

Institut für Geographie  
716409 VU Geoinformatik: Web Mapping  
Dozenten: Förster Klaus & Mag. Öggl Bernhard

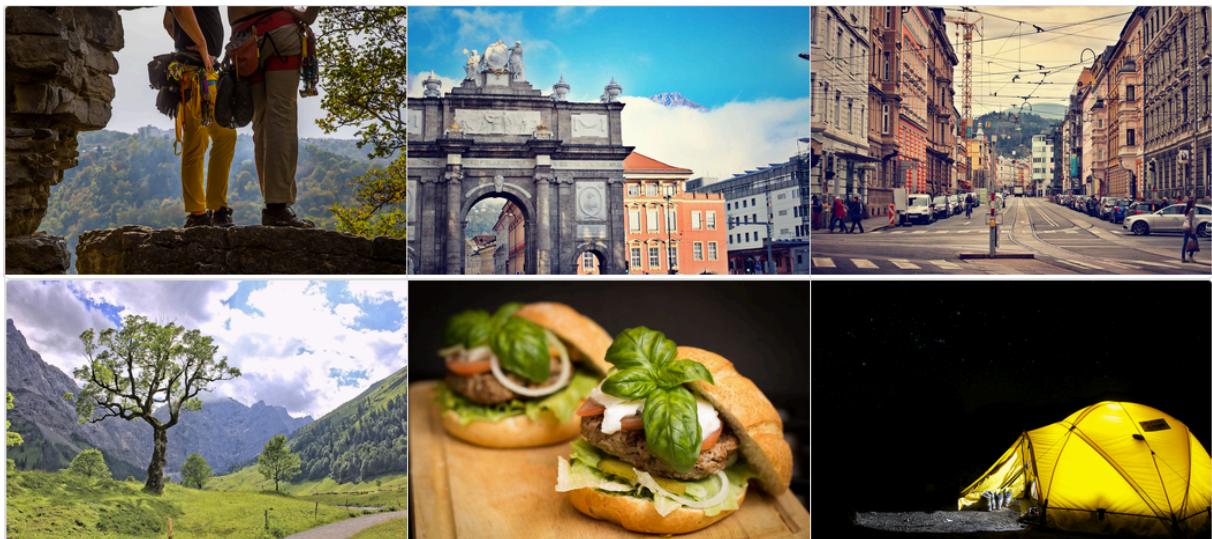


Sommersemester 2018

13.06.2018

# Entdecke Innsbruck

*Ein kleiner Reiseführer für eine abwechslungsreiche Woche im Herzen Tirols*



## Projektbericht zur Gestaltung der Website

von

Balling, Jessica

Heuser, Jacob

Weiß, Marlene

11708836

01218423

11711423

jessica.balling@student.uibk.ac.at jacob.heuser@student.uibk.ac.at marlene.weiss@student.uibk.ac.at

## Inhalt

1. Grundidee und Aufbau der Website .....	1
2. Umsetzung .....	2
2.1 Das Quiz .....	2
2.1.1 Umsetzung auf der html-Seite .....	2
2.1.3 Umsetzung der JavaScript-Seite.....	2
2.2 Bootstrapmodule im Quiz, auf der Start- und Sportseite .....	2
2.3 Kartenapplikation, Plugins, Marker und Tracks auf der Unterseite Wandern .....	3
2.3.1 Einbinden von Textpassagen und externen Inhalten .....	4
2.3.2 Anlegen eines Kartenfeldes mit Layern.....	4
2.3.3 Einbinden eines gpx-Files über ein Plugin .....	5
2.3.4 Einbinden eines Höhenprofil-Plug-Ins.....	7
2.3.5 Hinzufügen von Kartenelementen: Fullscreen-Plug-In, Maßstabsleiste, Kontrollfeld .....	8
2.4 Kartenapplikation, Plug-Ins und Marker auf den Seiten Kultur, Klettern, Hunger?, ÖPV .....	9
2.5 Cascade Style Sheets .....	10
3. Herausforderungen und Abweichungen .....	16
4. Quellenverzeichnis.....	17

## Abbildungsverzeichnis

Abbildung 1: JavaScript Code zum Erstellen des Quiz.....	2
Abbildung 2: Bootstrap-Modul auf der Startseite.....	3
Abbildung 3: Einbinden der Bootstrap-Module im Header .....	3
Abbildung 4: Einbindung des ZAMG-Codes in die wandern.html-Datei .....	4
Abbildung 5: Output auf der wandern.html-Seite .....	4
Abbildung 6: Anlegen einer mapdiv für die Karte .....	4
Abbildung 7: Anlegen der Hintergrundlayer in der wandern.js-Datei .....	5
Abbildung 8: Gruppieren der Layer Orthofoto und Nomenklatur .....	5
Abbildung 9: Einbindung des GPX-Plug-Ins in die wandern.html-Datei.....	5
Abbildung 10: Einbindung des GPX-Tracks in die wandern.js-Datei .....	5
Abbildung 11: Einbindung von Markern für die Start- und Endpunkte des Tracks mit passenden Icons6	

Abbildung 12: Gruppieren der Variablen des GPX-Tracks mit Start- und Endpunkt des Tracks .....	6
Abbildung 13: Ändern der Default-Farbe des GPX-Tracks .....	6
Abbildung 14: Ausgabe der Änderung auf der wandern.html-Seite .....	6
Abbildung 15: Einbinden des Höhenprofil Plug-Ins in die wandern.js-Datei .....	7
Abbildung 16: Icon zum Aufrufen des Höhenprofils .....	7
Abbildung 17: Einbindung des Fullscreen Plug-Ins in die wandern.html Datei.....	8
Abbildung 18: Einbindung des Fullscreen Plug-Ins in die wandern.js-Datei .....	8
Abbildung 19: Einbindung der Maßstabsleiste in die wandern.js-Datei .....	8
Abbildung 20: Anlegen des Kontrollfeldes in der wandern.js-Datei .....	8
Abbildung 21: JavaScript im JSON Format in kulturmarker.js.....	9
Abbildung 22: Schleife für den Aufruf der Marker.....	9
Abbildung 23: Marker mit Popups und Inhalten auf der kultur.html-Seite .....	10
Abbildung 24: Cascade Style Sheet body für alle Seiten .....	10
Abbildung 25: Gestaltung der header am Beispiel #headerkultur.....	11
Abbildung 26: Header am Beispiel kultur.html .....	11
Abbildung 27: Erstellung des divs topnav .....	12
Abbildung 28: Erstellen des Drop-Down Untermenüs .....	12
Abbildung 29: Gestaltung der Menüleiste .....	13
Abbildung 30: Beschreibung der Funktion des Buchstaben a.....	13
Abbildung 31: Gestaltung des Drop-Down Menüs im style.css .....	14
Abbildung 32: Anzeige des Untermenüs auf der sport.html-Seie.....	14
Abbildung 33: Einbindung des Textes über .article im CSS .....	14
Abbildung 34: Stylesheet für die Karten .....	15
Abbildung 35: Graphische Darstellung des Stylesheets für kultur.html .....	15
Abbildung 36: Stylesheet und Ausgabe des footers.....	15
Abbildung 37: Neues definieren der Schriftgröße im Footer .....	16

