 1em.
- Image Styling:** The `img` element within the `main` section is styled with a width of 100% and a height of auto.
- Footer Styling:** The `footer` element is styled with a background color of #e0e0e0 and a font weight of bold.
- Shadow Styling:** The `.shadow` class is styled with a box shadow of 0 0 10px 0 #000.

MAIN.JS & INDEX.HTML
→ SOMMER UND WINTER

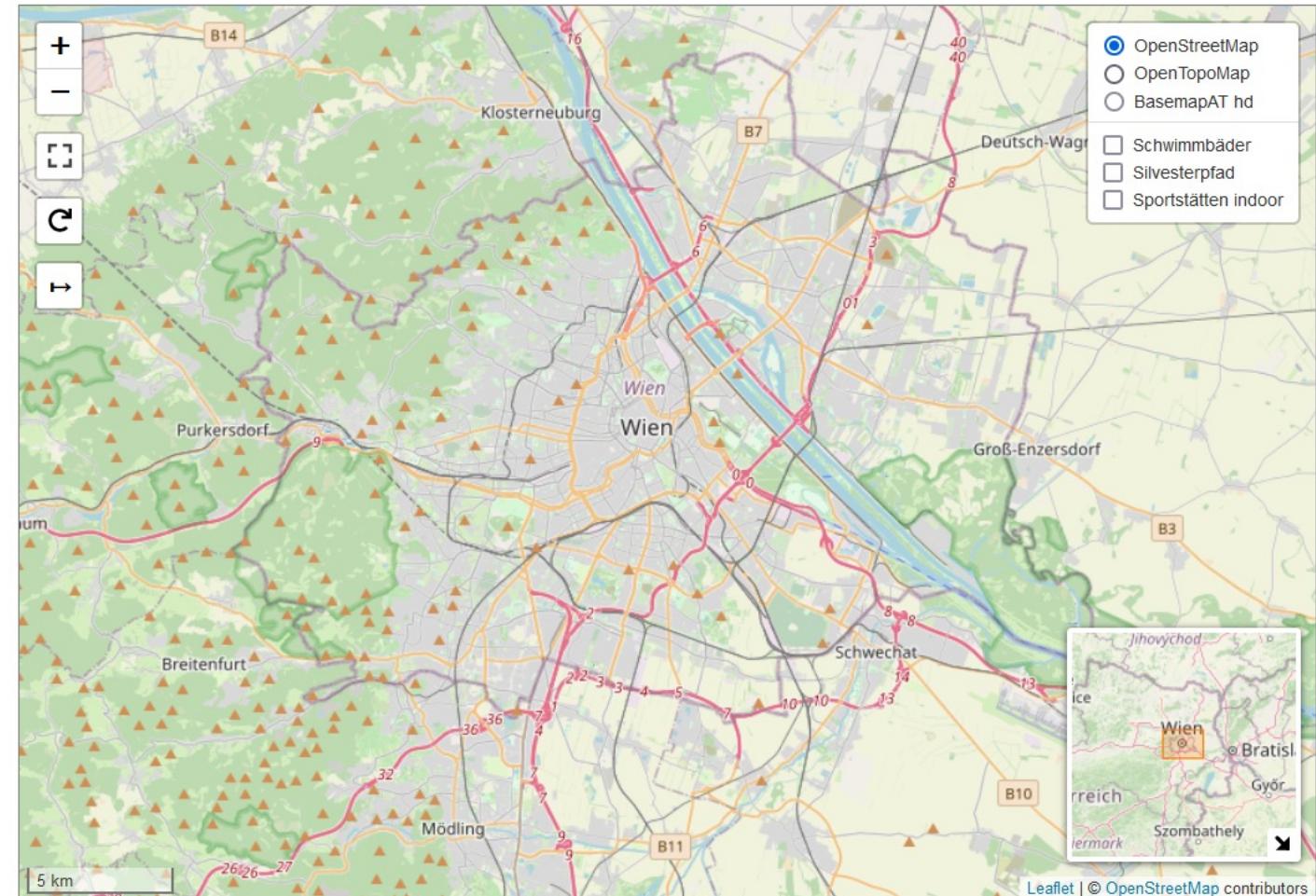
Sommer- und Winterseiten → index.html

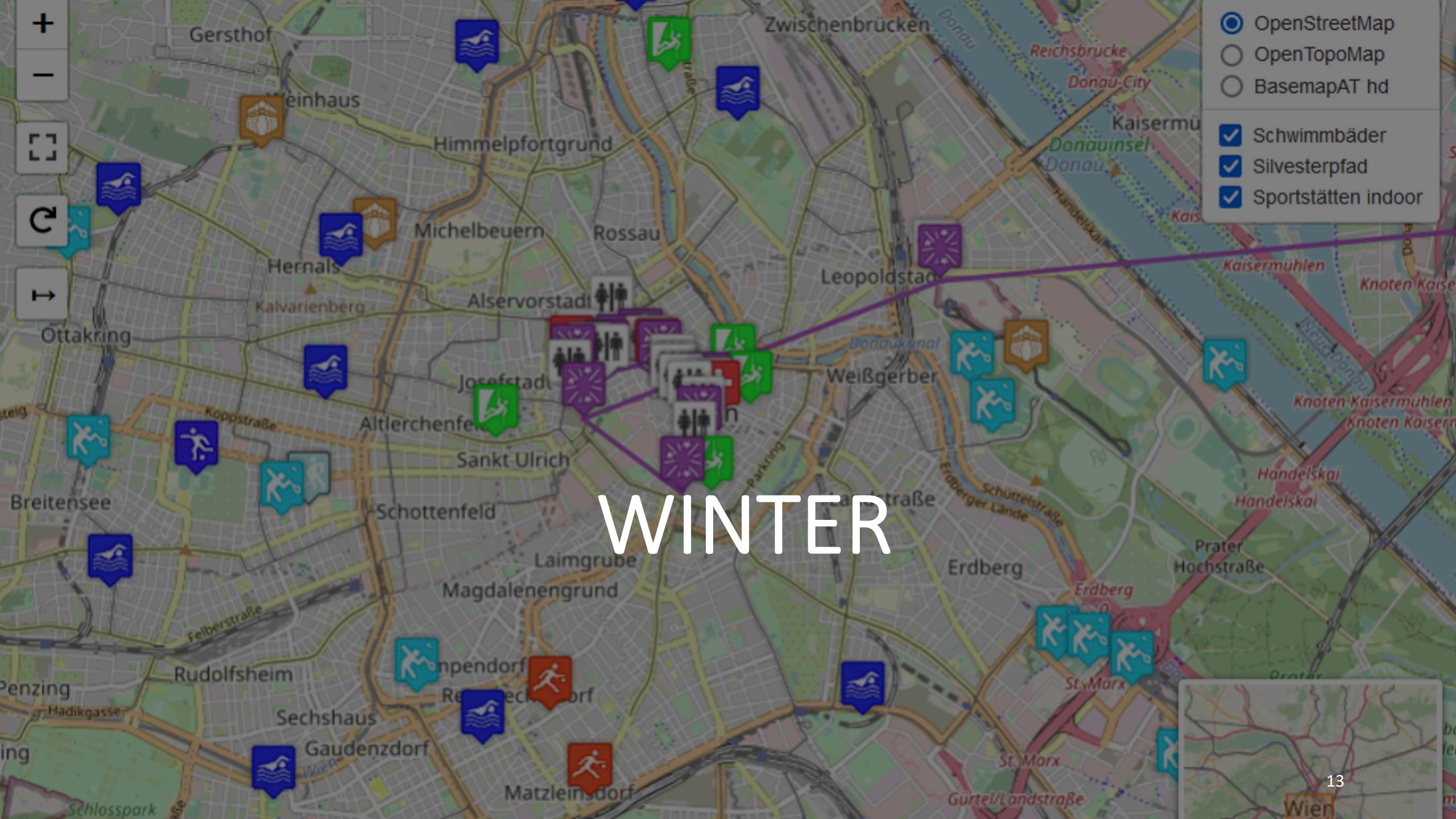
- Head – Bereich:
 - Shortcut Icon (Wiener Wappen),
 - Titel der Seiten,
 - Verlinkung zu Javascript und CSS-Stylesheet,
 - Implementieren der Plugins
- Body – Bereich:
 - Header
 - Article: Kartenbereich
 - Footer: verwendete Daten (mit Link zu Quelle)



Sommer- und Winterseiten → main.js

- Karte implementieren mit Plugins:
 - Maßstab
 - Fullscreen - Button
 - Minimap
 - Reset – Zoom – Button
 - Button zum Messen der Entfernung verschiedener Punkte
 - Layer- Overlay Control





