



# Projektbericht

*Outdoor Active: "Rafting Tirol"*

VU Geoinformatik: Web mapping

Förster Klaus

Öggl Bernhard

Peter Stampfl, Ayhan Kosan  
17.06.2020

---

## Inhaltsverzeichnis

1. Beschreibung des Projekts	2
2. Implementierungsschritte und eingesetzte Techniken "Outdoor Active"	2
2.1. Git & Visual Studio Code	2
2.2. HTML & CSS	3
2.3. Medien	13
2.4. Javascript	16
3. Quellen	23

---

## 1. Beschreibung des Projekts

Im Zuge des Projekts wurde ein Website für ein Raftingunternehmen entwickelt. Die Seite besteht aus einer Startseite, welche mit verschiedenen Medien (Bildern u. Videos) zum teilnehmen an einer Tour motivieren soll und den heutzutage üblichen Links zu Social Media im Footer. Auf der nächsten Seite befindet sich eine Karte auf welcher die Standorte der Touren, der Raftingbase und der Ein- und Ausstiege zu sehen sind. Des Weiteren bietet die Karte Informationen für den weiteren Aufenthalt der Gäste. So können Hotels, Restaurants, Bars, Clubs, Almhütten, Fahrradgeschäfte, Trinkwasserquellen, Picknickstellen und Fahrradtouren eingeblendet werden. Auf der letzten Seite befindet sich das Impressum und die Möglichkeit sich beim Newsletter anzumelden.

## 2. Implementierungsschritte und eingesetzte Techniken "Outdoor Active"




Die Implementierung des Projekts und die eingesetzten Techniken werden in den folgenden Punkten Git & Visual Studio Code, HTML, CSS, Medien und JavaScript beschrieben.

### 2.1 .Git & Visual Studio Code

Zur Bearbeitung und Austausch innerhalb des Projekts wurde die freie Software Git genutzt. Die Software ist Schnittstelle für verteilte Versionsverwaltung von Dokumenten für dieses Projekt.

.Git wurde in Kombination mit der freien Software Visual Studio Code genutzt. Visual Studio Code fungiert hier als Quelltext Editor. Über Erweiterungen in dem Programm wurde die Versionsverwaltung über Git. eingebunden.

Auf der Plattform "github.com" wurde das Projekt zusammengetragen und die Daten online verwaltet. Github bietet uns die Funktion der netzbasierten Versionsverwaltung und dient als Knotenpunkt für den Projektfortschritt.

3 people have access to this repository		
	csaw9279	Admin
	piet93	Admin
	iglatz	Read

Screenshot der Projektmitglieder auf <https://github.com/outdooractiv>.

### ***Herausforderungen bei der Nutzung von Github:***

Die einzige Herausforderung bei der Nutzung von Github war die Vergabe von Administrationsrechten innerhalb des Repositories um Pull und Push Befehle über Visual Studio Code zu gewähren.

Die Fehlermeldung: error 403 permission denied ist während des Projekts solange aufgetreten bis die Nutzerrechte angepasst worden sind.

## **2.2 .HTML & CSS**

### **Allgemein.**

#### ***Dokumente:***

Grundgerüst für die einzelnen Websites ist Hypertext Markup Language 5 (HTML5). Zusätzlich wurde Cascading Style Sheets (CSS) genutzt um Darstellungsvorgaben außerhalb der HTML Dateien festzulegen.

“Index.html” als dient als Startseite.

“Map.html” dient als Seite mit Fokus auf die Karten Applikation.

“Kontakt.html” dient als Kontaktseite.

“Main.css” für die Gestaltung der einzelnen Seiten.

### **Responsive:**

Das Dokument soll Inhalte und Elemente der Website automatisch an die Auflösung des Endgeräts anpassen. Durch Responsive Design wird mit dem Endgerät korrespondiert und auf die Bildschirmauflösung des Geräts reagiert.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

### **Schriften:**

Für die Website werden zwei verschiedene Schriftarten verwendet.

```
<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Karma">
<link
href="https://fonts.googleapis.com/css2?family=Merriweather&display=
swap" rel="stylesheet">
```

- Für den Fließtext wird die Sans-Serifen Schrift Karma verwendet.
- Für die Header Tags h2 und h3 wird die serifenbetonte Schrift Merriweather genutzt.
- Beide Schriftarten werden über die Api von Google Font eingebunden.
- Für eine gute Lesbarkeit wird die Schriftgröße für den Fließtext auf 18px gesetzt die der Überschriften auf 36px um sich von der normalen Schrift visuell gut abzusetzen.

### **Schriftstile:**

Verschiedene Font-Styles werden auf der Website umgesetzt um die Aufmerksamkeit des Nutzers durch fette und kursive Überschriften auf die Textblöcke zu ziehen.

## **Rafting Tirol**

*Rafting Tirol ist der erste und einzige Rafting Anbieter im Stubaital und im Großraum Innsbruck. Verschiedenste Touren stehen dir zur Auswahl: von gemütlich bis richtig wild, für Anfänger und Raftingprofis, Adrenalinjunkies und Familien - es ist wirklich für jeden etwas dabei. Detaillierte Informationen findest du in unserem [Newsletter](#).*

Screenshot der Textblöcke auf Index.html

Für den Fließtext wurden die Klassen `<div class="kursivp">` und `<div class="normalp">` definiert.

```
.kursivp p { font-style: italic; text-align: justify; margin: 1cm; }
.normalp p { font-style: normal; text-align: justify; margin: 1cm; }
```

- Der Fließtext ist über "justify" als Blocksatz definiert.
- Es gibt einen Rand von 1cm.

### **Text-Links:**

Es werden zwei Textlinks im Fließtext verwendet um den Nutzer der Website relevante Informationen über einen Klick zu bieten.

```
<a href="kontakt.html"> Newsletter.</a></p>
```

```
<a href="map.html"> hier</a> bist du richtig!</p>
```

### **Horizontale Navigationsleiste:**

Ermöglicht dem Endnutzer eine einfache Interaktion mit den einzelnen Seiten. Die unsortierten Listeneinträge verlinken die einzelnen Rubriken der Website.

Der Firmenname wird zudem über Stylesheet Elemente herausgehoben und in Fett dargestellt.

```
<div class="Navigationsleiste"> <ul>
<li style="font-weight: bold;"><a href="index.html">Rafting Tirol</a></li>
  <li><a href="map.html"> Unsere Touren</a></li>
  <li style="float:right"><a class="active"
href="kontakt.html">Kontakt</a></li> </ul> </div>
```

- Der "Firmenname" wird über einen style Eintrag fett dargestellt um diesen in der Navigationsleiste hervorzuheben.
- Das Hovern über den einzelnen Listeneinträgen färbt den Hintergrund schwarz ein um den Nutzer zu vermitteln welches Element angewählt wird.