# 1. Grundidee und Aufbau der Website

Die Grundidee der [Website](#) ist die Vorstellung verschiedener Aktivitäten und Orte in Innsbruck für Personen, die im Sommer die Stadt besuchen wollen. Die Zielgruppe sind vor allem junge, aktive Menschen aus dem Freundeskreis der Gestalter/-innen. Auf der Website befinden sich von den Autor/-innen vorgeschlagene Inhalte, die als Empfehlungen dienen sollen. Eine übersichtliche Zusammenstellung ausgewählter Inhalte war das Ziel; Vollständigkeit hingegen nicht.

Da drei Autor/-innen gleichzeitig an dieser Webseite gearbeitet haben, wurde ein Organization-Account mit dem Namen [Jemajo](#) auf [Github](#) angelegt. Im Bearbeitungsprozess konnte so gleichzeitig an dem Projekt gearbeitet und die Zwischenstände gespeichert werden. Außerdem wurde so der einfache Austausch der Dateien und ein direkter Aufruf der Website ermöglicht.

Um überhaupt auf die Seite zu gelangen muss eine kurze Quizfrage zum Thema Tirol beantwortet werden. Für alle, die die Frage nicht beantworten können, gibt es einen direkten Link zur Startseite. Die [Startseite](#) enthält eine kurze Beschreibung des Zwecks und des Inhaltes der Website sowie eine optisch ansprechende Darstellung mit Verlinkungen zu den Unterseiten. Der Aufbau der Unterseiten unterscheidet sich von dem der Startseite. Sie enthalten einen *header*, darunter eine Menüleiste, Textpassagen, zumeist eine Kartenapplikation sowie einen *footer* mit einem Link zu den Mailadressen der Autor/-innen.

Die Unterseiten decken folgende Bereiche ab:

- Kultur & Sehenswürdigkeiten
- ÖPNV
- Hungrig?
- Übernachtungsmöglichkeiten
- Sport
  - Wandern
  - Klettern

Diese Unterseiten unterscheiden sich in Details, haben aber grundsätzlich den gleichen Aufbau.

## 2. Umsetzung

### 2.1 Das Quiz

#### 2.1.1 Umsetzung auf der html-Seite

Für die Implementierung einer Quizfrage auf der ersten Seite, die erscheint sobald [jemajo.github.io](https://jemajo.github.io) aufgerufen wird, wurde der entsprechende Code auf der index.html-Seite geschrieben. Die optische Gestaltung wurde im gleichen Stil wie die Startseite gehalten (vgl. 2.2). Der *head*-Bereich enthält die Verlinkungen zu den benötigten Skripten. Das zentrale Skript mit dem Code für das Quiz wird aufgrund des Befehls *defer* zuerst geladen, um die korrekte Ausführung zu gewährleisten. Unter dem *body* steht der Code für die Inhalte des *Bootstraps*.

#### 2.1.3 Umsetzung der JavaScript-Seite

Für die Programmierung der Quizfrage wurden die Seminareinheiten zur Abfrage des Alters und des Namens als Grundlage genommen und mit Code-Beispielen zur Abfrage eines Passworts weiterentwickelt. Das zentrale Element ist eine *if*-Abfrage. Die richtige Antwort für auf die Frage ist im Prinzip das richtige Passwort. Einen besonders hohen Sicherheitsstandard hat diese Zugangskontrolle zu der Webseite nicht, da das Passwort bzw. die richtige Antwort im JavaScript-Code steht (Abb. 1).

```
function LogIn(){
    loggedin=false;
    password="piefke";
    password=prompt("Wie heißen 'bestimmte' Deutsche in Tirol?","");
    password=password.toLowerCase();
    if
        (password=="piefke") {
            loggedin=true;
            window.location="https://jemajo.github.io/frontpage";
        }
        if (loggedin==false) {
            alert("Sorry, du bist noch nicht bereit für deinen Aufenthalt in Innsbruck.");
        }
}
```

Abbildung 1: JavaScript Code zum Erstellen des Quiz

### 2.2 Bootstrapmodule im Quiz, auf der Start- und Sportseite

Auf der Startseite (frontpage.html) sind im *head*-Bereich Verlinkungen zu den Skripten von *Bootstrap* eingebaut. Weiterhin gibt es eine Verlinkung für die Schriftart Cookie und die Implementierung eines Icons in der Tableiste des Browsers. Der *body* der Startseite ist in drei Bereiche eingeteilt. Am Seitenanfang befindet sich ein schlichter *header* mit der Überschrift der Seite. Danach folgt ein Überblick

der verschiedenen Unterseiten. Am Ende der Seite ist ein *footer* eingebaut, in welchem Verlinkungen zu den Mailadressen der Autor/-innen eingebaut sind. Der Aufbau des Cascade Stylesheets wird in einem separaten Kapitel erklärt (s. Kapitel 2.5). Eine JavaScript-Datei ist für diese Seite nicht notwendig, da es um rein optische Darstellungen geht.

