

Methodiken

| oder auch

Wie erstelle ich eine Karte im Browser?

Vorab

Wer bin ich und warum stehe ich hier?

Was machen wir heute?

Wir probieren gemeinsam eine Karte in unserem Browser anzuzeigen.

Dafür verwenden wir Web-Technologien (HTML, CSS, JavaScript) und existierende Bibliotheken für Karten.

Fragen:

- Wie vertraut seid ihr mit Web-Technologien?
- Kennt schon jemand Bibliotheken für Karten?

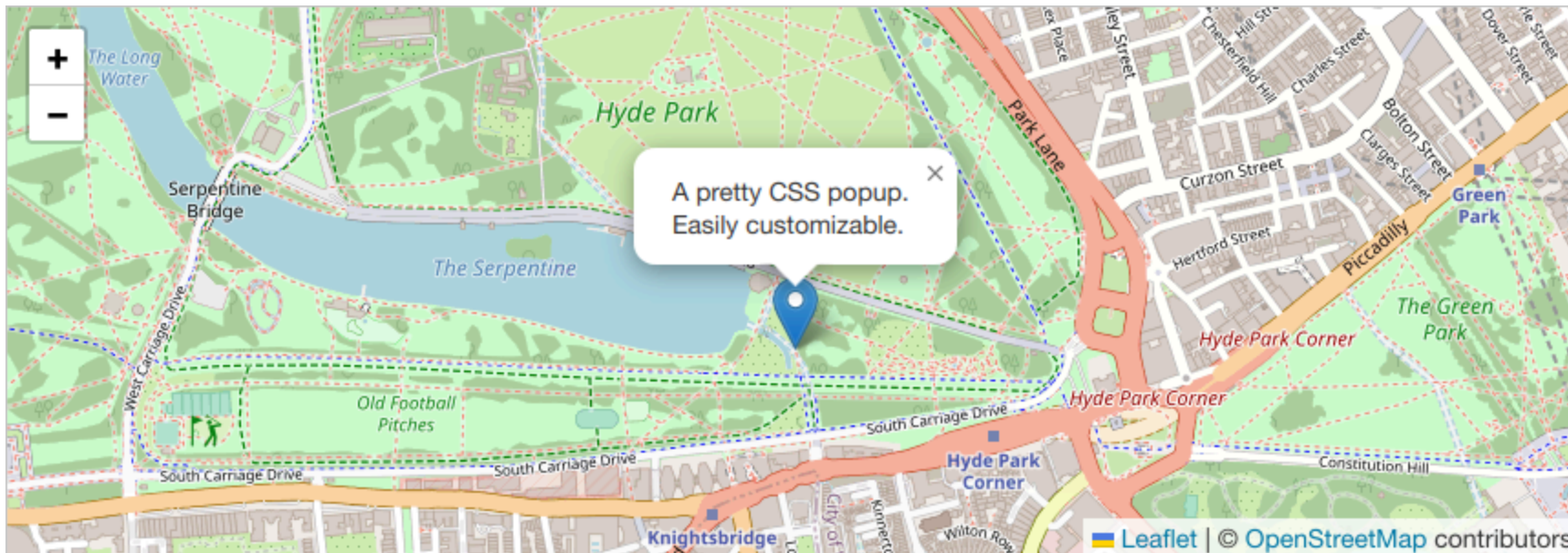
Bibliothek 1: Leaflet

Leaflet ist die führende Open-Source-JavaScript-Bibliothek für mobilfreundliche interaktive Karten. [...]

Bei der Entwicklung von Leaflet standen Einfachheit, Leistung und Benutzerfreundlichkeit im Vordergrund. [...] hat eine schöne, leicht zu bedienende und gut dokumentierte API und einen einfachen, lesbaren Quellcode, an dem man gerne mitarbeitet.

Klingt gut oder?

```
let map = L.map("map").setView([51.505, -0.09], 13);  
  
L.tileLayer("https://tile.openstreetmap.org/{z}/{x}/{y}.png", {  
  attribution: "OpenStreetMap",  
}).addTo(map);  
  
L.marker([51.5, -0.09]).addTo(map).bindPopup("A pretty CSS popup.<br> Easily customizable.").openPopup();
```



Code verstehen #1

```
let map = L.map("map").setView([51.505, -0.09], 13);
```

Code verstehen #1

```
let map = L.map("map").setView([51.505, -0.09], 13);
```

`let map` initialisiert unsere Kartenvariable

`L` ist das Leaflet-Objekt

`.map(id)` sucht ein HTML-Element mit der gegebenen ID für die Karte

`.setView([lat, long], zoom)` setzt die Ausgangsperspektive für die Karte

Code verstehen #2

```
L.tileLayer("https://tile.openstreetmap.org/{z}/{x}/{y}.png", {  
  attribution: "OpenStreetMap",  
}).addTo(map);
```