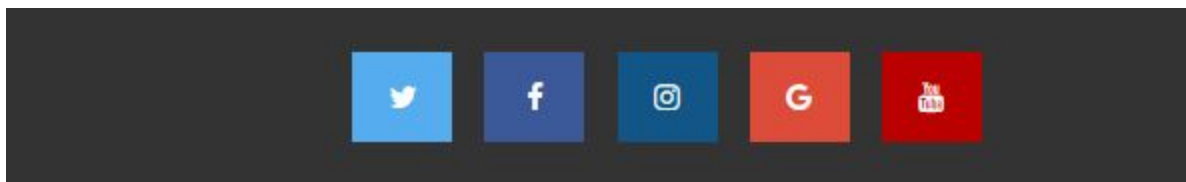
Screenshot der Navigationsleiste.

```
li a:hover:not(.active) { background-color: #111;}
```

- Die Navigationsleiste hat keinen Außen- und Innenabstand.
- Über width: 100% wird der komplette Anzeigebereich ausgefüllt.

```
.Navigationsleiste ul { list-style-type: none; margin: 0; padding: 0; overflow: hidden; background-color: #333; position: center; top: 0; width: 100%;}
```

### Footer:



Screenshot des Footers und der Social Media Icons.

Der Footer soll dem Nutzer die Möglichkeit geben "Rafting Tirol" auf Social Media Plattformen wieder zu finden und somit die Reichweite des Unternehmens zu erhöhen. Hierfür wurden Social Media Buttons verwendet die über die Font-Awesome Api eingebunden werden.

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

- Der Footer nutzt wie die Navigationsleiste die gesamte Bildschirmbreite.

```
.footer { position: center; left: 0; bottom: 0; width: 100%;
background-color: #333; color: white; text-align: center;
font-size: 15px;padding: 3px; margin: 5px;}
```

- Die Social media "buttons" werden über einen Anchor Tag implementiert.

```
<a href="https://google.com/RaftingTirol" class="fa
fa-google"></a>
```

- Über die Klasse .fa des Anchor Tags wurde jede einzelne Social Media Plattform in den styles integriert.

```
.fa-google { background: #dd4b39; color: white;}
```

## Index.html.

### Video:

Ein Video wurde eingebunden um die Aufmerksamkeit des Nutzers auf die Angebote der Firma zu lenken. Die Nutzung verschiedener Medienformate wird hier angewendet um die Nutzer Erfahrung attraktiver zu gestalten.

```
<div class="video"><video width="800" controls><source
src="videos\videorafting.mp4" type="video/mp4">
```



Screenshot des Videoplayers auf Index.html.

- Das Videoplayer hat eine feste Breite von 800.
- In dem Stylesheet Dokument wird u.a. "display flex und align items" genutzt um den Videoplayer mittig auf der Weissfläche der Webseite zwischen zwei Headern zu



platzieren.

```
.video { margin: auto; padding: center; display: flex; flex-direction: column; align-items: center;}
```

### Bildergalerie:



Screenshot der Bildergalerie auf der Startseite.

Die Bildergalerie bringt dem Nutzer Inhalte anhand von Bildern näher und gibt einen Eindruck zu den angebotenen Dienstleistungen von "Rafting Tirol".

```
<div class="responsive">
  <div class="gallery">
    <a target="_blank" href="images/imgh.jpg">

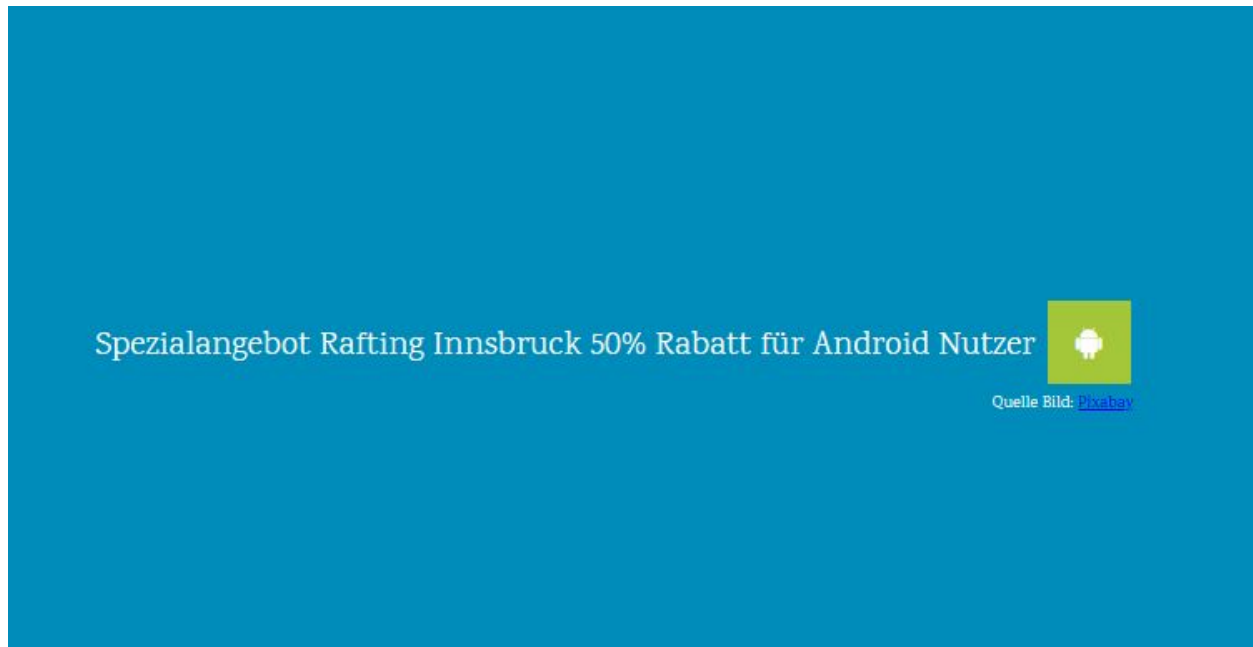
</a>
    <!-- <div class="desc">Add a description of the image here</div> -->
  </div>
</div>
```

- Die einzelnen Bilder reagieren auf verschiedene Auflösungen. Über die Klasse `.responsive` wird definiert wie bei einzelnen Auflösungsstufen Bilder angezeigt werden

```
.responsive { padding: 0 6px; float: left; width: 24.99999%;}
@media only screen and (max-width: 700px) { .responsive { width: 49.99999%; margin: 6px 0; }}
@media only screen and (max-width: 500px) { .responsive { width: 100%; }}
}}
```

- Vier Bilder werden bei einer Bildschirmgröße von mehr als 700px angezeigt.
- Zwei Bilder werden bei einer Bildschirmgröße von 700px und weniger angezeigt.
- Ist die Bildschirmgröße kleiner als 500px werden die Bilder vertikal gestapelt dargestellt.