Die Card-Module von *Bootstrap* (vgl. *Bootstrap*), die im Quiz, auf der Startseite und auf der Sport-Überblicksseite verwendet werden geben einen guten optischen Überblick. Sie enthalten jeweils ein Bild, eine Überschrift, einen kurzen Text und eine Verlinkung zu der jeweiligen Seite (s. Abb. 2).

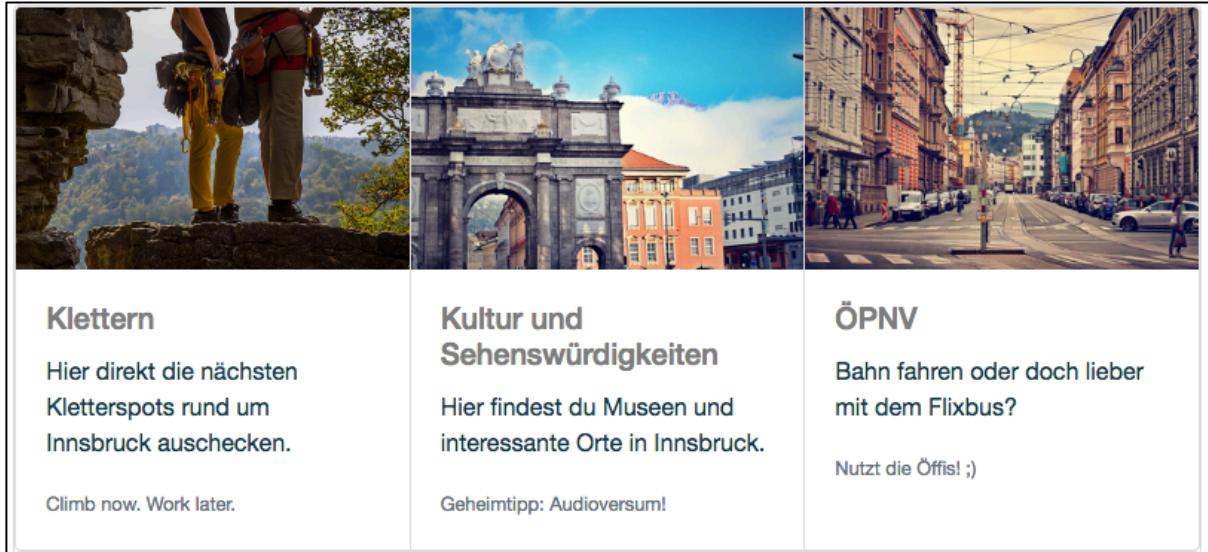


Abbildung 2: Bootstrap-Modul auf der Startseite

Damit die Seite schneller laden kann, werden die Module online abgerufen und sind im *header* folgendermaßen eingebunden:

```
<!-- Bootstrap-->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
| crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkVYIK3UENzM7KCKRR/rE9/0pg6aAZGJwFDMVNA/GpGFF93hXpG5KKN"
| crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B
+Y1QKtv3Rn7W3mgPxh9K/ScQsAP7UiB39j7fakFPsvXusvfa0b4Q"
| crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U2d8j0t6vLEHfe/JQGiRRSQxSfFWpi1MquDyjUar5+76PVcmYl"
| crossorigin="anonymous"></script>
```

Abbildung 3: Einbinden der Bootstrap-Module im Header

## 2.3 Kartenapplikation, Plugins, Marker und Tracks auf der Unterseite Wandern

Die Unterseite [Wandern](#) wurde mit einem selbst geschriebenen .html-File, einem JavaScript-File sowie den .gpx-Files des Adlerwegs umgesetzt.

### 2.3.1 Einbinden von Textpassagen und externen Inhalten

Unter der Menüleiste (s. Kapitel 2.5) wurde eine Textpassage eingefügt, die Informationen zu den Etappen 12 und 13 auf dem Adlerweg enthält. Verlinkungen zu externen Inhalten runden die Informationen der Textpassage ab. Dazu zählt auch die Vorhersage von Unwettern durch das ZAMG, die durch die Einbindung von entsprechendem html-Text von [www.zamg.at](http://www.zamg.at) als Widget erscheint (vgl. ZAMG).

```
84  <div id="zamg_wetterwarnungen">
85    <a href="http://warnungen.zamg.at/html/de/heute/alle/at/" target="_blank">
86      
88  </a>
89 </div>
```

Abbildung 4: Einbindung des ZAMG-Codes in die wandern.html-Datei

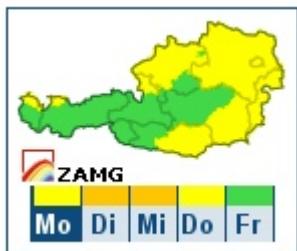


Abbildung 5: Output auf der wandern.html-Seite

### 2.3.2 Anlegen eines Kartenfeldes mit Layern

Für die Kartenapplikation wurde die JavaScript-Bibliothek Leaflet verwendet (vgl. Agafonkin). Im html-File wurde ein Bereich (*div*) für das Kartenfeld angelegt (s. Abb. 6). Sie wurde in der wander.js-Datei aufgegriffen (vgl. Github).

```
91   <div id="mapdiv">
92     </div>
```

Abbildung 6: Anlegen einer mapdiv für die Karte

Zwei Hintergrundlayer mit Beschriftung sind in der Kartenapplikation verfügbar. Dafür wurden Layer mit der Openstreetmap (vgl. Openstreetmap), der Nomenklatur (vgl. Nomenklatur) und einem Orthofoto (vgl. Stadt Wien und Österreichische Länder bzw. Ämter der Landesregierung) erstellt.

```

12 let myLayers = {
13   osm: L.tileLayer(
14     "https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png",
15     {
16       subdomains: ["a", "b", "c"],
17       attribution: "Datenquelle: <a href='https://www.openstreetmap.org'>openstreetmap.org</a>"
18     }
19   ),
20   tiris_nomenklatur: L.tileLayer(
21     "http://wmts.kartetirol.at/wmts/gdi_nomenklatur/GoogleMapsCompatible/{z}/{x}/{y}.png8", {
22       attribution: "<a href='https://www.data.gv.at/Katalog/dataset/land-tirol_elektronischekartetirol'>eKarte Tirol</a>",
23       pane: "overlayPane",
24     }
25   ),
26   bmaporthofoto30cm: L.tileLayer(
27     "https://{s}.wien.gv.at/basemap/bmaporthofoto30cm/normal/google3857/{z}/{y}/{x}.jpeg",
28     {
29       subdomains: ["maps", "maps1", "maps2", "maps3", "maps4"],
30       attribution: "Datenquellen: <a href='https://www.basemap.at'>basemap.at</a>"
31     }
32   );
33 };
34 myMap.addLayer(myLayers.osm);
35
36 //Gruppierung Orthophoto & Beschriftung
37 let ortho_m_beschr = L.featureGroup(
38   [myLayers.bmaporthofoto30cm,
39   myLayers.tiris_nomenklatur]
40 );

