

Allgemeines

Abgabe: bis **Fr. 13.12.2024, 12:00**

Bearbeiten Sie die Aufgabe mit Hilfe von einem Texteditor (z.B. Notepad++) oder einer Entwicklungsumgebung Ihrer Wahl (z.B. VisualStudioCode). Wichtig ist, dass Sie innerhalb der Abgabefrist Ihre Ergebnisse in moodle abgeben.

Erstellen Sie neben Ihrem Code auch eine Dokumentation über Ihre Vorgehensweise.

Bearbeiten Sie die Aufgaben selbständig und nutzen Sie möglichst das Tutorium!
Bei Fragen und Unklarheiten benutzen Sie bitte das Diskussionsforum im Moodle.

Dokumentation: Dokumentieren Sie so viel wie nötig und so wenig wie möglich! Wichtig ist, dass Ihre Vorgehensweise nachvollziehbar und die Ergebnisse zu erkennen sind. Gegebenenfalls notieren Sie alle Befehlseingaben.

Webmapping mit OpenLayers

In diesem Übungsblatt sollen Sie eine Website mit 3 Unterseiten erstellen. Jede der 3 Unterseiten soll eine OpenLayers Karte enthalten. Alle Seiten sollen mit Hilfe einer Navigationsleiste erreichbar sein.

1) HTML, CSS (2 Punkte)

Legen Sie zunächst alle .html Dateien an. Füllen Sie diese Dateien mit einem html Grundgerüst. Vernetzen Sie die 3 Seiten mit Hilfe einer Navigationsleiste.

2) Karte mit GeoJSON Daten und clientseitigem Styling (6 Punkte)

- Binden sie die OpenLayers Bibliothek ein (CSS + JS). (1 P)
- Erzeugen Sie eine OpenLayers Karte mit einem OSM Hintergrundlayer. (1 P)
- Fügen Sie einen weiteren Vector Layer hinzu, als Quelle soll die Datei „bundeslaender.geojson“ verwendet werden (Flächen) (1 P)
- Fügen Sie noch einen weiteren Vector Layer hinzu, als Quelle soll die Datei „numberOfWindTurbines.geojson“ verwendet werden (Punkte).(1 P)
- Benutzen Sie eine Style-Funktion, welche die **Fläche** der dargestellten Kreise proportional zur Anzahl (*value*) skaliert. Benutzen Sie folgende Formel um den passenden *radius* für jedes Feature zu berechnen. (2 P).

$$radius = radius_{min} \sqrt{\frac{value}{value_{min}}}$$

$radius_{min}$ bzw. $value_{min}$ entspricht dem Radius des kleinsten Symbols bzw. Wertes im Datensatz.

3) Karte mit *remote* Daten aus einem Tilecache (3 Punkte)

In dieser Teilaufgabe sollen Sie eine Karte (Tiles) von OSMLanduse (<https://osmlanduse.org>) mit OpenLayers einbinden.

- a) Binden sie die OpenLayers Bibliothek ein (CSS + JS). (1 P)
- b) Binden Sie nun den Layer „osmlanduse:osm_lulc_combined_osm4eo“ von OSMLanduse als Tile-Layer mit XYZ-Source ein. (2 P)

URL: https://maps.heigit.org/osmlanduse/tiles/osmlanduse:osm_lulc_combined_osm4eo/webmercator/{z}/{x}/{y}.png

4) Karte mit CSV Daten „upload“ (9 Punkte)

- c) Binden sie die OpenLayers Bibliothek ein (CSS + JS). (1 P)
- d) Erzeugen Sie eine OpenLayers Karte mit einem OSM Hintergrundlayer.(1 P)

Ermöglichen Sie dem Nutzer Ihrer Webseite eigene Daten auf der Karte anzuzeigen.

- e) Fügen Sie hierzu ein HTML Element <input> vom Typ „file“ hinzu, um dem Nutzer die Möglichkeit zu geben eine lokale Datei (CSV) von seinem Rechner auszuwählen.(1 P)
- f) Parsen sie die CSV Datei des Nutzers (Landeshauptstädte) mit der Bibliothek PapaParse. (2 P)
- g) Generieren Sie aus den eingelesenen Daten Features mit Punkt-Geometrien und Eigenschaften aus den Daten. (2 P)
- h) Fügen Sie einen Vector-Layer mit Vector-Source und Ihren zuvor generierten Features der Karte hinzu (2 P)