## Bild mit Overlay:



Screenshot des Bildes mit einem Textoverlay.

Um Dynamik zu erzeugen und dem Endnutzer mehr Interaktionsmöglichkeiten zu bieten wir dem Bild bei der Berührung mit einem Cursor eine einfarbige Fläche mit Text eingeblendet. Hier lässt sich ein Link zu einer Social Media Plattform klicken.

```
<div class="container">
  
<div class="overlay">
```

- Der DIV-Container für das Bild mit style Anweisungen damit das Bild mittig am auf dem Bildschirm ausgerichtet wird.
- Sobald der DIV-Container berührt wird die DIV-Klasse "overlay" angesprochen und ein Overlay legt sich über das Bild.

```
.container:hover .overlay { height: 100%;}
.overlay { position: absolute; bottom: 0; left: 0; right: 0;
background-color: #008CBA; overflow: hidden; width: 100%; height:
auto; transition: .5s ease;}
```

- In dem Overlay wird über die Klasse "text" Fließtext definiert.
- Der Text hat einen Anchor-Tag mit einem Link zu einer Social Media Plattform

```
<div class="text">Spezialangebot Rafting Innsbruck 50% Rabatt für
```

```
Android Nutzer <a href="android.com/RaftingTirol" class="fa
fa-android"></a>
```

- In dem Container ist zusätzlich eine Bildquelle über die DIV-id "flipquelle" zu dem verwendeten Bild implementiert.

```
div id="flipquelle"> Quelle Bild: <a
href="https://pixabay.com/de/photos/innsbruck-%C3%B6sterreich-winter
-schnee-5278879/">Pixabay</a> </div>
```

## Map.html:

Die Map.html Seite fokussiert sich auf die Übersichtskarte von "Rafting Tirol".

```
<div id="map"></div>
#map { width: 100%; height: 900px; border: 1px solid gray; text-align:
left;}
```

- Die Map.html gibt der Karte genügend Weißraum um die Aufmerksamkeit des Nutzers alleine auf die Karte zu ziehen.
- Neben der Karte ist ein Bild mit Bildquelle eingebunden um die Seite attraktiver zu gestalten.

```
<div style="text-align: right;">
  
  <div id="quelle">Quelle Bild: <a
href="https://pixabay.com/de/photos/rafting-fluss-wasser-sport-1377057/">P
ixabay</a> </div>
</div>
```

## Kontakt.html:

### Subscribe Form:

**Newsletter-Anmeldung**

Name

Email address

☒ Wöchentlicher Newsletter

Subscribe

Screenshot der Subscribe Form auf der Kontaktseite.

Auf der Kontaktseite ist ein Formular für die Anmeldung zum Newsletter ausgeführt.

- Das Formular verarbeitet den User-Input über eine `<form>` PHP-Formular.

```
<form action="/action_page.php">
```

- Der Input wird in den beiden Textfeldern für Name und E-Mail Adresse eingetragen.
- Durch den tag "required" muss der Text eingegeben werden bevor der Subscribe Button geklickt werden kann

```
<input type="text" placeholder="Name" name="name" required>
    <input type="text" placeholder="Email address"
name="mail" required>
```

- Zusätzlich zum Formular kann die Option eines Wöchentlichen Newsletters ausgewählt werden.

```
<label><input type="checkbox" checked="checked"
name="subscribe"> Wöchentlicher Newsletter</label>
```

- Die Gestaltung der Newsletter Elemente und den Subscribe button wird über Selektoren

festgelegt

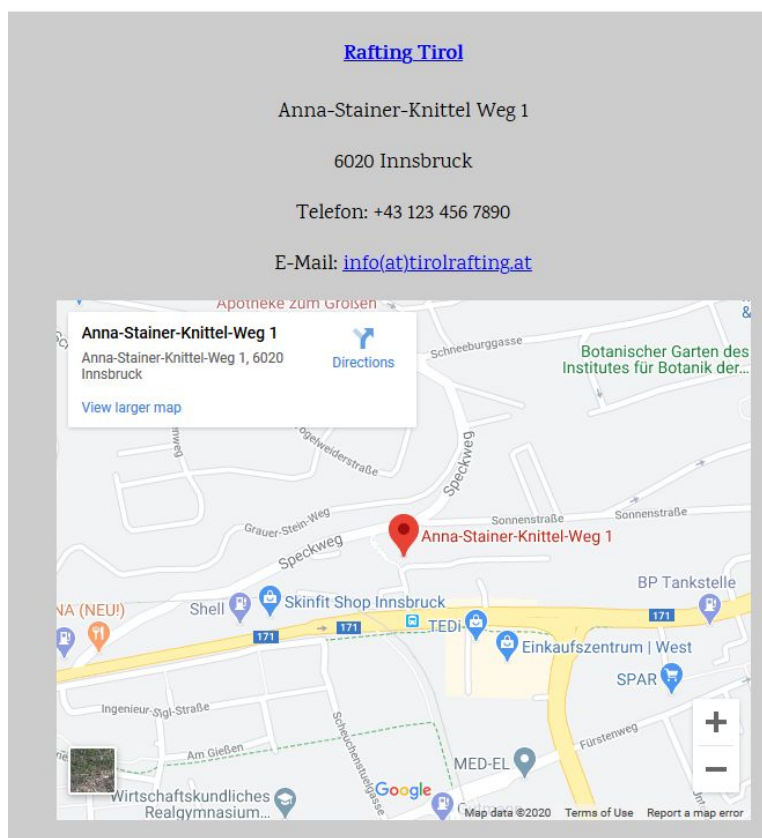
```
input[type=text],input[type=submit] { width: 100%; padding: 12px; margin: 8px 0; display: inline-block; border: 1px solid #ccc; box-sizing: border-box;}
```

- Der Submit button folgendermaßen definiert, und wird bei der Berührung von einem Cursor über "opacity" etwas transparenter.

```
input[type=submit] { background-color: #4CAF50; color: white; border: none;}

input[type=submit]:hover { opacity: 0.8;}
```

### Impressum:



Screenshot des Impressums mit der eingebetteten Karte.

Um die E-Mail Kommunikation mit Nutzern der Website zu ermöglichen wurde ein Mailto im href-Attribut hinterlegt im Fließtext bei "E-Mail:" verwendet.

```
<p>E-Mail: <a href="mailto:info@raftingtirol.at">info(at)tirolrafting.at</a></p>
```

- Wird der Link geklickt öffnet sich das E-Mail Programm des Benutzers automatisch damit dieser eine E-Mail an die festgelegte Mail-Adresse senden kann.