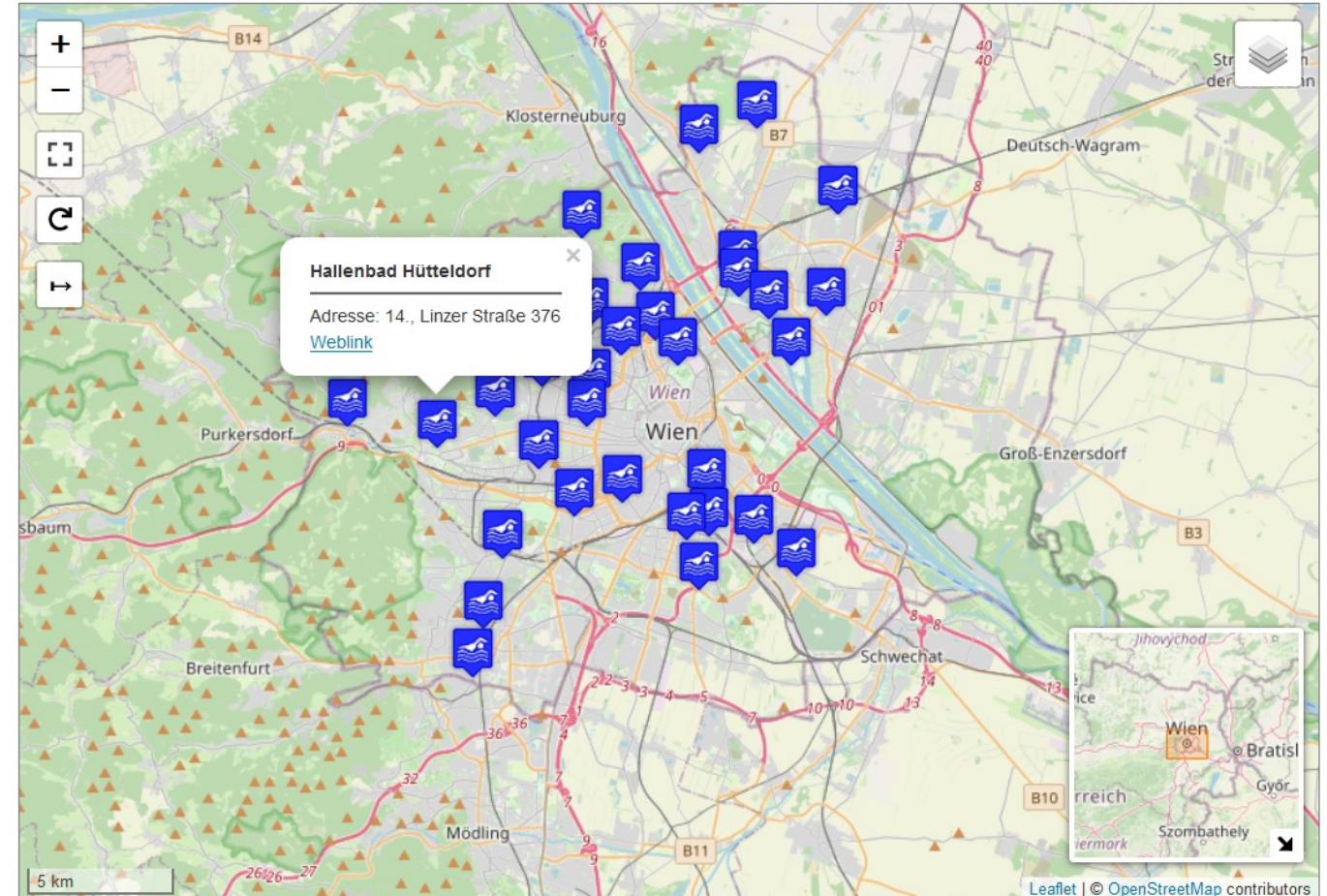
- OpenStreetMap
 - OpenTopoMap
 - BasemapAT hd
- Schwimmbäder
 - Silvesterpfad
 - Sportstätten indoor