```

Abbildung 7: Anlegen der Hintergrundlayer in der wandern.js-Datei

Durch eine Gruppierung der Layer des Orthofotos und der Nomenklatur liegen diese beiden in der Kartenapplikation übereinander.

```

37 let ortho_m_beschr = L.featureGroup(
38   [myLayers.bmaporthofoto30cm,
39   myLayers.tiris_nomenklatur]
40 );

```

Abbildung 8: Gruppieren der Layer Orthofoto und Nomenklatur

### 2.3.3 Einbinden eines gpx-Files über ein Plugin

Der .gpx-File des Adlerwegs stammt von der Website [www.data.gv.at](https://www.data.gv.at) (vgl. Adlerweg) und seine Einbindung wurde mithilfe eines Leaflet-Plugins umgesetzt (vgl. Mpetazzoni).

```

19   <link rel="stylesheet" href="../js/leaflet.elevation/leaflet.elevation-0.0.4.css" rel='stylesheet' />
20   <link rel="stylesheet" href="../js/leaflet.elevation/leaflet.elevation-0.0.4.css" />
21   <script src="../js/leaflet.elevation/leaflet.elevation-0.0.4.min.js"></script>

```

Abbildung 9: Einbindung des GPX-Plug-Ins in die wandern.html-Datei

Die Einbindung des Tracks der Etappe 12 erfolgte wie folgt:

```

77 let gpxTrack12 = new L.GPX("AdlerwegEtappe12.gpx", {
78   async: true,
79 });
80
81 gpxTrack12.on("loaded", function (evt) {
82   let track = evt.target;
83 }).addTo(myMap);
84

```

Abbildung 10: Einbindung des GPX-Tracks in die wandern.js-Datei

An den Start- und Endpunkten wurde ein passendes Icon (vgl. Mollet) gesetzt und über die Koordinaten aus dem gpx-File verortet.

```
51 myMap.addLayer(etappeGroup12);
52 const start12 = [47.35467730, 11.47713480];
53 const finish12 = [47.27091590, 11.39537570];
54
55 L.marker(start12, {
56   icon: L.icon({ iconUrl: '../icons/start.png', iconAnchor: [15, 35], })
57 }).addTo(etappeGroup12);
58
59 L.marker(finish12, {
60   icon: L.icon({ iconUrl: '../icons/finish.png', iconAnchor: [15, 35], })
61 }).addTo(etappeGroup12);
```

Abbildung 11: Einbindung von Markern für die Start- und Endpunkte des Tracks mit passenden Icons

Die Track-Variable wurde mit der Variable der Start- und Endpunkte gruppiert, um sie gemeinsam als Layer in der Kontrollbox auf der Karte anzeigen lassen zu können.

```
99 let etappe12 = L.featureGroup(
100 [
101   etappeGroup12,
102   gpxTrack12,
103 ],
104 );
```

Abbildung 12: Gruppieren der Variablen des GPX-Tracks mit Start- und Endpunkt des Tracks

Analog wurde für die Etappe 13 verfahren. Die Farbe des Tracks wurde in der leaflet.gpx-Bibliothek (vgl. Mpetazzoni) im File gpx.js von der Default-Farbe „blue“ auf „#b40000“ geändert.

```
60 var _DEFAULT_POLYLINE_OPTS = {
61   color: '#b40000'
62 };
```

Abbildung 13: Ändern der Default-Farbe des GPX-Tracks

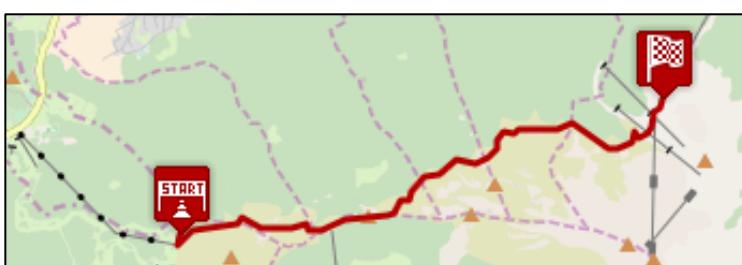


Abbildung 14: Ausgabe der Änderung auf der wandern.html-Seite

## 2.3.4 Einbinden eines Höhenprofil-Plug-Ins

Für Track 12 und 13 wurde wie folgt das Plug-In „Elevation“ eingebunden, um für jeden Track ein aufklappbares Höhenprofil anzeigen zu lassen (vgl. MrMufflon).

```
129 let hoehenProfil12 = L.control.elevation({
130   position: "topright",
131   theme: "steelblue-theme",
132   width: 600,
133   height: 125,
134   margins: {
135     top: 10,
136     right: 20,
137     bottom: 30,
138     left: 50
139   },
140   useHeightIndicator: true,
141   interpolation: "linear",
142   hoverNumber: {
143     decimalsX: 3,
144     decimalsY: 0,
145     formatter: undefined
146   },
147   xTicks: undefined,
148   yTicks: undefined,
149   collapsed: true,
150   imperial: false
151 }).addTo(myMap);
152
153 gpxTrack12.on("addline", function (evt) {
154   hoehenProfil12.addData(evt.line);
155 });
156 gpxTrack12.addTo(myMap);
157
158 L.control.scale({
159   maxWidth: 200,
160   metric: true,
161   imperial: false,
162   position: "bottomleft"
163 }).addTo(myMap);
```

Abbildung 15: Einbinden des Höhenprofil Plug-Ins in die wandern.js-Datei

Die Höhenprofile können auf der [Wander-Unterseite](#) unter diesem Icon aufgerufen werden.