Um den Standort von "Rafting Tirol" mit Benutzern zu teilen ist eine Google Maps Karte über ein iframe eingebunden.

```
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2707.4874840038847!2d11.369955715618762!3d47.26572437916357!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x479d6b1e0eb34857%3A0x56857f26f17a49cb!2sAnna-Stainer-Knittel-Weg%20!2C%206020%20Innsbruck!5e0!3m2!1sen!2sat!4v1592347436556!5m2!1sen!2sat" width="600" height="450" frameborder="0" style="border:0;" allowfullscreen="" aria-hidden="false" tabindex="0"></iframe>
```

## 2.3 .Medien

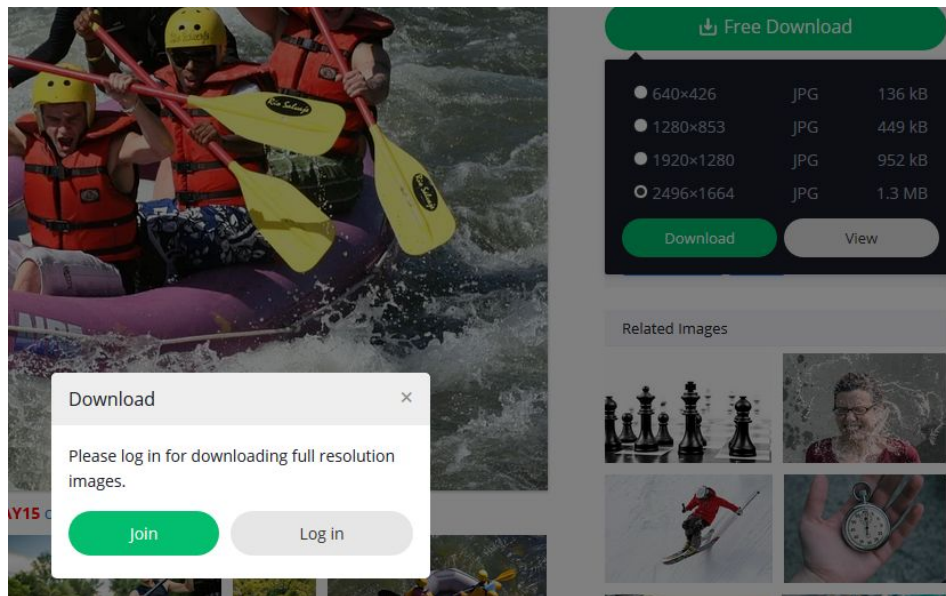
Alle eingebundenen Medien stehen der freien kommerziellen Nutzung zur Verfügung und benötigen keinen Bildnachweis. Bezogen wurden die Bilder auf der Webseite "pixabay.com".

Alle anderen Bilder und Videos, welche auf der Webseite eingebunden worden sind, haben keine urheberrechtlichen Einschränkungen, da diese eigenes geistliches Eigentum sind.

### **Pixabay:**

Auf der Website Pixabay werden verschiedene Medienformate zur freien Nutzung angeboten. Je nach Auflösung der Mediendateien ist der download der Daten kostenfrei. Ab einer bestimmten Größe der Medien ist die Nutzung eines Accounts der Webseite nötig.





Ausschnitt von Pixabay.

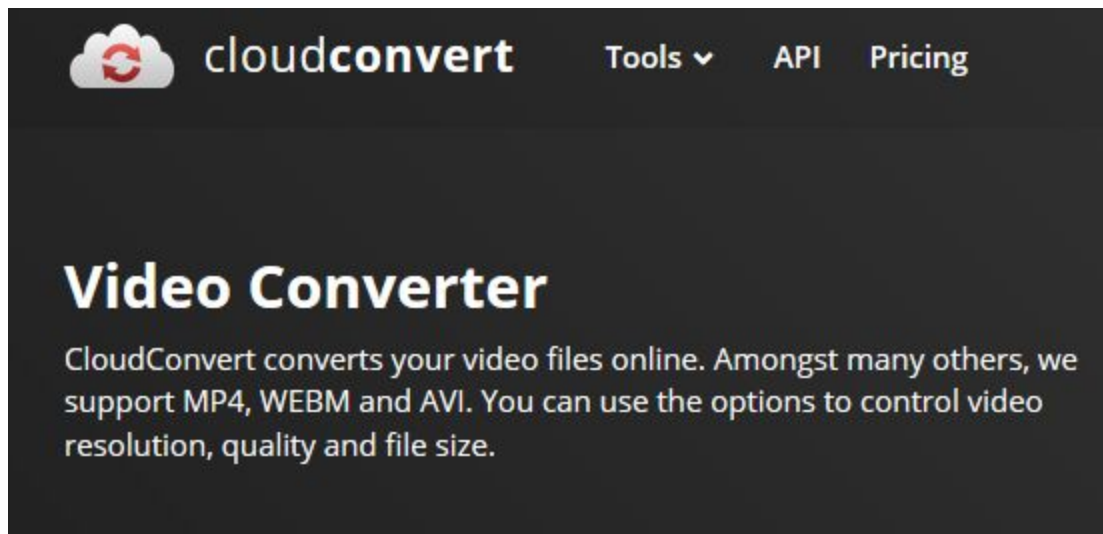
## Gimp:

Gimp ist eine freie Open Source Bildverarbeitungssoftware für Rasterdaten. Im Rahmen des Projekts wurde Bild genutzt um folgende Operationen durchzuführen:

- zuschneiden von Grafiken
- Exportieren der Grafiken in das JPG Format
- Weißabgleich der Bilder



Screenshot während des zuschneiden der Bilder in dem Programm Gimp.

**Cloudconvert:**

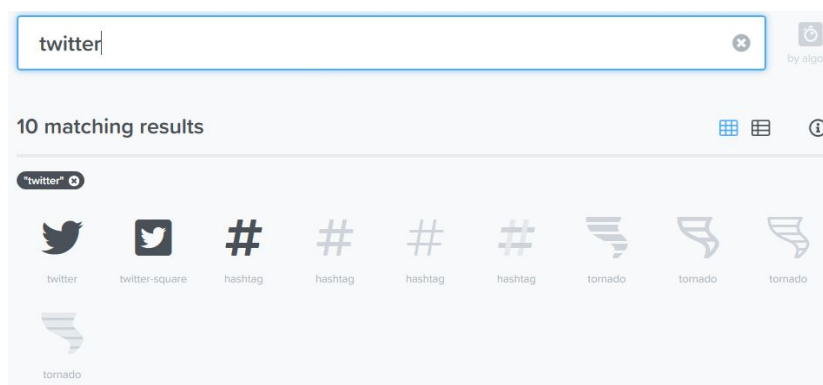
Ausschnitt aus der Website Cloudconvert.com.

