

20. Juni 2017

## **Projekt “Mit dem Rennrad von Innsbruck zum Gardasee”**

### **Inhaltsverzeichnis**

- 1. Aufbau der HTML-Seite**
- 2. Erstellen der gpx-tracks mittels “gpsies-Tourenplaner”**
- 3. Layout der Website**
- 4. Plugins und Struktur der JavaScript-Dateien**

#### **1. Aufbau der HTML-Seite**

Der Aufbau der Startseite der “Transalp von Innsbruck zum Gardasee” beginnt mit der Überschrift “Mit dem Rennrad von Innsbruck zum Gardasee”. Anschließend folgt eine Übersichtskarte mit den vier Etappen. Diese vier Etappen sind in der Startseite zu einer Etappe zusammengefasst. In der Karte sind außerdem die Kurzinformationen zu Länge und Dauer der Etappen in Markern am jeweiligen Startpunkt beschrieben. Ein Höhenprofil der gesamten Route gibt einen kurzen Überblick über die zu bewältigenden Höhenmeter, welche in den vier Seiten mit den einzelnen Etappen detaillierter aufgeschlüsselt werden. Das Skript für die Karte wird in eine eigene Datei ausgelagert und auf der HTML-Seite eingebunden. Als Kartengrundlage wird die frei zugängliche Open Street Map (2017) benutzt, da diese eine sehr gute Grundlage für Radfahrer bietet und außerdem die Route über eine Landesgrenze verläuft.

Der zugrundeliegende Style der Seite wird in einem ebenfalls ausgelagerten Stylesheet festgelegt. Dieses Stylesheet hat Gültigkeit für jede der fünf HTML-Seiten. Eine genauere Beschreibung des Layouts kommt im weiteren Verlaufe des Textes.

Die Karte wird mit dem Tool Leaflet, einem Tool zum Einbauen von Karten der Webseite hinzugefügt. Der Karte werden zwar zentrale Koordinaten zugeordnet, jedoch wird weiter im Skript der Sichtfokus auf die Route gelegt (Bild). Die Route wird als gpx-Datei mit x-,y- und z- Koordinaten eingebaut. Durch die Dreidimensionalität kann die Steigung der Route

zwischen den einzelnen GPS-Koordinaten durch verschiedene Farben eingezeichnet werden. Dies wird im weiteren Verlauf des Textes genauer erklärt. Eine starke Steigung mit größer  $20^\circ$  hat eine dunkelrote Farbgebung und ein starkes Gefälle mit größer  $20^\circ$  hat eine dunkelgrüne Farbe. Im nachfolgenden Screenshot sind die Steigungen bzw. Gefälle anhand der Farben sehr deutlich zu erkennen.