Code verstehen #2

```
L.tileLayer("https://tile.openstreetmap.org/{z}/{x}/{y}.png", {  
  attribution: "OpenStreetMap",  
}).addTo(map);
```

`.tileLayer(url, options)` erstellt eine Ebene mit Kartendaten von der gegebenen URL

`.addTo(map)` fügt die erstellte Ebene zu einer Karte hinzu

Code verstehen #3

```
L.marker([51.5, -0.09]).addTo(map).bindPopup("A pretty CSS popup.<br> Easily customizable.").openPopup();
```

Code verstehen #3

```
L.marker([51.5, -0.09]).addTo(map).bindPopup("A pretty CSS popup.<br> Easily customizable.").openPopup();
```

`.marker([lat, long])` erstellt einen Marker an den gegebenen Koordinaten

`.bindPopup(text)` erstellt ein Text-Popup für den vorher erstellten Marker

`.openPopup()` öffnet das erstellte Popup

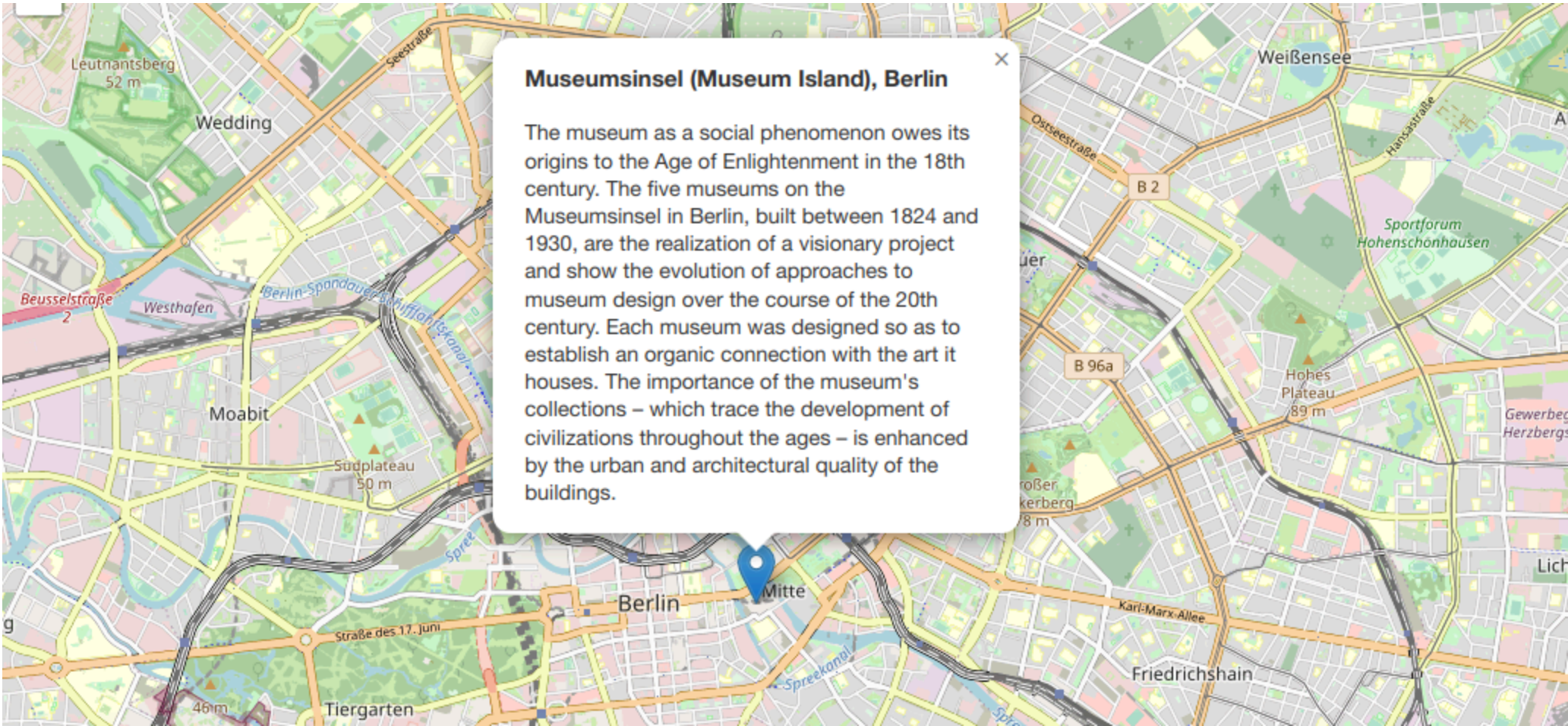
Leaflet einbinden

```
<html>
  <head>
    <!-- Leaflet CSS -->
    <!-- Leaflet JavaScript -->
    <script src="unser_script.js" defer></script>
    <style>
      #map {
        height: 50vh;
      }
    </style>
  </head>
  <body>
    <div id="map"></div>
  </body>
</html>
```