WINTER



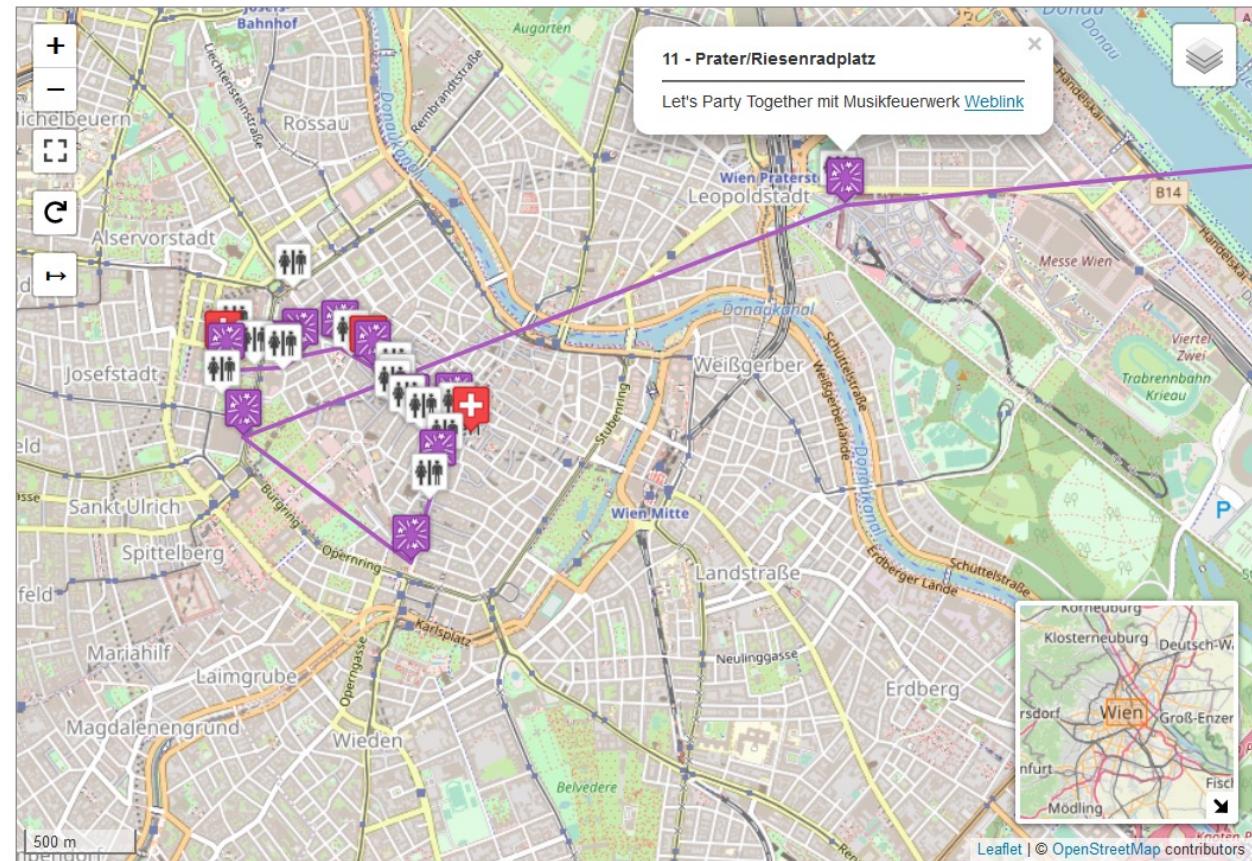
Darstellen der Schwimmbäder

- GeoJSON – Datei: Wiens Schwimmbäder
- mit Attribut, ob geöffnet und nicht ausgelastet
→ nur diese anzeigen (mit if-Abfrage selektiert)
- Passendes Icon für Marker und Popup



Darstellen des Silvesterpfads

- GeoJSON – Datei: Stände des Silvesterpfads (Punkte)
- Mit Attribut für Typ des Stands: Toilette, Erste Hilfe, Station (nummeriert)
→ Passendes Icon für Marker
- Stationen sollen in der richtigen Reihenfolge verbunden werden:



Silvesterpfad: Polyline zwischen Stationen

- Sortieren der features nach Bezeichnung (→ String)
- Sortieren der features nach der Nummer (→ Integer)
- Speichern der Koordinaten (in richtiger Reihenfolge in Array)
- Polylinie zeichnen, an overlay anhängen

```
geojson.features.sort(function (a, b) {
    return
    parseInt(a.properties.BEZEICHNUNG.split("-")[0]) >
    parseInt(b.properties.BEZEICHNUNG.split("-")[0]);
});

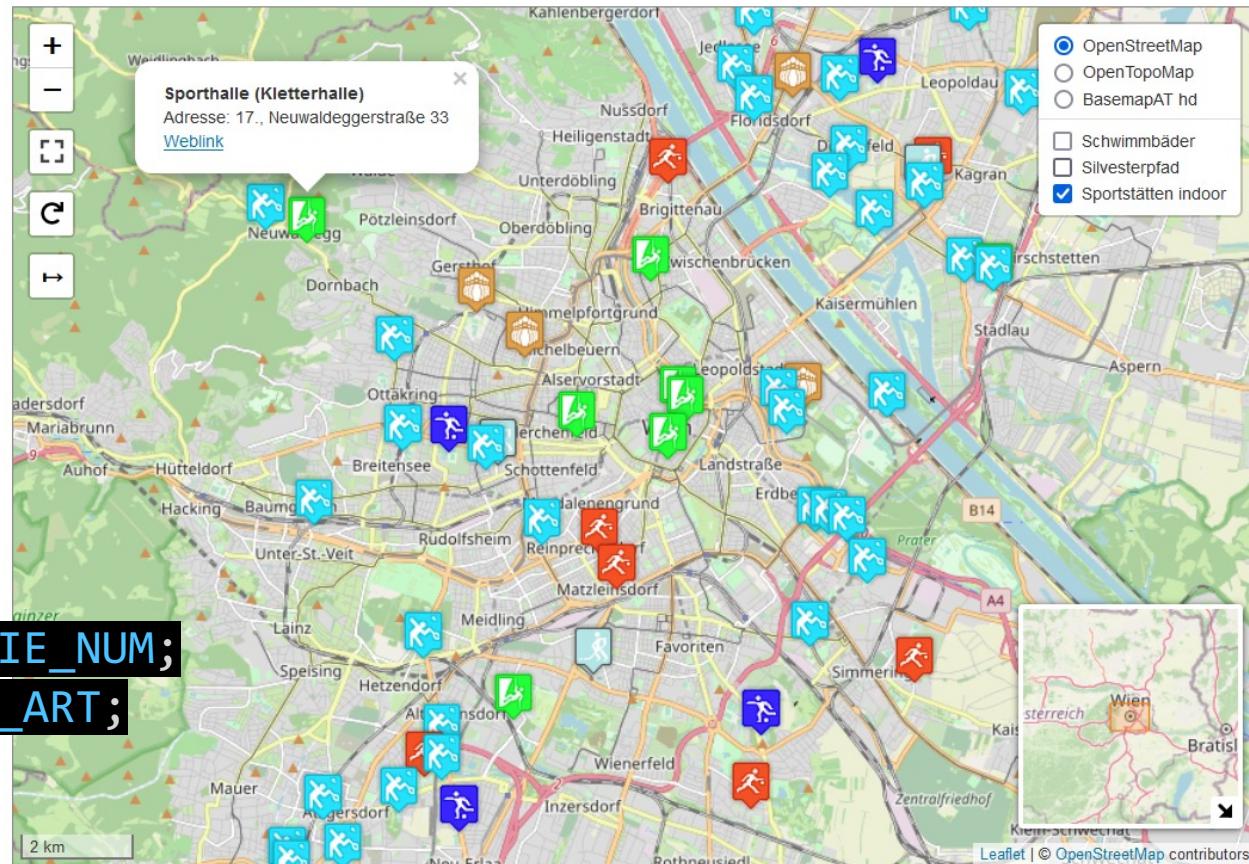
let stations = []
for (i = 0; i < geojson.totalFeatures; i += 1) {
    if (geojson.features[i].properties.TYP == 1) {
        stations.push([geojson.features[i].geometry
            .coordinates[1],
        geojson.features[i].geometry.coordinates[0]]))

        let polyline = L.polyline(stations, {
            color: '#ad59c2'}).addTo(overlays.silvester);
```

Darstellen von indoor Sportstätten

- GeoJSON – Datei: Sportstätten
- Mit Attribut (kategorie), ob indoor oder outdoor
→ nur indoor anzeigen (mit if-Abfrage selektiert)
- Passendes Icon für Sportart (in Attribut enthalten)
- Beispiel Klettern:

```
let kategorie = geoJsonPoint.properties.KATEGORIE_NUM;  
let art = geoJsonPoint.properties.SPORTSTAETTEN_ART;  
if (kategorie == 2 & art.includes('Kletter')) {  
    return L.marker(latlng, {  
        icon: L.icon({  
            iconUrl: "../icons/climbing.png",  
            . . .
```