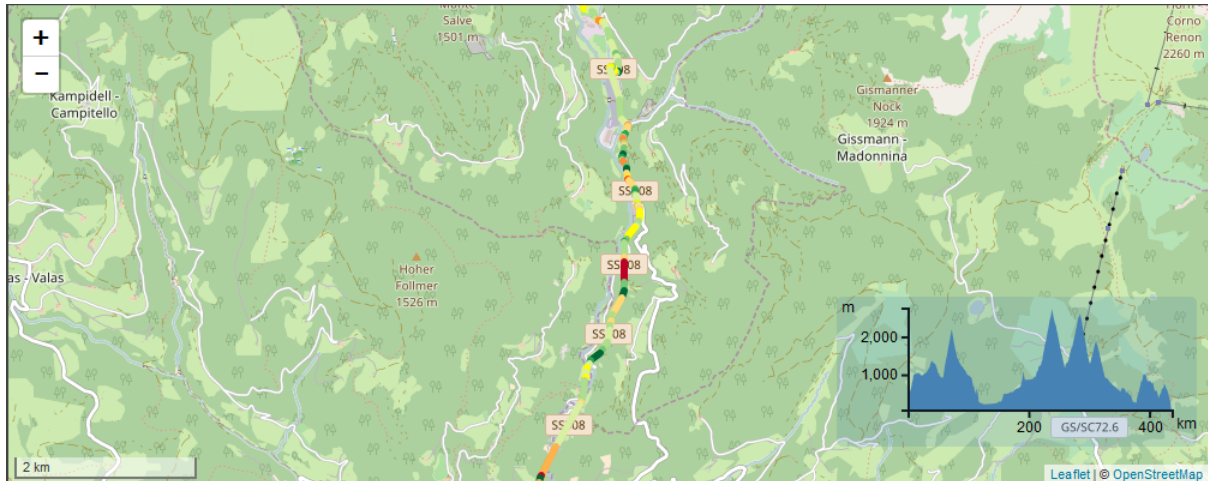
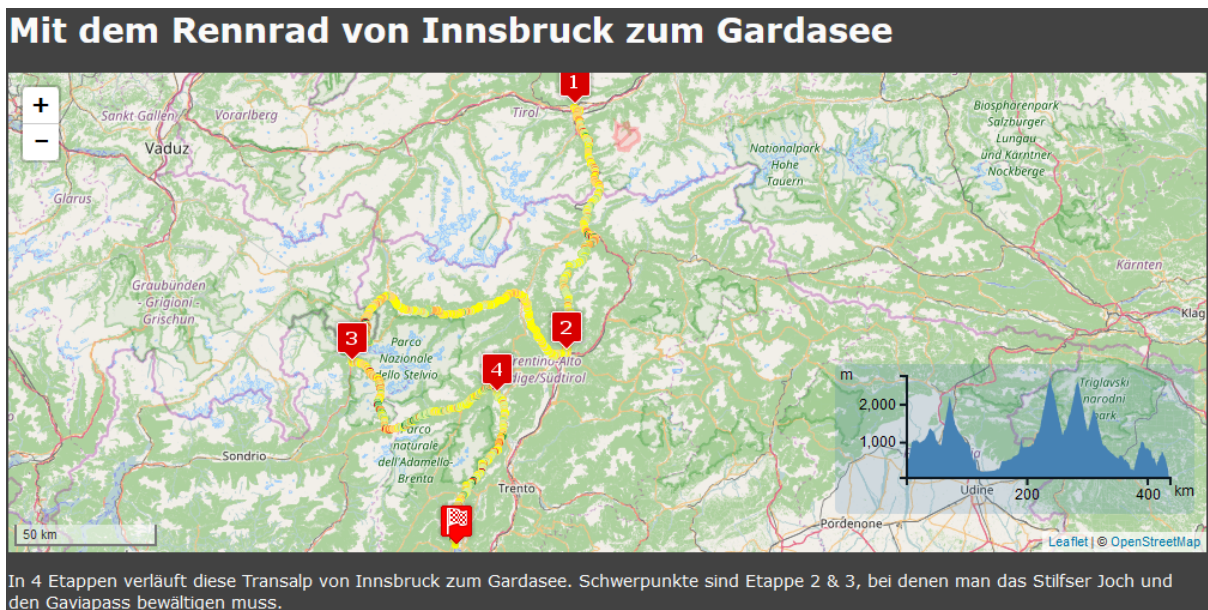


Abbildung 1: Darstellung Steigung und Gefälle

Die benötigten Plugins sind ebenfalls in anderen Skripten ausgelagert und werden im weiteren Verlauf des Textes genauer behandelt.

In der Karte sind außerdem eine Zoom-Kontrolle sowie eine Maßstabsleiste und Quellenangaben eingebettet. Der nachfolgende Screenshot zeigt die Überschrift mit Karte, den Plugins und Kurztext.



In 4 Etappen verläuft diese Transalp von Innsbruck zum Gardasee. Schwerpunkte sind Etappe 2 & 3, bei denen man das Stilsfer Joch und den Gaviapass bewältigen muss.

Abbildung 2: Karte der Startseite mit Plugins

Unterhalb der Karte findet sich eine Kurzbeschreibung der dargestellten Transalp.

Die vier anschließenden Bilder repräsentieren jeweils eine Etappe und sind per Klick mit den Seiten der einzelnen Etappen verlinkt. Um dies etwas offensichtlicher zu gestalten, werden die einzelnen Etappen unterhalb der Bilder nochmals genannt, wie im Screenshot zu sehen.