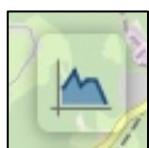


Abbildung 16: Icon zum Aufrufen des Höhenprofils

## 2.3.5 Hinzufügen von Kartenelementen: Fullscreen-Plug-In, Maßstabsleiste, Kontrollfeld

Ein Fullscreen-Plug-In (vgl. Github) wurde in der Kartenansicht verwendet.

```
23    <link rel="stylesheet" href="../js/leaflet.fullscreen/leaflet.fullscreen.css" />
24    <script src="../js/leaflet.fullscreen/Leaflet.fullscreen.js"></script>
```

Abbildung 17: Einbindung des Fullscreen Plug-Ins in die wandern.html Datei

```
1  let myMap = L.map("mapdiv", {
2    fullscreenControl: true,
3    fullscreenControlOptions: {
4      position: 'topleft'
5    }
6  );
7);
```

Abbildung 18: Einbindung des Fullscreen Plug-Ins in die wandern.js-Datei

Die Maßstabsleiste für die Kartenapplikation wird wie folgt definiert:

```
50  L.control.scale({
51    maxWidth: 200,
52    metric: true,
53    imperial: false,
54    position: "bottomleft"
55 }).addTo(myMap);
```

Abbildung 19: Einbindung der Maßstabsleiste in die wandern.js-Datei

Zudem wurde ein Kontrollfeld angelegt, dass das Umschalten zweier Hintergrundlayer sowie das Ein- und Ausblenden der Wandertracks ermöglicht.

```
113  let myMapControl = L.control.layers({
114    "Karte": myLayers.osm,
115    "Orthofoto": ortho_m_beschr,
116  },
117  {
118    "Der Goetheweg - Etappe 12": etappe12,
119    "Der Zirbenweg - Etappe 13": etappe13,
120  },
121  {
122    collapsed: false
123  });
124
125 myMap.addControl(myMapControl);
```

Abbildung 20: Anlegen des Kontrollfeldes in der wandern.js-Datei

## 2.4 Kartenapplikation, Plug-Ins und Marker auf den Seiten Kultur, Klettern, Hunger?, ÖPV

Die weiteren Unterseiten (Kultur, Klettern, Hunger? und ÖPV) unterscheiden sich nicht maßgeblich von der bereits beschriebenen wandern.html-Seite. Es wurden genauso Textpassagen geschrieben und eine Karte angelegt, sowie teilweise Wetterwarnungen eingebunden. Auch das Fullscreen- Plug-In wird in jeder Karte verwendet. Es erfolgte keine Einbindung des gpx-Plug-Ins und des Höhenprofil-Plug-Ins, da die Daten einer anderen Struktur unterliegen.

Für die Kartenapplikationen wurden JavaScript-Dateien im JSON Format selbst geschrieben. Sie beinhalten Koordinaten und weitere interessante Informationen. Die Daten der Kulturmarker (kulturmarker.js) schauen beispielsweise folgendermaßen aus:

```
1  const pois = [
2    { "lat": "47.268827", "lng": "11.393310", "name": "Goldenes Dachl", "what": "Das Wahrzeichen der Stadt!  
3      <br> Hier gibt es auch ein <a href='https://www.innsbruck.gv.at/goldenesdachl/'>Museum</a>.", "adresse":  
      "Herzog-Friedrich-Straße 15 <br> 6020 Innsbruck" },
4    { "lat": "47.256794", "lng": "11.433597", "name": "Schloss Ambras", "what": "Nationalmuseum mit  
      Porträtsammlungen, Kunstwerken und Rüstkammer im Schloss aus der Spätrenaissance.<br>Für mehr Infos  
      klicke <a href='https://www.schlossambras-innsbruck.at/'>hier</a>.", "adresse": "Schloßstraße 20 <br>  
      6020 Innsbruck" },
```

Abbildung 21: JavaScript im JSON Format in kulturmarker.js

In der JavaScript Datei (kultur.js), in der die Karte genau definiert ist, werden diese Daten in Abbildung 21 gezeigte Schleife abgerufen. Zum Abrufen der Informationen für die Marker wurde eine *for...of* Schleife verwendet, die auf das *Array* in dem Datensatz (*pois*) zugreift. Mit *bindPopup* wird jedem Marker ein Popup hinzugefügt, dass die Inhalte aus dem JSON-File abruft. Über *fitBounds* zoomt der Kartenausschnitt auf die gesetzten Marker.

```
69  for (let macc of pois) {
70    L.marker([macc.lat, macc.lng], {
71      icon: myIcon,
72    })
73    .bindPopup(`<h1>${macc.name}</h1>
74      <p> Was ist das? <br> ${macc.what} <br> <br> ${macc.adresse}</p>`)
75    .addTo(kulturLayer);
76  }
77  myMap.fitBounds(kulturLayer.getBounds());
```

Abbildung 22: Schleife für den Aufruf der Marker

Das Ergebnis ist folgendes:

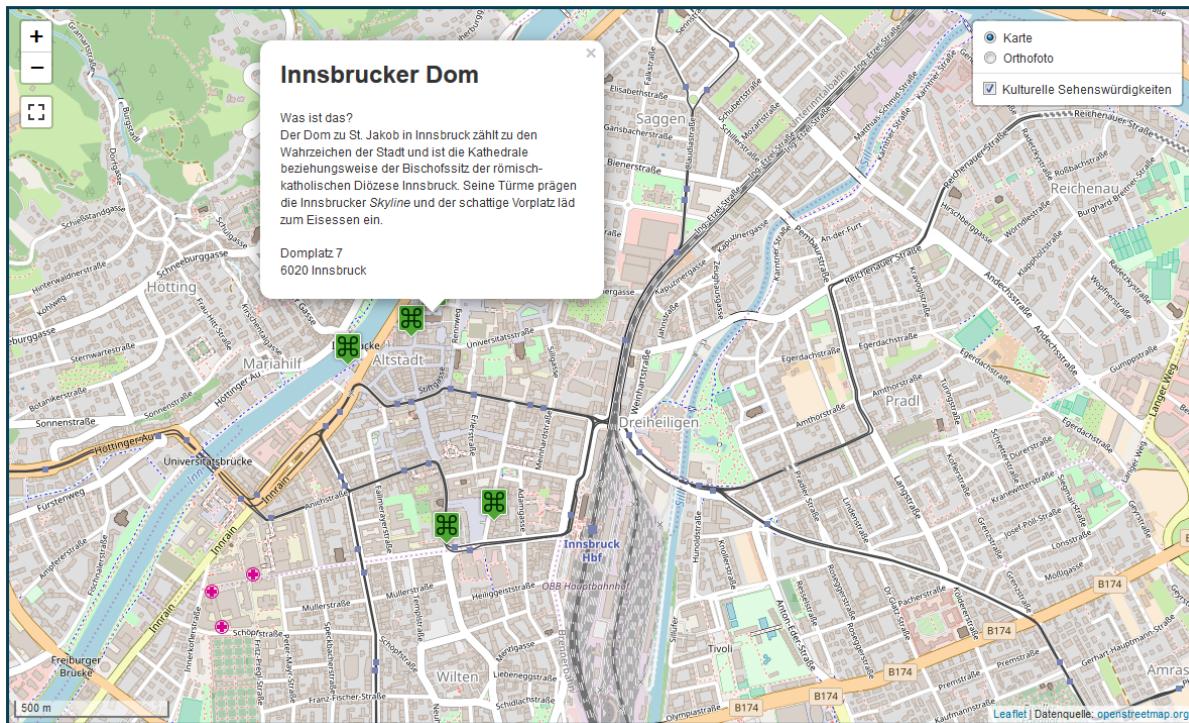


Abbildung 23: Marker mit Popups und Inhalten auf der kultur.html-Seite

Wie bereits angesprochen, wurde auf den anderen Unterseiten das gleiche Prinzip angewandt, weshalb diese hier nicht mehr im Detail erklärt werden.

## 2.5 Cascade Style Sheets

Für die komplette Website gibt es lediglich zwei *Cascade Style Sheets*. Zu Zwecken der Übersichtlichkeit ist das *Stylesheet* des Quiz (*quiz.css*) auf der *index.html* Seite separat gestaltet, da sich sein Aufbau von dem der anderen Seiten unterscheidet und gleiche *Bootstrap Card-Module* verwendet wurden. Das *Quiz-Stylesheet* zentriert das Modul mit den Inhalten, definiert die Ränder und die Breite.

Für alle anderen Seiten wurde ein gemeinsames *Stylesheet* erarbeitet. Als erstes wurde die generelle Anzeige des *bodys* auf allen Seiten definiert.

```
1  body{  
2      max-width: 90%;  
3      margin-left: auto;  
4      margin-right: auto;  
5      font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
6      color: #063E53;  
7  }
```

Abbildung 24: Cascade Style Sheet body für alle Seiten

Die Breite des Inhaltes wurde auf 90% festgelegt, da so unabhängig von der Bildschirmgröße gewährleistet ist, dass sich der Inhalt automatisch anpasst. Durch das Setzen von *margin-right* bzw. *margin-left* auf *auto* wird sichergestellt, dass die Inhalte zentriert angezeigt werden. Die *font-family* ist an die Schriftart der *Bootstrap*-Module angepasst und bezieht sich auf den Text. Die Schriftfarbe ist dem generellen Design der Seite angepasst.

Für die einzelnen Header wurden jeweils Bereiche (*div*) mit eigener *id* angelegt, da sie nur die Komponenten *height*, *padding* und *font-family* gemeinsam haben und sich in sonst allen Komponenten unterscheiden.

```
#headerkultur {  
    height: 500px;  
    padding: 30px;  
    text-align: right;  
    font-size: 3.0em;  
    color: #063E53;  
    background-image: url("images/kultur.jpg");  
    font-family: 'Cookie', cursive;  
    background-position-x: center;  
    background-position-y: 26%;  
}
```

Abbildung 25: Gestaltung der header am Beispiel #headerkultur

Die Gestaltung der *header* werden am Beispiel von *#headerkultur* erklärt. *Height* und *padding* definieren die Anzeigegröße des *headers* im Allgemeinen. Die Ausrichtung des Textes (*text-align*) und die Farbe (*color*) wurden je nach Headerbild und Lesbarkeit der Schrift auf dem Bildhintergrund definiert. Die Schriftart *Cookie* wurde aus Design-Gründen gewählt und unterscheidet sich von der Schriftart der Artikel darunter. Außerdem wurde für jede Seite dem Thema entsprechend ein separates Headerbild gewählt. Der gezeigte Ausschnitt der Bilder wurde über *background-position-x/y* optimal einggerichtet.