Sommerseite – Feature.Point



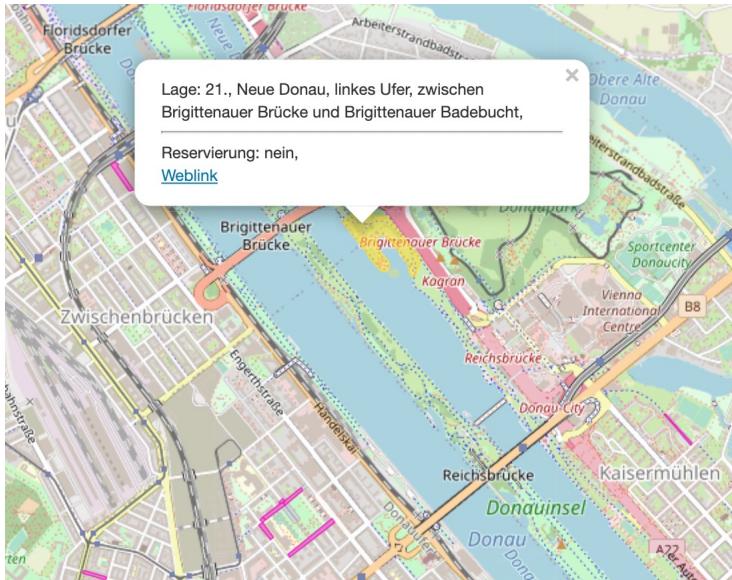
```
Waldspielplätze – Geometrie des 1. Eintrags: main.js:229
▼ {type: 'Point', coordinates: Array(2)} ⓘ
  ► coordinates: (2) [16.2441393759616, 48.15125127099312]
    type: "Point"
  ► [[Prototype]]: Object
```

```
// Waldspielplätze OKEEE
-----
1 → async function loadWaldspiel(url) {
  let response = await fetch(url);
  let geojson = await response.json();
  console.log("Waldspielplätze: ", geojson);
  console.log("Waldspielplätze – Geometrie des 1. Eintrags: ", geojson.features[0].geometry);

  L.geoJSON(geojson, {
    pointToLayer: function (geoJsonPoint, latlng) {
      let popup = `...`;
      ;
      return L.marker(latlng, {
        icon: L.icon({
          iconUrl: `../icons/wspielplatz.png`,
          iconAnchor: [16, 37],
          popupAnchor: [0, -37]
        })
      }).bindPopup(popup);
    }
  }).addTo(overlay.waldspielplaetze);
}

loadWaldspiel("https://data.wien.gv.at/daten/geo?service=WFS&request=GetFeature&version=1.1.0&typeName=ogdwien:WALDSPIELPLAETZE&srsName=EPSG:4326&outputFormat=json");
```

Sommerseite – Feature.Polygon



Grillzonen Geometrie des 1. Eintrags: [main.js:109](#)
▼ {type: 'Polygon', coordinates: Array(1)} [i](#)
 ► coordinates: [Array(444)]
 type: "Polygon"
 ► [[Prototype]]: Object

```
1  async function loadGrill(url) {  
2    let response = await fetch(url);  
3    let geojson = await response.json();  
4    console.log("Grillzonen: ", geojson);  
  
5    L.geoJSON(geojson, {  
6      style: function (feature) {  
7        return {  
8          color: "#FFDC00",  
9          weight: 1,  
10         opacity: 0.9,  
11         fillOpacity: 0.5,  
12       }  
13     }  
14   }).bindPopup(function (layer) {  
15     return `  
16       Lage: ${layer.feature.properties.LAGE},  
17       Reservierung: ${layer.feature.properties.  
18         RESERVIERUNG},  
19       <a href="${layer.feature.properties.WEBLINK1}"  
20         target="_blank" >Weblink</a>  
21     `;  
22   })  
23   .addTo(overlay.grillzonen);  
24 }  
25  
26 loadGrill("https://data.wien.gv.at/daten/geo?  
27 service=WFS&request=GetFeature&version=1.1.0&  
28 typeName=ogdwien:GRILLZONE0GD&srsName=EPSG:4326&  
29 outputFormat=json");
```