Praktische Übung

Erstellt ein Leaflet-Projekt, bei dem die Karte zu sehen ist, und setzt den Startpunkt und einen Marker auf eine Cultural Heritage Site eurer Wahl, mit einer kleinen Information in dem Popup.

| <https://whc.unesco.org/en/list/> > <https://leafletjs.com/>



Erstes Fazit zu Leaflet?

Viele Studierende entscheiden sich bei ihren Kartenprojekten für Leaflet, um Daten schnell und unkompliziert zu visualisieren.

Bibliothek 2: Maplibre GL JS

Open-Source TypeScript-Bibliothek zur Veröffentlichung von Karten auf Ihrer Website. Hohe Leistung durch GPU-beschleunigtes Rendering von Vektorkacheln.

Quelle: maplibre.org

MapLibre GL JS ist eine Open-Source-Abspaltung (Fork) der Mapbox GL JS Bibliothek, die entstand, als Mapbox ihre Lizenz 2020 von Open Source zu proprietär änderte.

Vor- und Nachteile

- **Vorteile**
 - Modern und nutzt die GPU
 - in den StyleSheets lässt sich alles bis aufs kleinste Detail anpassen
- **Nachteile**
 - Viel Komplexer als Leaflet
 - Base-Maps schwieriger zu finden, oder kostenpflichtig über API-Keys

Code verstehen #1

```
let map = new maplibregl.Map({  
  container: "map",  
  style: "https://tiles.basemaps.cartocdn.com/gl/voyager-gl-style/style.json",  
  center: [13.4008747, 52.5169723],  
  zoom: 4,  
});
```

Code verstehen #1

```
let map = new maplibregl.Map({  
  container: "map",  
  style: "https://tiles.basemaps.cartocdn.com/gl/voyager-gl-style/style.json",  
  center: [13.4008747, 52.5169723],  
  zoom: 4,  
});
```

`let map` initialisiert unsere Kartenvariable

`new maplibregl.Map(options)` erstellt eine neue Karte mit den gegebenen

Optionen

`container` ist das HTML-Element, in dem die Karte angezeigt wird

`style` ist der Link zu einem Style-Sheet für die Karte

`center` und `zoom` sind die Ausgangsperspektive für die Karte

Code verstehen #2

```
let popup = new maplibregl.Popup({ offset: 25 }).setHTML(`A pretty CSS popup.<br> Easily customizable.`);  
let marker = new maplibregl.Marker().setLngLat([city.Long, city.Lat]).setPopup(popup);  
marker.addTo(map);
```

Code verstehen #2

```
let popup = new maplibregl.Popup({ offset: 25 }).setHTML(`A pretty CSS popup.<br> Easily customizable.`);  
let marker = new maplibregl.Marker().setLngLat([city.Long, city.Lat]).setPopup(popup);  
marker.addTo(map);
```

`new maplibregl.Popup(options)` erstellt ein neues Popup

`offset` ist der Abstand des Popups zum Marker

`setHTML(text)` setzt den Text für das Popup

`new maplibregl.Marker()` erstellt einen neuen Marker

`setLngLat([long, lat])` setzt die Koordinaten für den Marker

`setPopup(popup)` setzt das Popup für den Marker

`marker.addTo(map)` fügt den Marker zur Karte hinzu

Maplibre GL JS einbinden

```
<html>
  <head>
    <!-- Maplibre GL CSS -->
    <!-- Maplibre GL JavaScript -->
    <script src="unser_script.js" defer></script>
    <style>
      #map {
        height: 50vh;
      }
    </style>
  </head>
  <body>
    <div id="map"></div>
  </body>
</html>
```

Mein eigenes Fazit zu Maplibre GL JS?

Hat mir persönlich immer etwas besser gefallen als Leaflet

Die Karten sind schön, modern und schnell

Viele Anpassungsmöglichkeiten

Styles muss man sich zusammenbasteln

Wenn man darauf keine Lust hat, oder einfach schneller fertig sein möchte ->

Leaflet

Vielen Dank für eure Aufmerksamkeit!

Fragen?