Abbildung 3: Navigation der vier Etappen

Der nächste Screenshot zeigt die Einbindung und Verlinkung der Bilder mit den jeweiligen Seiten.

```
<div class="Bild1"><p><a href="Etappe1.html">

</a></p><p>Etappe 1: Innsbruck-Bozen</p></div>

<div class="Bild2"><p><a href="Etappe2.html">

</a></p><p>Etappe 2: Bozen-Bormio</p></div>

<div class="Bild3"><p><a href="Etappe3.html">

</a></p><p>Etappe 3: Bormio-Cles</p></div>

<div class="Bild4"><p><a href="Etappe4.html">

</a></p><p>Etappe 4: Cles-Riva del Garda</p></div>
```

Abbildung 4: Verlinkung der Etappenbilder

Die zu erreichenden Seiten der vier Etappen sind sehr ähnlich aufgebaut. Als Marker angezeigt wird jeweils nur der Start- und Endpunkt der Etappe. Für jede Etappenseite ist auch ein eigenes Skript zur Einbindung der Karte vorhanden, welches nur die jeweiligen Marker und die jeweilige Etappe anzeigt. Der nachfolgende Screenshot zeigt die Struktur der Java-Skripts und die jeweiligen Skripts für die Startseite und die vier Etappen.

Branch: master BuFabia.github.io / projekt / js /		Create new file Find file History
BuFabia Dauer und Länge hinzugefügt, Titel geändert		Latest commit 92ce396 18 hours ago
..		
etappeninfo	html seiten, js, css, quellen	19 hours ago
leaflet-elevation	html seiten, js, css, quellen	19 hours ago
leaflet-hash	html seiten, js, css, quellen	19 hours ago
leaflet-omnivore	html seiten, js, css, quellen	19 hours ago
leaflet	html seiten, js, css, quellen	19 hours ago
map.js	Dauer und Länge hinzugefügt, Titel geändert	18 hours ago
map_etappe1.js	Data Pfad geändert und gpxLoad	19 hours ago
map_etappe2.js	Data Pfad geändert und gpxLoad	19 hours ago
map_etappe3.js	Data Pfad geändert und gpxLoad	19 hours ago
map_etappe4.js	Data Pfad geändert und gpxLoad	19 hours ago

Abbildung 5: Struktur der JavaScripts auf github.com

In der obersten Zeile der Etappenseiten findet sich ein Link zurück zur Startseite. Anschließend folgen der Titel der Etappe, eine Kurzbeschreibung und die Karte, wie im untenstehenden Screenshot zu sehen.