MarkerClusterGroup



```
<!-- Leaflet MarkerCluster css.default, css & js plugin-->
<link rel="stylesheet"
  href="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.5.3/MarkerCluster.Default.css"
  integrity="sha512-6ZCLMiYwTeli2rVh3XAPxy3YoR5fVxGdH/pz+KMCzRY2M65Emgkw00Yqmhh8qLGeYQ3LbVGdmOX9KujSKr0TA=="
  crossorigin="anonymous" referrerPolicy="no-referrer" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.5.3/MarkerCluster.css"
  integrity="sha512-m077VzAakzdnWdofI/1M1ksNv89uEaibR0ANsNneSTM/
```

```
// Spielplätze OKEEE-----
async function loadSpiel(url) {
  let response = await fetch(url);
  let geojson = await response.json();
  console.log("Spielplätze: ", geojson);

  let cspielplaetze = L.markerClusterGroup({
    disableClusteringAtZoom: 16
  });
  cspielplaetze.addTo(overlay.spielplaetze);

  L.geoJSON(geojson, {
    pointToLayer: function (geoJsonPoint, latlng) {
      let popup = `...`;
      return L.marker(latlng, {
        icon: L.icon({
          ...,
        })
      }).bindPopup(popup);
    }
  }).addTo(cspielplaetze);

  loadSpiel("https://data.wien.gv.at/daten/geo?service=WFS&request=GetFeature&version=1.1.0&typeName=ogdwien:SPIELPLATZPUNKT0GD&srsName=EPSG:4326&outputFormat=json");
}
```

Literatur

Plugins

- Leaflet: <https://leafletjs.com/examples/quick-start/>
- Leaflet – providers: <https://cdnjs.com/libraries/leaflet-providers>
- Fullscreen: <https://cdnjs.com/libraries/leaflet.fullscreen>
- Minimap: <https://cdnjs.com/libraries/leaflet-minimap>
- Reset view: <https://github.com/drustack/Leaflet.ResetView>
- Polyline measure: <https://github.com/ppete2/Leaflet.PolylineMeasure>
- MarkerClusterGroup: <https://github.com/Leaflet/Leaflet.markercluster>

Font awesome Stylesheet

- Stylesheets: <https://fonts.googleapis.com/>
- Daten
 - Schwimmbäder: https://www.data.gv.at/katalog/dataset/stadt-wien_schwimmbderstandortewien/resource/29041605-d50d-4696-8827-bb496ac9ad75
 - Silvesterpfad: <https://www.data.gv.at/katalog/dataset/f5d7a560-0599-4ee6-b9c1-5efec380b912>
 - Sportstätten: <https://www.data.gv.at/katalog/dataset/71084c02-973d-4544-b804-7ed82bd53027>
 - Badestellen: https://www.data.gv.at/katalog/dataset/stadt-wien_badestellenstandortewien/resource/53195ad6-3c6d-47f9-b56f-b42e4e8af83c
 - Waldspielplätze: <https://www.data.gv.at/katalog/dataset/waldspielplatze-wien/resource/e942f7d6-16c9-4fd4-9bbe-bd547f516160>
 - Spielplätze: <https://www.data.gv.at/katalog/dataset/spielplatze-standorte-wien/resource/6f27a91a-bb1e-44f4-a683-98cb80f63379>
 - Grillzonen: <https://www.data.gv.at/katalog/dataset/grillzonen-der-stadt-wien/resource/2341ba7e-15cf-41f9-abf4-e76d6c317eb0>
 - Parkanlagen: <https://www.data.gv.at/katalog/dataset/parkanlagen-standorte-wien/resource/e2aecfe2-9862-4178-a43c-8c994d676782>
 - Fußgängerzonen: https://www.data.gv.at/katalog/dataset/stadt-wien_fugngerzonewien/resource/8972d745-60cc-47c5-9d42-a28a995de177

Icons & Images

- Icons: <https://mapicons.mapsmarker.com/>
- Colors: <https://clrs.cc/>
- Images: <https://pixabay.com/de/>

Sonstige

- GitHub webmapping: <https://github.com/webmapping>

Projektpräsentation

VU LV: 716409 VU Geoinformatik: Web mapping

Kursleitung: B.Sc.. Klaus Förster, Mag. Bernhard Öttl

Sommersemester 2022

30. Juni 2022

