

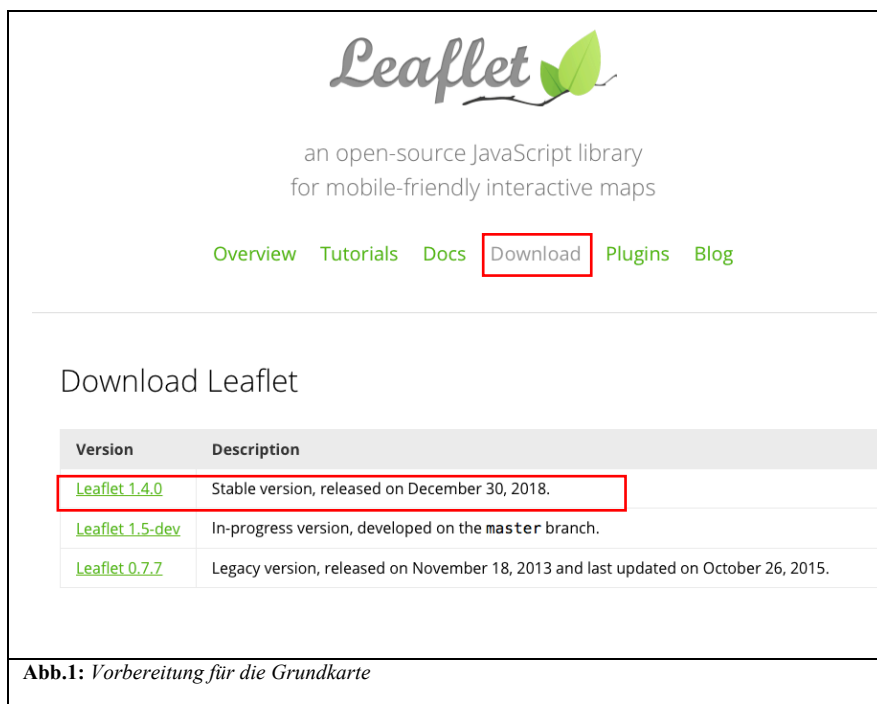
Cheat Sheet zum Thema Grundkarte im Leaflet | Workload 6 |

1. Was ist „Leaflet“? Für was wird „Leaflet“ verwendet?

Der Begriff Leaflet wird in der Programmiersprache von JavaScript im Geographischen Informationssystem als Flugblatt oder Flyer definiert und dient als Open-Source Webmapping Bibliothek für die Erstellung von Web-Maps. Mithilfe von Leaflet ist es möglich Open-Source Karten (z.B. Übersichtskarten) sowie Daten auf einer selbst erstellten Webseite (GitHub) einzubinden und zu visualisieren. Für die Bearbeitung bzw. der Darstellung von Karten werden die Dateien im HTML, CSS und im JavaScript verwendet.

2. Einbau der Grundkarte im HTML

Der Erster Schritt, um entsprechend Web-Maps programmieren zu können ist die Einfügung des Open-Source Codes in das HTML Grundgerüst. Unter den Link <https://leafletjs.com> unter Download



Leaflet

an open-source JavaScript library
for mobile-friendly interactive maps

[Overview](#) [Tutorials](#) [Docs](#) [Download](#) [Plugins](#) [Blog](#)

Download Leaflet

Version	Description
Leaflet 1.4.0	Stable version, released on December 30, 2018.
Leaflet 1.5-dev	In-progress version, developed on the master branch.
Leaflet 0.7.7	Legacy version, released on November 18, 2013 and last updated on October 26, 2015.

Abb.1: Vorbereitung für die Grundkarte

findet man die im Kurs angewandte Leaflet Version 1.4.0. Die grundsätzliche Voraussetzung dabei ist, dass „*leaflet.css*“ sowie „*leaflet.js*“ im Index eingebunden sein müssen, um eine Verbindung zur CSS-Datei sowie zur JavaScript-Datei für die Kartengenerierung herstellen zu können. Den entsprechenden Code für die Erstellung der Grundkarte findet man auf der aufgeführten Downloadseite unter den Bereich der aufgelisteten Download Versionen (Abb.2). Dieser Code wird kopiert und ins HTML-Verzeichnis (index.html) im Bereich <head> positioniert (Abb.3).

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css" />
<script src="https://unpkg.com/leaflet@1.4.0/dist/leaflet.js"></script>
```

Abb. 2: HTML Code

<https://leafletjs.com/download.html>

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title> 03-Etappe Lake Tekapo </title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" type="text/css" media="screen" href="main.css">

  <link href="https://fonts.googleapis.com/css?family=Open+Sans|Roboto" rel="stylesheet">

  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css"
    integrity="sha384-50oBUHEmvpQ+1lW4y57PTFmhCaXp0ML5d60M1M7uH2+nqUivzIebhnd0JK28anvf" crossorigin="anonymous">

  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css" />
  <script src="https://unpkg.com/leaflet@1.4.0/dist/leaflet.js"></script>

  <script src='https://cdnjs.cloudflare.com/ajax/libs/leaflet-hash/0.2.1/leaflet-hash.js'></script>

  <script src='https://api.mapbox.com/mapbox.js/plugins/leaflet-fullscreen/v1.0.1/Leaflet.fullscreen.min.js'></script>
  <link href='https://api.mapbox.com/mapbox.js/plugins/leaflet-fullscreen/v1.0.1/leaflet.fullscreen.css'
    rel='stylesheet' />
  <script src='https://cdnjs.cloudflare.com/ajax/libs/leaflet-hash/0.2.1/leaflet-hash.js'></script>

  <link href="plugins/control.coordinates.css" rel="stylesheet">
  <script src="plugins/control.coordinates.js"></script>

  <script defer src="main.js"></script>
</head>
```