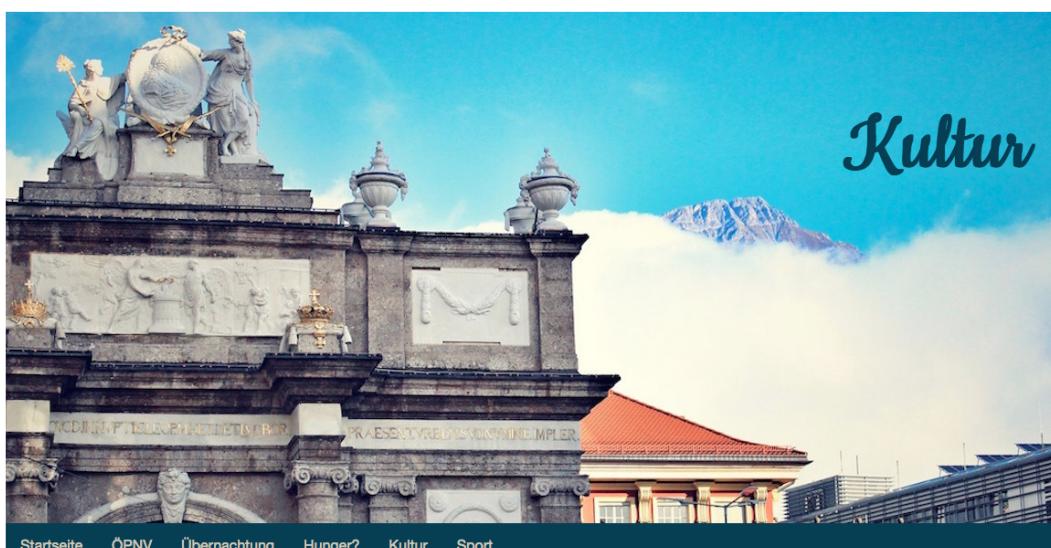


Abbildung 26: Header am Beispiel kultur.html

In Abbildung 26 ist außerdem zu sehen, dass direkt nach dem *header* mit Bild eine Menüleiste folgt. Für die Gestaltung der Menüleiste wurden horizontal angeordnete Kacheln erstellt. Bei der Gestaltung der Startseite wurde auf die Leiste verzichtet und die *Bootstrap Module* für die Navigation auf der Seite verwendet. Diese Variante wurde gewählt, da die Startseite neben der Darstellung der verschiedenen Unterseiten keinen eigenen Inhalt hat.

Zum Erstellen der Menüleiste wurde als erster Schritt im html-Dokument jeder Seite die *div* mit der *id „topnav“* erstellt. In dieser *div* sind die Links zu jeder der Themenseiten mit entsprechender Bezeichnung eingebaut.

```
33   <div id="topnav">
34     <a href="#">.../frontpage.html">Startseite</a>
35     <a href="#">.../an_abreise/an_abreise.html">ÖPV</a>
36     <a href="#">.../accommodation/accommodation.html">Übernachtung</a>
37     <a href="#">.../eat/eat.html">Hunger?</a>
38     <a href="#">.../kultur/kultur.html">Kultur</a>
39     <a href="#">sport.html">Sport</a>
40   </div>
```

Abbildung 27: Erstellung des divs topnav

Für die Kletter-, und Wandern-Unterseite wurde zur Auswahl der jeweils anderen Sportart ein „*Drop-Down*“-Untermenü in die Menüleiste integriert. Auf der allgemeinen Sport-Themenseite wurde aus den gleichen Gründen wie bei der Startseite auf eine Menüleiste und die „*Drop-Down*“ Funktion verzichtet.

```
48   <div class="dropdown">
49     <button class="dropbutton">Sportliche Aktivitäten</button>
50     <div class="dropdown-content">
51       <a href="#">klettern.html">Klettern</a>
52       <a href="#">wandern.html">Wandern</a>
53     </div>
```

Abbildung 28: Erstellen des Drop-Down Untermenüs

Zur Gestaltung der Menüleiste und der „*Drop-Down*“ Funktion muss in der style.css-Datei auf die beidem *divs* verwiesen werden. Im style.css wird als erstes die Menüleiste in ihrer grundlegenden Form gestaltet. Im zweiten Schritt wurde programmiert, dass sich die Kacheln in der Farbe verändern, sobald mit der Computermaus über sie gefahren wird (s. Abbildung 29).

```

111 #topnav{
112     overflow: hidden;
113     display: block;
114     position: relative;
115     background-color: #063E53;
116     color: #D1C5B7;
117
118 }
119 #topnav a{
120     float: left;
121     display: block;
122     color: #D1C5B7 ;
123     text-align: center;
124     padding: 14px 16px;
125     text-decoration: none;
126 }
127 #topnav a:hover{
128     background-color: #0193B7;
129     color: #D1C5B7;
130 }

```

Abbildung 29: Gestaltung der Menüleiste

Der Buchstabe a hinter der *id* der *div* unterscheidet zwischen angeklicktem und nicht angeklickten Link.

```

a.list, a:visited{
    color: grey;
    text-decoration: none;
}

```

Abbildung 30: Beschreibung der Funktion des Buchstabens a

Die Gestaltung der „*Drop-Down*“ Funktion folgt dem gleichen Prinzip. Zusätzlich zu der nicht ausgeklappten Version musste noch die ausgeklappte Version, die andere Farbe beim darüberfahren mit der Computermaus und die angeklickte Version gestaltet werden.

```

161 .dropbutton{
162     background-color: #D1C5B7;
163     color: #063E53;
164     padding : 14px 16px;
165     font-size : 12px;
166     border: none;
167 }
168 .dropdown{
169     position: relative;
170     display: inline-block;
171 }
172 .dropdown-content{
173     display : none;
174     position : absolute;
175     background-color: #D1C5B7;
176     min-width: 160px;
177     box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
178     z-index: 1;
179 }

```

```

180 .dropdown-content a{
181   color : black;
182   padding: 8px 16px;
183   text-decoration: none;
184   display: block;
185 }
186 .dropdown:hover .dropdown-content{
187   display: block;
188 }
189 .dropdown:hover .dropbutton{
190   background-color: #0193B7;
191   color: #D1C5B7;
192 }

```

Abbildung 31: Gestaltung des Drop-Down Menüs im style.css

Auf der Webseite wird dies wie folgt angezeigt:



Abbildung 32: Anzeige des Untermenüs auf der sport.html-Seite

Nach der Menüleiste folgt die Klasse *.article* in der die Ausrichtung des Textes genauer definiert wird. *padding* ist in diesem Zusammenhang sehr hilfreich, da die Abstände vom Artikel zum *header* und *footer* auf allen Seiten gleich definiert sind und das Layout so immer stimmig ist.

```

191 .article {
192   margin: auto;
193   width: 100%;
194   padding: 3% 0 3% 0;
195 }

```

Abbildung 33: Einbindung des Textes über *.article* im CSS

Zwischen den Artikeln sind häufig Karten eingebunden. Die Höhe und die Breite der Karten wurde definiert (s. *width*, *height*), sie haben einen Rand und sind automatisch mittig zentriert. Außerdem sind Abstände nach oben und unten zum Text angegeben.

```

197  #mapdiv {
198      width: 100%;
199      height: 660px;
200      border: 3px solid #063E53;
201      margin-right: auto;
202      margin-left: auto;
203      padding: 2% 0 2% 0;
204 }

```

Abbildung 34: Stylesheet für die Karten

Auf der Seite `kultur.html` wird das wie folgt dargestellt:



Abbildung 35: Graphische Darstellung des Stylesheets für `kultur.html`

Auch der `footer` ist auf allen Seiten gleich. Auf den Seiten wird der Kontakt zu den Autor/-innen angezeigt. Der Inhalt ist zentriert und hat die gleiche Schriftart wie die Header.

```

210  #footer{
211      text-align: center;
212      margin: auto;
213      color: #063E53;
214      padding: 1% 0 5% 0;
215      font-family: 'Cookie', cursive;
216 }

```

*Autor\*innen*  
Jessica Balling - Marlene Weiß - Jacob Heuser

Abbildung 36: Stylesheet und Ausgabe des `footers`

Neben dem generellen Aufbau der Seite gab es weitere kleine Anpassungen. Beispielsweise wurde die Schriftfarbe von Links nach dem Aufrufen farblich anders dargestellt. Weitere Anpassungen wurden bei den html-Seiten mit `Bootstrap` Modulen vorgenommen, da durch das Einbinden von `Bootstrap` die Formatierung für die komplette Seite überschrieben wurde. Diese Überschreibung

kann wiederum überschrieben werden, wenn die Dinge im CSS definiert sind. Dies war in den *footern* und auf der Sport-Seite im *header* notwendig.

```
184  #footer{  
185      text-align: center;  
186      margin: auto;  
187      color: ■#063E53;  
188      padding: 1% 0 5% 0;  
189      font-family: 'Cookie', cursive;  
190  }  
191  #footer h2 {  
192      font-size: 32px;  
193  }
```

Abbildung 37: Neues definieren der Schriftgröße im Footer

### 3. Herausforderungen und Abweichungen

Eine kleine Hürde stellte am Anfang des Prozesses das Schreiben und Einbinden der selbstgeschriebenen Geojson/JS-Dateien dar. Im Team gelang es letztendlich die Dateien in der richtigen Struktur zu verfassen und per Schleifen in die jeweiligen Kartenansichten einzubinden.

Das Layout einiger Seiten wurde mit Bootstrap-Modulen ausgestattet. Dabei ist erst im Nachhinein aufgefallen, dass bereits definierte Stylesheets von Bootstrap überschrieben wurden. Durch genauere Definitionen im Stylesheet konnten diese Überschreibungen umgangen werden. Für zukünftige Projekte könnte es sinnvoll sein, die kompletten Formatierungen von *Bootstrap* zu übernehmen. Dabei geht allerdings eigene Kreativität verloren. Über *Bootstrap* ist jedoch auch eine optisch ansprechende Anzeige auf mobilen Endgeräten gegeben.

Auf der Unterseite ÖPV konnten die Stadtrad-Standorte nicht in die eigene Kartenapplikation eingebunden werden, da passende Daten nicht frei zugänglich im Netz aufzufinden waren und auf Nachfrage die Dateien nicht herausgegeben werden dürfen.

Vom ursprünglichen Konzept wurde in einigen Punkten abgewichen. Der Adlerweg wurde eingebunden und die Startseite als *Bootstrap*-Seite konzipiert. Nicht umgesetzt wurden hingegen die Kommentarfunktion und die Ausgestaltung eines Impressums. Die Unterseite „Mountainbike“ wurde ebenfalls ausgelassen, da die Autor/-innen aufgrund fehlender Ausstattung und Kenntnisse ihre Besucher/-innen dabei nicht begleiten würden und dazu bereits genug Webseiten bestehen.

Eine weitere Herausforderung war die Zeit. Das Erreichen der Ziele, die anfangs gesteckt wurden, dauerte oft länger als gedacht; außerdem erforderte die Umsetzung spontaner, kreativer Einfälle ebenfalls viel Zeit, zumal es sich dabei um Inhalte handelte, die im Rahmen der Vorlesung nicht besprochen wurden. Bei einem *Usability* Test wurde außerdem herausgefunden, dass es sinnvoll ist, die Links in einem neuen Tab öffnen zu lassen, sodass der User auf der Seite bleibt. Dies kann beim weiteren Ausbau der Seite – der geplant ist – eingebaut und berücksichtigt werden.

## 4. Quellenverzeichnis

- Agafonkin, Vladimir: „Download Leaflet“ unter: <https://leafletjs.com/download.html> (zuletzt aufgerufen am 11.06.2018).
- Bootstrap: „Bootstrap Cards“, unter: <https://getbootstrap.com/docs/4.0/components/card/> (zuletzt aufgerufen am 11.06.2018).
- Bundesministerium für Digitalisierung und Wirtschaftsstandort: „Adlerweg“, unter: <https://www.tirol.at/action/download?id={c875c6c5-8d33-de4a-230c-7366b366cd6c}> (zuletzt aufgerufen am 11.06.2018).
- Github: „Leaflet.fullscreen“, unter: <https://github.com/Leaflet/Leaflet.fullscreen> (zuletzt aufgerufen am 11.06.2018).
- Land Tirol: „Elektronische Karte Tirols“, unter: [https://www.data.gv.at/katalog/dataset/land-tirol\\_elektronischekartetirol](https://www.data.gv.at/katalog/dataset/land-tirol_elektronischekartetirol) (zuletzt aufgerufen am 11.06.2018).
- Mollet, Nicolas: „Mapicons“, unter: <https://mapicons.mapsmarker.com/> (zuletzt aufgerufen am 11.06.2018).
- Mpetazzoni: „Leaflet-gpx“, unter: <https://github.com/mpetazzoni/leaflet-gpx> (zuletzt aufgerufen am 11.06.2018).
- MrMufflon: „Leaflet.Elevation“, unter: <https://github.com/MrMufflon/Leaflet.Elevation> (zuletzt aufgerufen am 11.06.2018).
- Openstreetmap, unter: <https://www.openstreetmap.org/> (zuletzt aufgerufen am 11.06.2018). Falsch, findet aber grad auch nicht
- Stadt Wien und Österreichische Länder bzw. Ämter der Landesregierung: „Basemap“, unter: <https://www.basemap.at/> (zuletzt aufgerufen am 11.06.2018).
- ZAMG (Zentralanstalt für Meteorologie und Geodynamik): „Wetter/Wetterwarnungen“ unter: <http://warnungen.zamg.at/warn4hp/de/heute/alle/at/> (zuletzt aufgerufen am 11.06.2018).