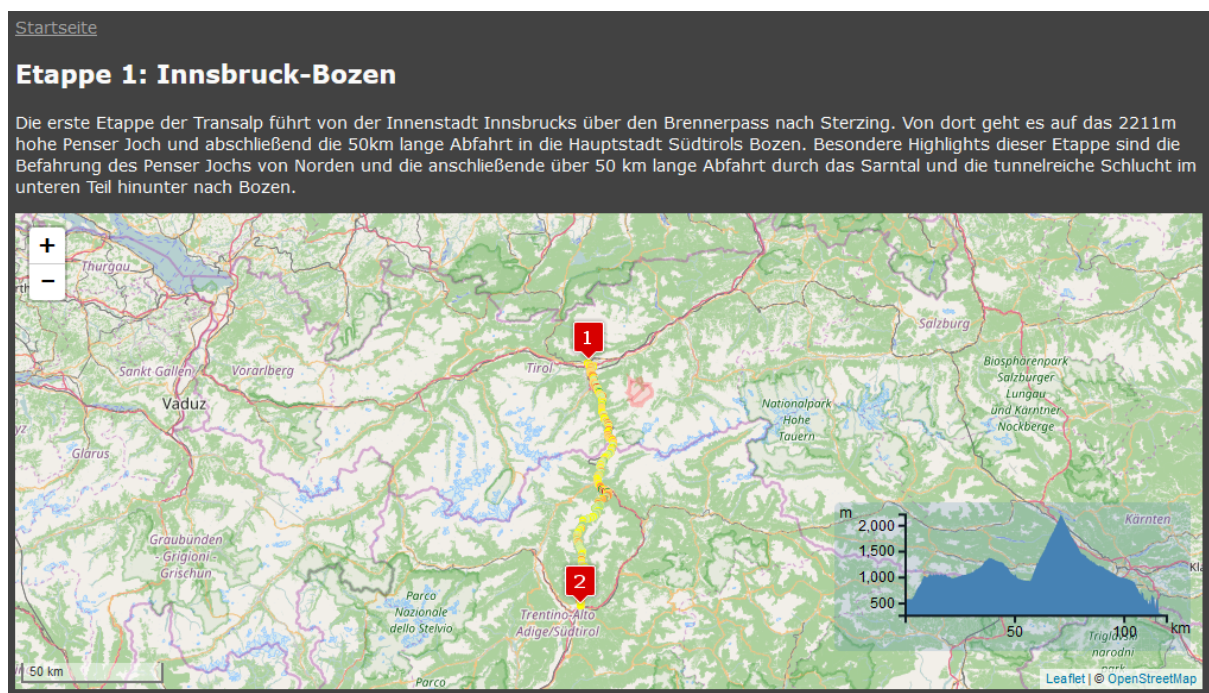


Abbildung 6: Etappe 1 mit Kurzbeschreibung

Als nächstes kommen eine ausführliche Streckenbeschreibung, Angaben zu den Höhenmetern, dem höchsten Punkt, dem Schwierigkeitsgrad, der Länge, der in etwa benötigten Zeit ohne Pausen und die zu fahrenden Pässe. Die Struktur der Verlinkung mit den anderen Etappen erfolgt wie bei der Startseite mit den Bildern. Die auf der Website dargestellten gpx-Tracks wurden mithilfe einer online Tourenplaner-Software erstellt und wie



dies genau funktioniert und auf was man dabei beachten muss wird im folgenden Kapitel ausführlicher beschrieben.

## 2. Erstellen der gpx-tracks mittels “gpsies-Tourenplaner”

Die gpx-tracks für dieses Projekt werden mit der Tourenplaner-Software der Website “gpsies.com” erstellt. Die Erstellung der gpx-tracks funktioniert wie folgt: Zu Beginn sucht man sich einen Startpunkt auf der bereits bestehenden Karte des Routenplaners aus. Diese Karte wurde auch mit dem leaflet-tool eingefügt und Kartengrundlage ist auch “openstreetmap”. Dort setzt man dann einen Marker mittels Doppelklick, um den Startpunkt für die jeweilige Tour zu kennzeichnen. Anschließend muss man oben bei den Tools die Streckenführung-Variante auswählen. Da das Ziel dieses Projekts eine Rennradstrecke sein soll, muss man oben auf das Icon klicken, welches die Rennradstrecken berechnet. Dieses Icon findet man oben rechts auf der Karte unter der Überschrift “Wegen folgen”, wie man auf Abbildung 7 gut erkennt.

Entdecken - Erstellen - Magazin - Konvertieren - Über GPSies - fusulierchristopher

### Strecke planen

Geschwindigkeit (km/h)  Pace (min/km)  Geschätzte Dauer  Name der Strecke

☐ Wegen folgen

☐ GraphHopper

☐ Google

☐ Scrollzoom

☐ Zentrieren

☐ Wegepunkte

☐ Meter / km

☐ Punkte auswählen und entfernen

2230 PT | 0 WPT

118,071 km

Höhe: 829 m

Hilfe

Rückgängig (5/20)

Höhendaten

Altitude (m)