Cloudconvert ist ein Online Datei Konverter. Die Nutzung der Website ist kostenfrei bei 25 und weniger Konvertierungen am Tag. Im Rahmen des Projekts wurde die Website genutzt um das Video auf der Startseite vom mov. Format in das mp4. Format zu konvertieren.

- Nicht jedes Videoformat wird durch gängige Browser unterstützt deswegen war eine Konvertierung der Datei nötig.
- Das mp4. Datenformat wird auf allen gängigen Browsern unterstützt.

**Font Awesome:**

Font-Awesome ist einer der größten Anbieter von verschiedenen Schriftstilen im Internet. Die Nutzung der Inhalte ist zum größten Teil kostenfrei. Im Rahmen des Projekts wurde Font-Awesome für Social Media Icons verwendet.



Screenshot von Icons auf der Font-Awesome Page.



### Google Fonts:

Über die Api von Google Fonts können Schriften von Nutzern kostenfrei genutzt werden. Im Rahmen des Projekts wurden die Schriftarten Merriweather und Karma genutzt. Die Nutzung der Schriftarten ist kostenfrei.

## 2.4 .JavaScript

JavaScript wurde verwendet um die interaktive Kartenapplikation zu erstellen. Als Basis diente die open-source JS-library "Leaflet", welche alle benötigten Funktionen zur Verfügung stellte. Zu allererster wurde der Startbasemaplayer erstellt. Es wurde dafür der Layer "OpenStreetMap.Mapnik" verwendet.

```
let startlayer = L.tileLayer.provider("OpenStreetMap.Mapnik");
```

Um diesen auch sichtbar zu machen wurde im folgenden Schritt die Karte erstellt und um zwei, in einem Kontrollmenü auszuwählende Baselayer, einen Satellitenbaselayer und einen topographischen Layer, ergänzt. Das Zentrum der Karte wurde so gewählt, dass die Layer, (Raftingtouren) welche von Beginn an sichtbar sind, zu sehen sind.

```
// map
let map = L.map("map", {
  center: [47.181075, 11.377461],
  zoom: 12,
  layers: [
    startlayer
  ]
});
L.control.layers({
  "TopoMap": L.tileLayer.provider("OpenTopoMap"),
  "OpenStreetMap": L.tileLayer.provider("OpenStreetMap.Mapnik"),
  "Satellit": L.tileLayer.provider("Esri.WorldImagery")
})
```

Die geeigneten Baselayer wurden mit Hilfe der Seite <https://leaflet-extras.github.io/leaflet-providers/preview/> gesucht. Die Seite bietet eine Vorschau der verschiedenen Baselayer, welche über Leaflet eingebunden werden können.

Der nächste Schritt bestand im Herunterladen des Openstreetmap Österreich Datensatzes im Format .shp. Da dieser sehr umfangreich ist, wurde mit dem Programm QGIS die benötigten Daten herausgefiltert und als .geojson exportiert.

The OpenStreetMap data files provided on this server do **not** contain the user names, user IDs and changeset IDs of the OSM objects because these fields are assumed to contain personal information about the OpenStreetMap contributors and are therefore subject to data protection regulations in the European Union. Extracts with full metadata are available to OpenStreetMap contributors only.

#### Commonly Used Formats

- **europe-latest.osm.pbf**, suitable for Osmium, Osmosis, impotom, osm2pgsql, mkgmap, and others. This file was last modified 8 hours ago and contains all OSM data up to 2020-06-16T20:59:02Z. File size: 21.6 GB; MD5 sum: f818b86236d155e2f09898696c6381.
- **europe-latest-free.shp.zip** is not available for this region; try one of the sub-regions.

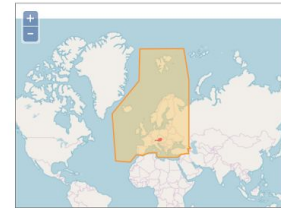
#### Other Formats and Auxiliary Files

- **europe-latest.osm.bz2**, yields OSM XML when decompressed; use for programs that cannot process the .pbf format. This file was last modified 2 hours ago. File size: 34.9 GB; MD5 sum: 6ddc54ec0f1416e58caa04e1b72327b8.
- **europe-internal.osm.pbf** The history file contains personal data and is available on the [internal server](#) only. See notice above for further information.
- **.poly** file that describes the extent of this region.
- **.osm.os** files that contain all changes in this region, suitable e.g. for Osmosis updates
- **raw\_directory\_index** allowing you to see and download older files

#### Sub Regions

Click on the region name to see the overview page for that region, or select one of the file extension links for quick access.

Sub Region	Quick Links		
	.osm.pbf	.shp.zip	.osm.bz2
Albania	[.osm.pbf] (40.6 MB)	[.shp.zip]	[.osm.bz2]
Andorra	[.osm.pbf] (1.6 MB)	[.shp.zip]	[.osm.bz2]
Austria	[.osm.pbf] (583 MB)	[.shp.zip]	[.osm.bz2]
Azores	[.osm.pbf] (9.9 MB)	[.shp.zip]	[.osm.bz2]
Belarus	[.osm.pbf] (209 MB)	[.shp.zip]	[.osm.bz2]
Belgium	[.osm.pbf] (366 MB)	[.shp.zip]	[.osm.bz2]
Bosnia-Herzegovina	[.osm.pbf] (86 MB)	[.shp.zip]	[.osm.bz2]
Bulgaria	[.osm.pbf] (98 MB)	[.shp.zip]	[.osm.bz2]
Croatia	[.osm.pbf] (119 MB)	[.shp.zip]	[.osm.bz2]
Cyprus	[.osm.pbf] (15.3 MB)	[.shp.zip]	[.osm.bz2]
Czech Republic	[.osm.pbf] (716 MB)	[.shp.zip]	[.osm.bz2]
Denmark	[.osm.pbf] (347 MB)	[.shp.zip]	[.osm.bz2]
Estonia	[.osm.pbf] (80 MB)	[.shp.zip]	[.osm.bz2]
Faroe Islands	[.osm.pbf] (3.5 MB)	[.shp.zip]	[.osm.bz2]
Finland	[.osm.pbf] (524 MB)	[.shp.zip]	[.osm.bz2]
France	[.osm.pbf] (3.6 GB)	✗	[.osm.bz2]
Georgia (Eastern Europe)	[.osm.pbf] (47.1 MB)	[.shp.zip]	[.osm.bz2]
Germany	[.osm.pbf] (3.1 GB)	✗	[.osm.bz2]
Great Britain	[.osm.pbf] (1.1 GB)	✗	[.osm.bz2]
Greece	[.osm.pbf] (195 MB)	[.shp.zip]	[.osm.bz2]
Hungary	[.osm.pbf] (168 MB)	[.shp.zip]	[.osm.bz2]
Iceland	[.osm.pbf] (41.3 MB)	[.shp.zip]	[.osm.bz2]
Ireland and Northern Ireland	[.osm.pbf] (175 MB)	[.shp.zip]	[.osm.bz2]
Isle of Man	[.osm.pbf] (2.3 MB)	[.shp.zip]	[.osm.bz2]
Italy	[.osm.pbf] (1.4 GB)	✗	[.osm.bz2]