Abb. 3: Position im HTML-Verzeichnis

Schritt 2:

Damit eine Übersichtskarte im weiteren Verlauf angezeigt werden kann, muss im HTML-Verzeichnis, im <body> ein div-Element (id =“map“) mit weiteren relevanten Attributen eingebaut werden (siehe Abb.4). Die Koordinationsdaten für die spätere Positionsmarkierung sind als „*data-lat*“ [Breitengrad] und „*data-lng*“ [Längengrad] mit dazugehörigen „*data-title*“ zu benennen. Zu beachten sind die einheitlichen Schreibversionen, denn jeder Tippfehler kann Auswirkungen auf die Kartenerzeugung haben.

```

<h3>Übersichtskarte</h3>
<div id="map" data-lat="-44.0035" data-lng="170.482333" data-title="Church of the good Shepherd">
</div>

```

Abb.4: relevante Attribute für die Grundkarte

Schritt 3:

Um den Grundkasten (siehe Abb.5) für die Übersichtskarte zu erhalten müssen im „main.css“ File sowie im „main.js“ File die folgenden Grundbefehle eingegeben werden:

Befehl → main.css – Datei

```

1 #map {
2   height: 360px;
3   width: 90%;
4   border: 1px solid black;
5 }

```

Befehl → main.js – Datei

```

1 //Karte initialisieren
2
3 let karte = L.map("map");
4
5 karte.setView(
6   [breite, laenge],
7   13);
8

```

Erscheint durch die eingegebenen Befehle ein Kasten mit grauem Hintergrund ist eine Reaktion bzw. Verbindung mit Leaflet vorhanden. Dementsprechend kann zum nächsten Schritt hingearbeitet werden, indem eine Basiskarte aus OpenStreetMap als Hintergrundlayer eingefügt wird.

Übersichtskarte

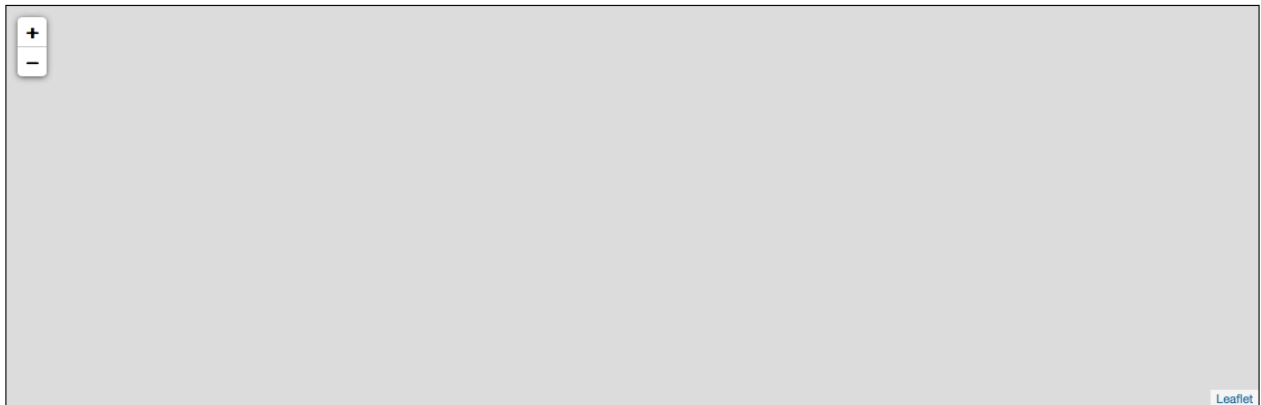


Abb. 5: Grundkasten für die Basiskarte

3. Anzeige einer einfachen Karte im OpenStreetMap als Hintergrundlayer

Um eine einfache OSM Basiskarte, wie im aufgeführten Beispiel als Hintergrundlayer zu erhalten, müssen im HTML-Verzeichnis die Koordinationsdaten ermittelt und aufgeführt werden.

Übersichtskarte



Abb.6: Beispiel einer einfachen Basiskarte

Dementsprechend werden die Koordinationsdaten [als Attribute] im „main.js“ File unter dem Schlüsselwort „const“ als Konstante aufgeführt. Zu den schon vorhandenen Befehlen aus dem Schritt drei, wie der „Karten Initialisierung“ und dem „Kartenausschnitt“ wird der Befehlscode für das Hinzufügen der OpenStreetMap Karte ergänzt. In diesem Bereich bzw. mit dem Befehlscode können nach eigener Wahl sog. „Kacheln“ [tiles] hinzugefügt werden.

```
1 //alert("Hallo Welt!");
2
3 const div = document.getElementById("map");
4 const breite = div.getAttribute("data-lat");
5 const laenge = div.getAttribute("data-lng");
6 const titel = div.getAttribute("data-title");
7
8 //console.log(breite, laenge, titel)
9
10 //Karte initialisieren
11
12 let karte = L.map("map");
13
14 //console.log(karte);
15 //aus Ausschnitt zoomen
16
17 karte.setView(
18   [breite, laenge],
19   13);
20
21 //openstreetmap hinzufügen
22 L.tileLayer("http://{s}.tile.osm.org/{z}/{x}/{y}.png").addTo(karte);
```

4. Einfügen eines Positionsmarkers

Für das Einfügen eines Positionsmarkers in der Karte, wird folgender Befehl im JavaScript – File eingegeben. Die „breite“ und „laenge“ sind hierbei, dass unter „const“ definierte Attribute der Koordinationsdaten.

```
//Positionsmarker hinzufügen
let pin = L.marker([breite, laenge]).addTo(karte);
```