HAPPY HIKING

Gruppenprojekt Webmapping:
Fritz, Kathleen & Johanna

HTTPS://HAPHIK.GITHUB.IO

index.html

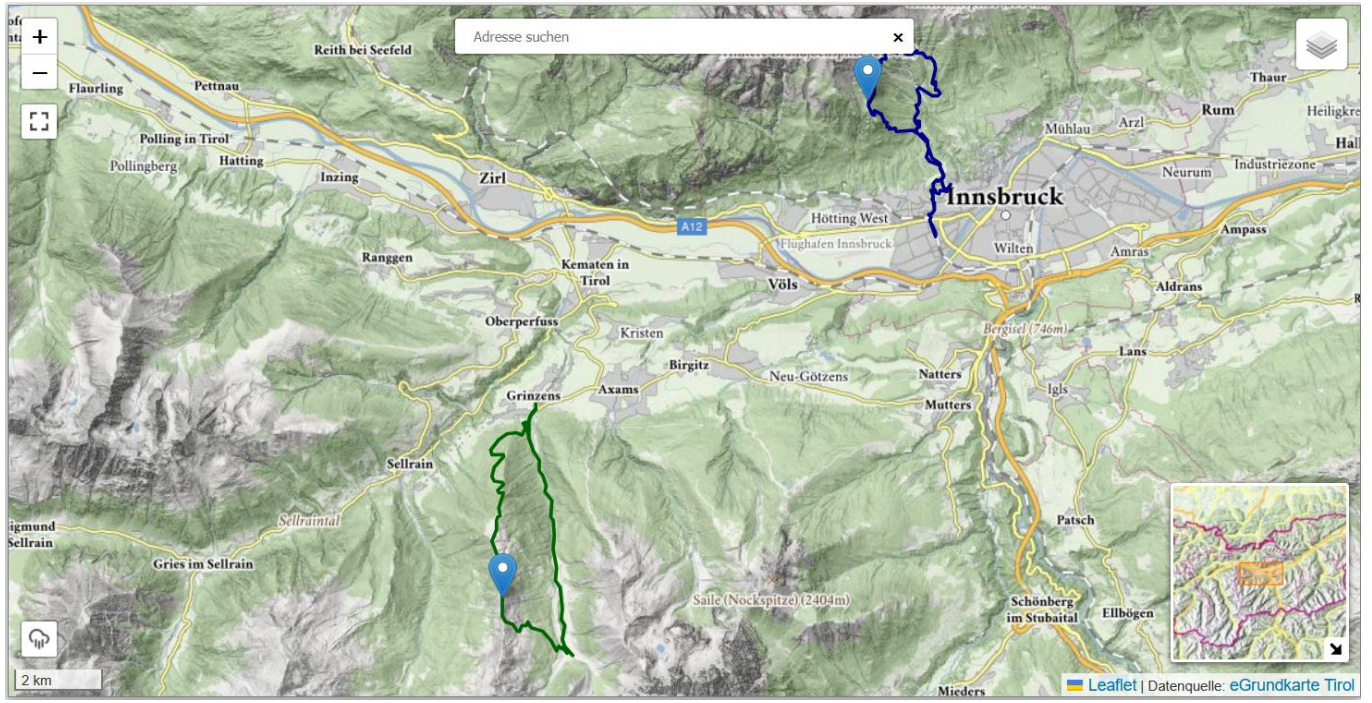
```
<header>
    <img src="trekking.jpg" alt="Trekking" />
    <!--https://pixabay.com/de/photos/wandern-gruppe-alpin-linie-299000/-->

    <!--KI_BEGIN-->
    <div class="overlay-nav">
        <h2><i class="fa-solid fa-mountain-sun"></i> Happy Hiking</h2>
        <nav>
            <ul class="menu">
                <li><a href="../index.html">Startseite</a></li>
                <li class="dropdown">
                    <a href="#">Routen ▾</a>
                    <ul class="dropdown-content">
                        <li><a href="../route1/index.html"><i class="fa-solid fa-person-hiking"></
                        i> Salfeinsee</a>
                        </li>
                        <li><a href="../route2/index.html"><i class="fa-solid fa-mountain"></i>
                        Achselkopf & Höttinger Alm</a></li>
                    </ul>
                </li>
            </ul>

        </nav>
        <!--KI_END-->
    </div>
</header>
```