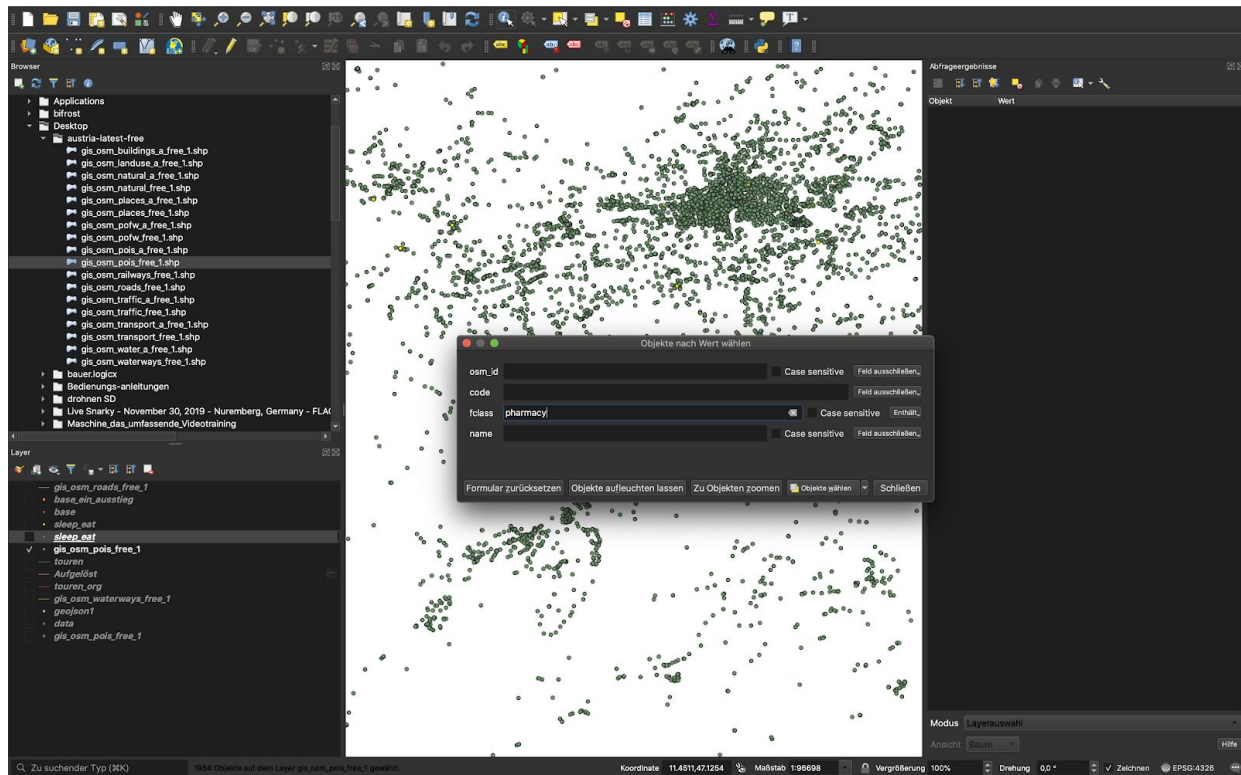


Not what you were looking for? Geofabrik is a consulting and software development firm based in Karlsruhe, Germany specializing in OpenStreetMap services. We're happy to help you with data preparation, processing, server setup and the like. [Check out our web site](#) and contact us if we can be of service.

Nicht das Richtige dabei? Die Geofabrik ist ein auf OpenStreetMap spezialisiertes Beratungs- und Softwareentwicklungsunternehmen in Karlsruhe. Gern helfen wir Ihnen bei der Datenaufbereitung, Datenkonvertierung, Serverinstallation und ähnlichen Aufgaben. Besuchen Sie unsere Webseite und sprechen Sie mit uns, wenn wir Ihnen helfen können.

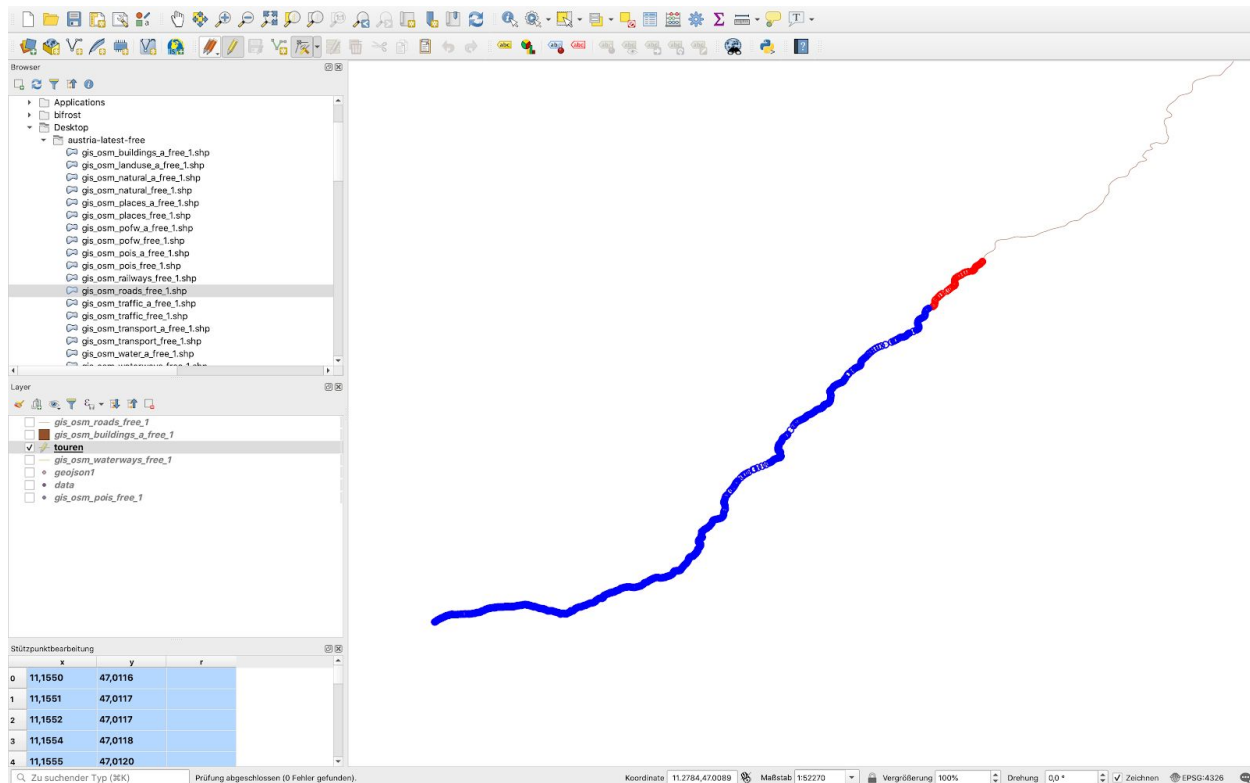
Die Website auf welcher der OSM-Datensatz heruntergeladen wurde.