Abbildung 7: Planung der Etappen

Wenn man dies nicht tut, berechnet der Routenplaner lediglich die Luftlinie oder wenn man ein anderes Icon auswählt berechnet er die kürzeste Strecke, welche auch über Feldwege oder Singletrails führen kann. Dies würde man auswählen, wenn man beispielsweise eine Transalp mit dem Mountainbike oder dem Trekkingrad plant. Nachdem man die Routenparameter eingestellt hat kann man mit dem Erstellen der jeweiligen Route beginnen. Der Startpunkt dieser Transalp ist Innsbruck und nun geht es darum, die Etappen in etwa gleich groß aufzuteilen auf dem Weg nach Riva del Garda. Die Ziele der jeweiligen Etappen sind außerdem so auszuwählen, dass man jedes Mal relativ leicht eine Übernachtungsmöglichkeit findet, weshalb sich größere Städte wie Bozen oder vom Tourismus geprägte Orte, wie Bormio, gut eignen. Da man ja mit dem Rad unterwegs ist, sollten Start und Ziel der jeweiligen Etappen auch am selben Ort sein, um eine Anreise zum nächsten Ziel, was ja zusätzliche Kilometer wären, zu vermeiden. Um noch einige Highlights der Alpen in die Transalp mit einzubauen, wird ein Umweg über das Vinschgau und das Veltlin gefahren. Hierdurch ermöglicht es der Tour noch die beiden unter Rennradfahrern sehr beliebten Pässe Stilsfer Joch und Gavia in die Route mit einzubauen. Nun zum Ablauf des Erstellens der gpx-tracks: Nachdem man den Anfangspunkt per Doppelklick auf der Karte ausgewählt hat, kann man sich weitere Punkte in der Karte per Doppelklick einzeichnen und diese verbinden sich dann automatisch und folgen der für Rennradler am besten geeigneten Straßen. Meistens folgt der Routenplaner verkehrsarmen Landstraßen oder versucht die in der Umgebung gelegenen Radwege zu nutzen und somit die Route möglichst autofrei zu gestalten. Die auf Abbildung 7 erkennbaren Wegpunkte wurden nicht manuell eingetragen, sondern werden von der Software selbst eingetragen und sollen helfen die Genauigkeit zu erhöhen, damit man unterwegs auf dem GPS-Gerät dem Track möglichst einfach folgen kann. Wenn man genug Wegpunkte eingezeichnet hat und man mit dem Streckenverlauf zufrieden ist, kann man sich ein Ziel suchen und dort mit erneutem Doppelklick wieder die Routenführung beenden. Anschließend reicht es die Strecke abzuspeichern unter einem entsprechenden Namen. Bei diesem Projekt werden die gpx-Dateien Etappe1.gpx-Etappe4.gpx genannt. Anschließend kann man die Touren online auf der Website von gpsies abrufen und erhält dort bereits ein Höhenprofil und die Software berechnet auch die Höhenmeter auf- und abwärts. Man kann auf dieser Plattform außerdem die gpx-Daten frei zur Verfügung (falls gewünscht) stellen und somit können sich andere Nutzer dieser Website die Tourdaten kostenfrei herunterladen und frei nutzen, wie man auf den Abbildungen 8 und 9 am Beispiel der ersten Etappe erkennt. Um auf der Startseite dieses Projekts den gesamten Track sichtbar zu machen, werden die gpx-Dateien mit dem "Notepad++" bearbeitet. Es reicht dabei die einzelnen Dateien in einem Script zusammen zu fügen und unter einem neuen Namen abzuspeichern. Diesen neu erstellten gpx-track muss

dann im Pfad der html-Startseite verlinkt werden. Wenn man nun die Startseite aufruft, ist die gesamte Route mitsamt Höhenprofil sichtbar.