ÜBERBLICK - KARTE AUF DER STARTSEITE

- **Ziel:** Übersicht über alle Wanderrouten
- Basierend auf Leaflet



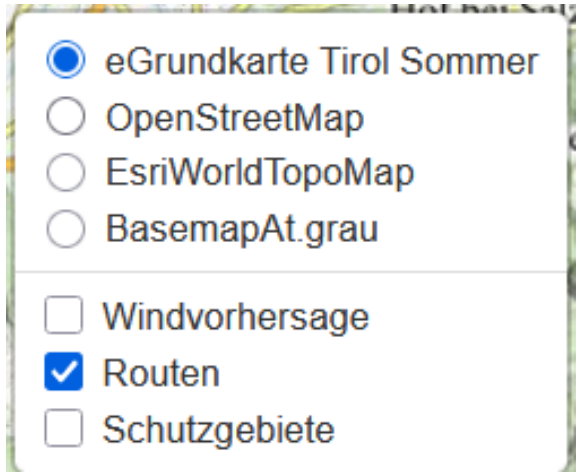

```
<div id="map"></div>
```

```
// Innsbruck
let ibk = {
    lat: 47.267222,
    lng: 11.392778
};

// Karte
let map = L.map("map").setView([ibk.lat, ibk.lng], 9);
```

HINTERGRUNDKARTEN UND OVERLAYS

- Auswahl per Layercontrol




```
// Hintergrundkarten
let eGrundkarteTirol = {
    sommer: L.tileLayer("https://wmts.kartetirol.at/gdi_summer/{z}/{x}/{y}.png", {···
    }),
    nomenklatur: L.tileLayer("https://wmts.kartetirol.at/gdi_nomenklatur/{z}/{x}/{y}.png", {···
    })
}

// / Layer control mit eGrundkarte Tirol und Standardlayern
L.control.layers({
    "eGrundkarte Tirol Sommer": L.layerGroup([···
    ]).addTo(map),

    "OpenStreetMap": L.tileLayer.provider("OpenStreetMap.Mapnik"),
    "EsriWorldTopoMap": L.tileLayer.provider("Esri.WorldTopoMap"),
    "BasemapAt.grau": L.tileLayer.provider("BasemapAT.grau"),
}, {
    "Windvorhersage": overlays.wind,
    "Routen": overlays.routen,
    "Schutzgebiete": overlays.schutzgebiete,
}).addTo(map);
```

OVERLAYS

Routen

```
//GPX-Route laden
new L.GPX("Salfeinsee.gpx", {
    async: true,
    marker_options: {
        startIconUrl: null,
        endIconUrl: null,
        shadowUrl: null
    },
    polyline_options: {
        color: "darkgreen",
        wight: 5
    }
}).on('loaded', function (e) {
    map.fitBounds(e.target.getBounds());
}).addTo(overlays.routen);
```

Windlayer

```
async function loadWindLayer() {
    try {
        const response = await fetch('https://geographie.uibk.ac.at/data/ecmwf/data/wind-10u-10v-europe.json');
        const data = await response.json();
```

```
const velocityLayer = L.velocityLayer({
    displayValues: true,
    displayOptions: {
        velocityType: "Wind",
        position: "bottomleft",
        speedUnit: "k/h",
        emptyString: "Keine Winddaten verfügbar",
        angleConvention: "meteo",
        showCardinal: true,
        directionString: "Richtung",
        speedString: "Geschwindigkeit (km/h)"
    },
    data: data,
    minVelocity: 0,
    maxVelocity: 10,
    velocityScale: 0.005,
    colorScale: [
        "#3288bd", "#66c2a5", "#abdda4", "#e6f598",
        "#fee08b", "#fdae61", "#f46d43", "#d53e4f"
    ],
    opacity: 0.97
}).addTo(overlays.wind);
```

Schutzgebiete

```
//Schutzgebiete
async function loadSchutzgebiete(url) {
    //console.log(url;)
    let response = await fetch(url);
    let jsondata = await response.json();
    //console.log(jsondata);
    L.geoJSON(jsondata, {
        attribution: "Datenquelle: <a href='https://www.data.gv.at/s
        style: function (feature) {
            //console.log(feature);
            return {
                color: "#F012BE",
                weight: 1,
                opacity: 0.4,
                fillOpacity: 0.1,
            };
        }
    }).addTo(overlays.schutzgebiete);
}
//GeoJSON laden und visualisieren
loadSchutzgebiete("Schutzgebiete.geojson")
```

PLUGINS

Maßstabsanzeige

```
// Maßstab
L.control.scale({
    imperial: false,
}).addTo(map);
```

MiniMap

```
//Minimap
var wmts = new L.TileLayer("https://wmts.kartetirol.at/gdi_summer/{z}/{x}/{y}.png", { minZoom: 0, maxZoom: 13 });
var miniMap = new L.Control.MiniMap(wmts, {
    toggleDisplay: true,
    minimized: false,
}).addTo(map);
```

Vollbildmodus

```
//Fullscreen
map.addControl(new L.Control.Fullscreen());
```

GeoSearch

```
//Geo Search
const searchControl = new GeoSearch.GeoSearchControl({
    provider: new GeoSearch.OpenStreetMapProvider(),
    style: "bar",
    searchLabel: "Adresse suchen"
});
map.addControl(searchControl);
```