Quelle: <http://download.geofabrik.de/europe.html>



Das herausfiltern der benötigten Daten aus dem OSM-Datensatz mit QGIS.

Der `touren.js` Datensatz, wurde aus dem OSM Flussdatensatz erzeugt. Dafür wurden die zwei benötigten Flüsse die Ruetz und der Inn herausgefiltert und in einem neuen Layer exportiert. Da die Touren aber nicht entlang des ganzen Flusses verlaufen, wurden die Liniendatensätze so bearbeitet, dass nur mehr die Strecken übrig blieben, welche auf geraftet werden. Zum Schluss wurde noch die Attributtabelle angepasst.



Das Bearbeiten des Flusslayers um daraus den Tourenlayer zu erhalten.

Der `base_ein_ausstieg.js`-Datensatz wurde in Qgis selbst erstellt. Die Koordinaten für die wenigen benötigten Punkte, wurden auf Tiris (<https://www.tirol.gv.at/statistik-budget/tiris/>) gesucht, per Hand in einem Punktlayer eingetragen und die Attributtabelle angepasst.

Darauf wurden die Datensätze in die Karte implementiert. Die Touren, die Raftingbase und die Ein- und Ausstiege, wurden als fixe Layer implementiert, welche vom Nutzer nicht ein und ausgeschaltet werden können. Die restlichen Layer sind optionale Layer, welche vom Nutzer erst aktiviert werden müssen. Da alle Datensätze im Format `.geojson` vorhanden sind, wurde auf die Leaflet-Funktion `geoJson()` zurückgegriffen.

```
L.geoJSON(TOUREN, {
  style: function (geojsonFeature) {
    return {
      weight: 8,
      color: "darkblue"
    }
  }
})
```

```

    }

    },
    onEachFeature: function (feature, layer) {
        //console.log("tourenFeature", feature);
        //console.log("tourenLayer", layer);
        layer.bindPopup(`

#### 


```

Bei den Touren wurde mit der Option Style die dicke der Linie und die Farbe verändert. Die onEachFeature Option eignet sich sehr gut um eine Funktion an jedem Feature eines Datensatzes, wie das einbinden eines Popups, welches auf die Attributtabelle zurückgreift, auszuführen. Wie bei jedem Datensatz wurde auch hier implementiert, dass das Popup eines Features bei einem Mouseover aktiviert und bei einem Mouseout deaktiviert wird.

Der nächste Schritt war das implementieren der Raftingbase und der Ein- und Ausstiege. Dafür mussten erst geeignete Icons für die Marker gefunden werden. Diese wurden auf der Seite "mapicons.mapsmarker.com" gefunden. Da sich die Features alle im gleichen Datensatz befinden, jedoch alle verschiedene Icons benötigen wurde mit eine if-Abfrage der Typ abgefragt und das richtige Icon zugewiesen.

```

let homeIcon = L.icon({
    iconUrl: 'icon/home.png',
    iconSize: [52, 57],
    iconAnchor: [26, 57],
    popupAnchor: [0, -57]
});
if (point.properties.type === "base") {
    //console.log("start");
    let marker = L.marker(latlng, {
        riseOnHover: true,
        icon: homeIcon
    });
    marker.bindPopup(`

#### 


```

Um die zuvor aus OSM geladenen "Points of Interests" (POIs) zu implementieren wurde eine Funktion geschrieben, da diese sich in verschiedenen Datensätzen befinden und alle verschiedene Icons benötigen.

```
displayPoi(data = NIGHT, icon1 = "bar_cocktail", icon2 = "dancinghall", poi =
overlay.party, fclass1 = "pub", fclass2 = "nightclub");
```

Als Input nimmt die Funktion den Datensatz, den String für den Namen des ersten und zweiten Icons, den Overlay (optionaler Layer), zu welchen der Datensatz hinzugefügt wird und den ersten und zweiten String nach welchem im Datensatz gesucht werden soll.

Die Funktion besteht:

- aus einer for-Schleife, welche durch den Datensatz (Input: data) iteriert
- mit einer if-else if-Abfrage wird nach den Strings gesucht welche als Inputs angegeben wurden (Input: fclass1; fclass2)
- das Icon festgelegt und angepasst (Input: icon1; icon2)
- der Marker mit dem im Datensatz stehenden Koordinaten (latlng) und dem zuvor erstellten Icon generiert
- mit einem Popup (Namen, Featureclass) versehen
  - da nicht alle POIs einen Namen besitzen, wird mit einer if-Abfrage geschaut ob dieser auch vorhanden ist; ist das nicht der Fall scheint im Popup nur die F-class auf
- zum Schluss zum Overlay hinzugefügt (Input: poi)

```
function displayPoi(data, icon1, icon2, poi, fclass1, fclass2) {
  for (let i = 0; i < data.features.length; i++) {
    const element = data.features[i];

    if (element.properties.fclass === fclass1) {
      L.geoJSON(element, {
        pointToLayer: function (point, latlng) {
          //console.log("Point", point.properties.fclass);
          //console.log("latlng", latlng);
          let myIcon = L.icon({
            iconUrl: `icon/${icon1}.png`,
            iconAnchor: [16, 37],
            popupAnchor: [0, -37]
          });
          let marker = L.marker(latlng, {
            icon: myIcon
          });
          //console.log(point.properties.name === null)
          if (point.properties.name !== null) {
            marker.bindPopup(`<h4>${point.properties.name}
            (${point.properties.fclass})</h4>`);
          } else {
```

```

        marker.bindPopup('<h4>${point.properties.fclass}</h4>');
    }
    marker.on('mouseover', function (e) {
        marker.openPopup();
    });
    marker.on('mouseout', function (e) {
        marker.closePopup();
    });
    return marker
}
}).addTo(poi);

```

Damit die Features auch in der "Controllbar" angezeigt werden, müssen sie erst erzeugt und hinzugefügt werden.

```

let overlay = {
    sleep: L.featureGroup(),

    "Schlafen": overlay.sleep,
}).addTo(map);

```

Der letzte Schritt im Skript war das implementieren der Fahrradrouten. Diese wurden als frei verfügbarer Shapefile-Datensatz von Tiris heruntergeladen und in Qgis zu .geojson umgewandelt.