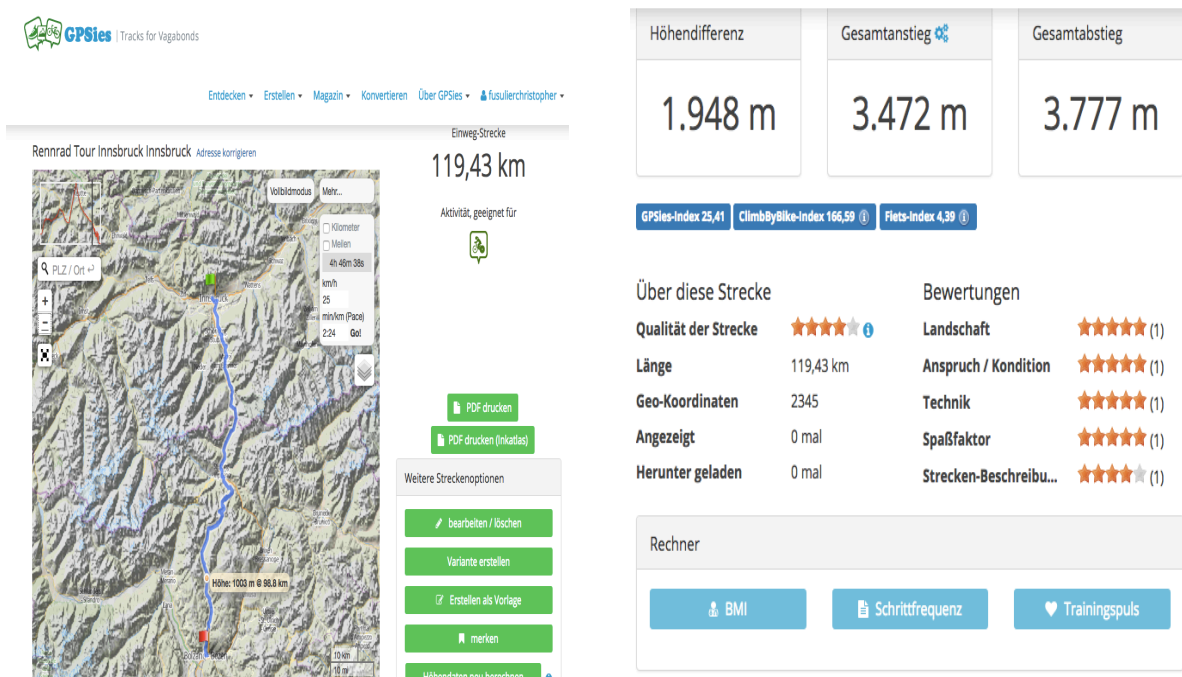


Abbildung 8: Planung der Etappen mittels gpsies.com und Details der Streckenplanung

Im folgenden Kapitel wird nun kurz auf das Stylesheet, die auf der Website eingebetteten Bilder sowie auf die beschreibenden Texte der Website eingegangen.

### 3. Layout der Website

Das Layout der Website wird mittels Stylsheet (CSS-Datei) separat erstellt und im Header der jeweiligen Html-Seiten mit eingebunden. Es gilt also eine CSS-Datei für sämtliche Html-Seiten. Ziel dieser Datei ist das einheitliche Layout der Html-Seiten und das korrekte einrücken der Bilder, die zu den jeweiligen Etappen-Seiten führen. Der Seite werden zwei Rahmen zur sauberen Verteilung der Seitenelemente hinzugefügt. Dieser Rahmen ermöglicht es beispielsweise die Etappen Bilder in einer einheitlichen Größe darzustellen. Die Bilder der Website stammen von der Suchmaschine "Pixabay" und sind somit frei zugänglich ohne rechtliche Einschränkung. Die URL der jeweiligen Bilder sind im Quellenverzeichnis angegeben. Da die Bilder in einer relativ hohen Auflösung und unterschiedlichen Größen sind, werden die Bilder mittels Paint bearbeitet und in eine einheitliche Größe zugeschnitten. Die Beschreibungen der Etappen sind selbst erstellt und anhand von Tourenportalen (quäldich.de; suedtirol-kompakt.com) erstellt und an die jeweiligen Touren angepasst. Die Streckenbeschreibungen werden anhand der Abbiegehinweise der gpx-tracks erstellt.

```

html {
  background: #424242;
  padding: 5px 2%;
}
body {
  padding: 15px;
  margin: 15px;
  position: relative;
  font-family: Verdana;

  height: 80em;
  outline: 1px solid;
  width: 50em;

  font-size: 14px;
  color: #FAFAFA;
}
html, body, #map {
  height: 600px;
  width: 1000px;
}
#map {
  height: 400px;
  width: 1000px;
}
.Bild1, .Bild2, .Bild3, .Bild4 {
  float: left;
  outline: 1px solid;
  width: 250px;
  height: 140px;
}
a {
  color: #A4A4A4
}

```

Abbildung 9: Layout der Homepage

Die Website wird außerdem mit einigen Plugins versehen, wie beispielsweise Leaflet-Hash, Leaflet-Elevation und Omnivore. Wie genau bei diesen Plugins vorzugehen ist, wird im folgenden Kapitel beschrieben.

#### 4. Plugins und Struktur der JavaScript-Dateien

Um die Funktionalität und die verschiedenen Möglichkeiten der Darstellung von Karten mittels Leaflet noch zu erweitern, werden einige Leaflet Plugins verwendet, die frei zugänglich sind. Damit diese Plugins funktionieren, müssen sie in der index.html-Datei eingebettet werden (siehe Screenshot).

```

<script src="js/leaflet/leaflet.js"></script>
<script src="js/leaflet-hash/leaflet-hash.js"></script>
<script src="js/leaflet-elevation/leaflet.elevation-0.0.4.min.js"></script>

```

Abbildung 10: Einbettung der JavaScript-Dateien in index.html

Das erste Plugin, das benutzt wird, ist das generelle Leaflet Plugin, mit dem die Karte in das JavaScript eingebettet wird. Gleichzeitig wird auch die css-Datei des Plugins eingefügt, sodass der Style des Plugins übernommen wird.