Rainviewer

```
// Rainviewer initialisieren

L.control.rainviewer({
    position: 'bottomleft',
    nextButtonText: '>',
    playStopButtonText: '▶/⏸',
    prevButtonText: '<',
    positionSliderLabelText: "Zeit:",
    opacitySliderLabelText: "Deckkraft:",
    animationInterval: 300,
    opacity: 0.5
}).addTo(map);
```

MARKER UND POPUP

Routeninformationen

```
const ROUTE = [
    {
        lat: 47.188247,
        lng: 11.24499,
        zoom: 13,
        title: "Wanderung Salfeinsee",
        link: "../route1/index.html",
        length: "16.1 km",
        duration: "6:50 h",
        difficulty: "mittelschwer",
    },
```

Wetterdaten abrufen

```
async function showForecastForRoute(route, marker) {
    let url = `https://api.met.no/weatherapi/locationforecast/2.0/compact?lat=${route.lat}&lon=${route.lng}`;

    try {
        let response = await fetch(url);
        let jsondata = await response.json();

        let details = jsondata.properties.timeseries[0].data.instant.details;
        let timestamp = new Date(jsondata.properties.meta.updated_at);
        let temperature = details.air_temperature;
```

Popup-Inhalt zusammenstellen

```
let markup = `
    <h2><a href="${route.link}" target="_blank">${route.title}</a></h2>
    <ul>
        <li><strong>Länge:</strong> ${route.length}</li>
        <li><strong>Dauer:</strong> ${route.duration}</li>
        <li><strong>Schwierigkeit:</strong> ${route.difficulty}</li>
    </ul>
    <h4>Wettervorhersage</h4>
    <p>aktuelle Temperatur: ${temperature.toFixed(1)}°C</p>
    <p class="weather-meta">Stand: ${timestamp.toLocaleString()}</p>
    <div class="weather-icons">
`;
```

Wettervorhersage Symbole

```
for (let i = 0; i <= 12; i += 3) {
    let forecast = jsondata.properties.timeseries[i];
    if (!forecast?.data?.next_1_hours?.summary) continue;

    let symbol = forecast.data.next_1_hours.summary.symbol_code;
    let time = new Date(forecast.time);
    let hour = time.toLocaleTimeString([], { hour: '2-digit', minute: '2-digit' });

    markup += `
        <div class="weather-icon">
            <img src="https://api.met.no/images/weathericons/svg/${symbol}.svg" width="32" title="${hour}">
            <br><span>${hour}</span>
        </div>
    `;
```

Marker und Popup setzen

```
// Marker + Popup erstellen
for (let route of ROUTE) {
    let marker = L.marker([route.lat, route.lng]).addTo(overlays.routen);
    showForecastForRoute(route, marker);
}
```

ROUTENSEITEN

Header: Menüführung

Article:

- Titel
- Font Awesome Icons, und GPX-Track download
- Kurzbeschreibung, Foto, Streckenbeschreibung, Einkehrmöglichkeiten, Anreise
- Leaflet-Map mit "Locate-Control" und "Mini-Map"
- GPX-Track und "profile"

Footer:

- Datenquellen mit Links

Ziele:

- Beschreibung der Wanderung
- Übersichtliche und ansprechende Darstellungen
- Locate-Control für während der Wanderung
- Einbindung des GPX-Tracks und Visualisierung mit "profile"
- Einfacher Aufbau für die Replikation mit anderen Wanderungen

ZENTRALE PLUGINS FÜR DIE ROUTENSEITEN

Leaflet-Elevation Plugin

```
<div id="map"></div>
<div id="profile"></div>
```

```
// leaflet plugin elevation
var controlElevation = L.control.elevation({
    theme: "hike",
    time: false,
    elevationDiv: "#profile",
    height: 300,
}).addTo(map);
controlElevation.load("AchselbodenundHoettingerAlm.gpx")
```

```
#profile {
  border: 1px solid gray;
  padding: 1em;
  margin-top: 1.5em;
}
```

Locate-Control Plugin

```
// Leaflet Locate Control
L.control.locate({
    strings: {
        title: "eigenen Standort anzeigen"
    },
    drawCircle: false,
}).addTo(map);
```

HERAUSFORDERUNGEN UND FAZIT

Leaflet-Velocity Plugin und die richtige Windrichtung

Rainviewer Plugin, MET Norway und die fehlende Prognosefunktionen

KI-Nutzung für CSS

Erweiterung der Webseite

- Mehr Routenseiten
- Sinnvolle Plugins, wie ÖBB Scotty Plugin
- Angelehnte Konzepte wie Happy Skiing, Happy Biking etc.