**Open Government Data - Tirol/tiris**

**Radrouten Tirol**  
 Letzte Aktualisierung vor 4 Monaten | 673 Datensätze

Übersicht | **Daten** | API-Explorer

Herunterladen | APIs

Der Datensatz beinhaltet freigegebene Mountainbikerouten, überregionale Radwandererouten, Radwege und freigegebene Singletrail-Strecken in Tirol.

[Link zu den Metadaten](#)  
 Mehr ▾

**Attribute**  
 Diagramm • Kartenvisualisierung

<b>BEZIRK_REGION</b> Text	<b>CREATETIMESTAMP</b> Datum oder Uhrzeit	<b>FAHRZEIT</b> Text	<b>GEMEINDE_NAME</b> Text	<b>HM_BERGAB</b> Zahl	<b>HM_BERGAUF</b> Zahl
<b>HOEHE_START</b> Zahl	<b>HOEHE_ZIEL</b> Zahl	<b>LAENGE_HAUPTROUTE_KM</b> Zahl	<b>OBJEKT</b> Text	<b>ROUTEN_TYP</b> Text	

**Informationen**  
 opendata  
 Freigegeben von: tiris.ogd  
 Datenquelle: services3.arcgis.com  
 Metadaten anzeigen  
 Webkarte erstellen  
 Eine Story-Map erstellen

Die Webseite von welcher der Fahrradroutendatensatz heruntergeladen wurde.

Quelle:

<https://data-tiris.opendata.arcgis.com/datasets/radrouten-tirol?geometry=11.245%2C47.198%2C11.466%2C47.238>

Bei der Implementierung wurde darauf geachtet, dass die verschiedenen Schwierigkeiten der Routen in verschiedenen Farben dargestellt werden. Die Popups enthalten den Namen, den Typ, die Fahrzeit und eine kurze Beschreibung.

```
if (geojsonFeature.properties.SCHWIERIGK === "leicht") {  
    return {  
        weight: 3,  
        color: "green"  
    }  
} else if (geojsonFeature.properties.SCHWIERIGK === "schwierig") {  
    return {  
        weight: 3,  
        color: "red"  
    }  
}
```

### 3. QUELLENANGABEN

#### Datensätze:

1. Verändert aus OSM Austria Datensatz (night.js, shopping.js, sleep\_eat.js, touren.js, water\_picnic.js) <http://download.geofabrik.de/europe.html>. Zuletzt aufgerufen am 17.06.2020.
2. radrouten\_tirol.js  
<https://data.tiris.opendata.arcgis.com/datasets/radrouten-tirol?geometry=11.245%2C47.198%2C11.466%2C47.238>. Zuletzt aufgerufen am 17.06.2020.

#### Grafiken Pixabay:

1. Grafik "header.jpg":  
<https://pixabay.com/de/photos/innsbruck-%C3%B6sterreich-winter-schnee-5278879/>.  
Zuletzt aufgerufen am 17.06.2020.
2. Grafik "RaftingInnsbruck.jpg":  
<https://pixabay.com/de/photos/innsbruck-%C3%B6sterreich-winter-schnee-5278879/>.  
Zuletzt aufgerufen am 17.06.2020.
3. Grafik "mapimage.jpg":  
<https://pixabay.com/de/photos/rafting-fluss-wasser-sport-1377057/>. Zuletzt aufgerufen am 17.06.2020.

#### Grafiken und Videos Peter Stampfl:

1. Grafik "imga.jpg": Peter Stampfl
2. Grafik "imgb.jpg": Peter Stampfl
3. Grafik "imgc.jpg": Peter Stampfl
4. Grafik "imgd.jpg": Peter Stampfl
5. Grafik "imge.jpg": Peter Stampfl
6. Grafik "imgf.jpg": Peter Stampfl
7. Grafik "imgg.jpg": Peter Stampfl
8. Grafik "imggh.jpg": Peter Stampfl
9. Video "videorafting.mp4": Peter Stampfl

#### Social Media Buttons:

1. <https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css>



zuletzt aufgerufen am 17.06.2020

2. <https://fontawesome.com/icons?d=gallery>. Zuletzt aufgerufen am 17.06.2020.

### Map Icons

1. <https://mapicons.mapsmarker.com> zuletzt aufgerufen am 17.06.2020

### Übersichtskarte Impressum:

1. Embedded Map: <https://www.google.com/maps>. Zuletzt aufgerufen am 17.06.2020.

### Schriftarten:

1. Schriftarten: <https://fonts.google.com/>. Zuletzt aufgerufen am 17.06.2020.
2. Schriftart Karma: <https://fonts.googleapis.com/css?family=Karma>. Zuletzt aufgerufen am 17.06.2020.
3. Schriftart Merriweather: <https://fonts.googleapis.com/css2?family=Merriweather&display=swap>. Zuletzt aufgerufen am 17.06.2020.

### Literatur:

Mansfield, R. (2005). *CSS Web Design For Dummies* (1. Aufl.). USA: For Dummies.

<https://www.codecademy.com/learn/learn-html>. Zuletzt aufgerufen am 17.06.2020.

<https://css-tricks.com/snippets/css/complete-guide-grid/>. Zuletzt aufgerufen am 17.06.2020.

<https://www.w3schools.com/html/>. Zuletzt aufgerufen am 17.06.2020.

<https://web.stanford.edu/class/cs142/lectures/CSS.pdf>. Zuletzt aufgerufen am 17.06.2020.

<https://web.stanford.edu/class/cs142/lectures/HTML.pdf>. Zuletzt aufgerufen am 17.06.2020.

<https://web.stanford.edu/class/cs142/lectures/URLs.pdf>. Zuletzt aufgerufen am 17.06.2020.

<https://web.stanford.edu/class/cs142/lectures/JavaScriptBasics.pdf>. Zuletzt aufgerufen am 17.06.2020.

<https://web.stanford.edu/class/cs142/lectures/RWD.pdf>. Zuletzt aufgerufen am 17.06.2020.

<https://stackoverflow.com/>. Zuletzt aufgerufen am 17.06.2020.

<https://leafletjs.com/reference-1.6.0.html> Zuletzt aufgerufen am 17.06.2020.