Neben dem generellen Leaflet Plugin stellt das Plugin Leaflet-Omnivore einen sehr wichtigen Teil in diesem Projekt dar. Leaflet an sich kann in seiner Grundeinstellung nur Dateien lesen, die im GeoJSON-Format vorliegen. Gesetz dem Fall es besteht die Notwendigkeit, dass mit anderen Formaten wie CSV oder GPX gearbeitet werden muss, so macht dieses Plugin das Lesen der Dateien möglich. Dieses Projekt basiert auf gpx-



Koordinaten, sodass das Plugin Leaflet-Omnivore hier Anwendung findet. Es wird, ebenso wie alle anderen Plugins, am Anfang der index.html-Datei eingebettet.

Ein weiteres Plugin, das in diesem Projekt Verwendung findet, nennt sich Leaflet-Hash. Dieses Plugin ist zuständig für die Darstellung von Koordinaten in der URL-Leiste. Diese Koordinaten sind jedoch dynamisch, sodass sie, je nachdem an welchem Punkt sich der Mittelpunkt der Karte befindet, ihre Koordinaten ändert. Die Datei ist, so wie alle anderen Plugins, frei verfügbar auf der jeweiligen Github-Domain.

Nachdem Web-Karten vor allem den Endnutzer visuell ansprechen sollen, die Handhabung möglichst einfach sein soll und es in diesem Projekt um die Überquerung der Alpen geht, ist das Plugin Leaflet-Elevation ein hilfreiches Werkzeug. Mit diesem Plugin kann man ein interaktives Höhenprofil von Polylinien der Karte hinzufügen. Nachdem dieses Plugin aber noch in der Entwicklungsphase ist muss beachtet werden, dass nur die Browser *Chrome*, *Firefox* und *IE 10+* dieses Plugin anzeigen können. Zudem werden nur gpx-Daten unterstützt. Damit aber das Höhenprofil auch angezeigt werden kann, müssen die Höheninformationen für jeden Punkt des Tracks angegeben sein. Bei der Visualisierung der Höhenprofile kann zudem unter drei verschiedenen Farben gewählt werden, die bei diesem frei zugänglichen Plugin mit dabei sind. Die Farben sind lime, purple und steelblue. Für eine weitere Visualisierung der Strecke in der Karte, werden für bestimmte Steigungen und Gefälle im Gelände in verschiedenen Farben dargestellt. Das ist möglich, da die gpx-Tracks auch die Höhenmeter beinhalten. Für diese Darstellung muss zuerst eine Variable gebildet werden, die die jeweilige Einfärbung dann definiert.

```
var farbe;
switch(true) { // checks if condition is true, not for certain values of a variable
  case (deg >= 20) : farbe = "#bd0026"; break;
  case (deg >= 15) : farbe = "#f03b20"; break;
  case (deg >= 10) : farbe = "#fd8d3c"; break;
  case (deg >= 5) : farbe = "#feb24c"; break;
  case (deg >= 1) : farbe = "#fed976"; break;
  case (deg >= -1) : farbe = "yellow"; break;
  case (deg >= -5) : farbe = "#d9f0a3"; break;
  case (deg >= -10) : farbe = "#add8e6"; break;
  case (deg >= -15) : farbe = "#78c679"; break;
  case (deg >= -20) : farbe = "#31a354"; break;
  case (deg < -20) : farbe = "#006837"; break;
}
```

Abbildung 11: Klassenbildung und Einfärbung von Steigung und Gefälle

Damit nun auch die Plugins funktionieren müssen sie nun auch in die jeweiligen JavaScript-Dateien implementiert werden. Das muss für alle Dateien der Fall sein (map.js, map\_etappe1.js, map\_etappe2.js, map\_etappe3.js, map\_etappe4.js), da die Features in allen Etappen gleich sein sollen. Wie das aussieht soll an folgendem Bild erläutern werden:

```
// Höhenprofil control hinzufügen
var profileControl = L.control.elevation({
  position : 'bottomright',
  theme : 'steelblue-theme',
  width: 300,
  height: 125,
});
profileControl.addTo(map);
```

Abbildung 12: Einbettung von Leaflet-Elevation

Um ein Höhenprofil in der Karte anzeigen zu können muss zuerst eine Variable erstellt werden, die in diesem Fall den Namen profileControl hat. In dieser Variable wird nun auf das Leaflet-Elevation Plugin Bezug genommen. Im gleichen Zug wird die Position und die Größe bestimmt, dazu kommt das Aussehen des Höhenprofils (siehe Bild).

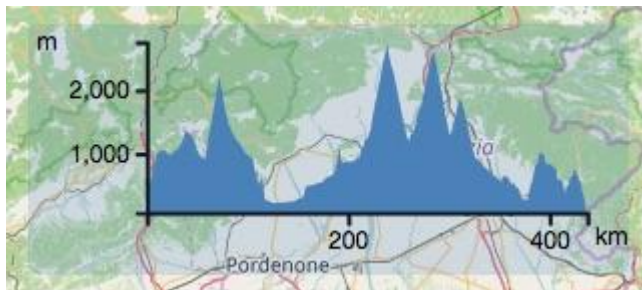


Abbildung 13: Höhenprofil einer Etappe

Zur generellen Struktur der JavaScript-Dateien ist zu sagen, dass sie in allen Etappen und der Startseite gleich aufgebaut ist. Nach dem Einbetten der Karte, des Maßstabs und des Höhenprofils müssen die gpx-Tracks der Karte zugeordnet werden. Wie schon erwähnt benötigt man hier das Leaflet-Omnivore Plugin. Innerhalb dieser Funktion werden auch die Ausschnitte der jeweiligen Etappen gesetzt, die einzelnen Koordinaten definiert, das Höhenprofil auf die jeweilige Etappe zugeschnitten und Steigung und Gefälle der Strecke berechnet, sodass hier die typische Einfärbung erfolgen kann. Anschließend wird das Plugin Leaflet-Hash eingebunden. Um jetzt nun noch den Start und Anfangspunkt der Etappen visualisieren zu können, werden Marker auf einer eingängigen Koordinate gesetzt. Bei der JavaScript-Datei für die Startseite des Projekts werden Marker für alle Etappen gesetzt, sowie für den Zielpunkt. Die Anzahl der Marker wird, je nach Etappe, angepasst. Der letzte und zugleich nicht zu vergessene Schritt ist der Befehl die Etappen zu laden. Auf der Startseite soll, wie schon erläutert, ein Überblick über die Route gegeben werden, auf den Seiten der einzelnen Etappen jedoch soll auch nur die jeweilige Etappe angezeigt werden.

## **Quellenverzeichnis:**

Etappe 1: Innsbruck-Bozen. URL:

<http://www.gpsies.com/map.do;jsessionid=F7C0F3875B84096E5F116BF096D626C9.fe3?fileId=moknjyaypngqfre> (Stand: 20.06.2017)

Etappe 2: Bozen-Bormio. URL:

<http://www.gpsies.com/map.do;jsessionid=F7C0F3875B84096E5F116BF096D626C9.fe3?fileId=zukrsrkqfxbvlwjf> (Stand: 20.06.2017)

Etappe 3: Bormio-Cles. URL:

<http://www.gpsies.com/map.do;jsessionid=F7C0F3875B84096E5F116BF096D626C9.fe3?fileId=hpliajucmavsvdpw> (Stand: 20.06.2017)

Etappe 4: Cles-Riva del Garda. URL:

<http://www.gpsies.com/map.do;jsessionid=F7C0F3875B84096E5F116BF096D626C9.fe3?fileId=poxpgubcrvllqcc> (Stand: 20.06.2017)

Gardasee Panorama-bild. URL: <https://pixabay.com/de/gardasee-lago-di-garda-see-natur-2147692/> (Stand: 20.06.2017)

Innsbruck Panorama-Bild. URL: <https://pixabay.com/de/sommer-innsbruck-tirol-panorama-1528200/> (Stand: 20.06.2017)

Open Street Map (2017): OSM. URL: <https://www.openstreetmap.org/copyright> (Stand: 19.06.2017).

Stilfser Joch Passstraße URL: <https://pixabay.com/de/stilfser-joch-passstra%C3%9Fengebirgspass-69363/> (Stand: 20.06.2017)

Vinschgau Panorama-bild. URL: <https://pixabay.com/de/urlaub-italien-s%C3%BCdtirol-schenna-724451/> (Stand: 20.06.2017)

Plugins:

Leaflet.elevation: <https://github.com/MrMufflon/Leaflet.Elevation>

Leaflet-hash: <https://github.com/mlevans/leaflet-hash>

Leaflet-omnivore: <https://github.com/mapbox/leaflet-omnivore>