**Wiki Документация**

**Интерактивная folium карта**

**1.Кратко о проекте**Проект представляет собой веб-сайт с минималистическим дизайном, позволяющий визуально рассмотреть состояние стран по рейтингу счастья.  
Разработка проекта состояла из 2-х частей:  
а) Python разработка:  
Используя библиотеку folium, можно достаточно легко и быстро получить готовую карту в .html формате.  
Сам же folium является более упрощенным вариантом взаимодействия с библиотекой leaflet для JavaScript.  
б) Web разработка:  
Получив нужные варианты карт при помощи folium (а их 27 штук),можно переходить к разработке уже web части сайта. Используя языки HTML , CSS, JavaScript мы привели сайт к наиболее эргономичному виду.  
**2.Документация по файлам.***main.py* – файл python, который как раз таки позволяет на выходе получить карту в формате .html  
*main.html* – стартовая(основная) страница веб-сайта  
*css* – папка, содержащая все файлы(style.css, leaflet.css) таблицы стилей, используемые для привидения сайта к надлежащему виду  
*maps* – папка, содержащая все карты  
*countries.json* – файл с гео-данными  
*script.js –* файл со скриптами javascript  
*Data* – папка, содержащая файлы с данными для создания карт(2019…2015.csv)  
**3.Документация по коду**а) *main.py*  
Используемые библиотеки:  
1) folium для создания карт  
2) pandas для анализа данных  
3) json для гео-данных  
Как это работает?  
1) В geojson помещаем файл с гео-данными в формате json.  
2) В data помещаем файл с данными (в нашем случае данные для карты счастья)  
3) В folium.Choropleth меняем аргументы:  
geo\_data – гео-данные  
data – данные со значениями  
Далее в Columns выбираем те 2 колонки из файла с данными(2019…2015.csv), по которым хотим получить карту.

ВАЖНО: первой колонкой должны быть названия карт, чтобы folium смог синхронизировать данные из .csv файла с гео-данными из country.json  
А во второй можно выбрать любую из оставшихся колонок .csv файла.  
На выходе получаем .html файл с нужной картой.  
б) main.html и style.css  
В первом html файле все достаточно просто, так как весь код является добавочным, то есть не содержит код, созданный библиотекой folium, кроме кода в <header>(помимо этого в header содержится название сайта, лого сайта, подключение шрифта), так как в нем подключены файлы через <link> для карты.  
Тег <div> используется как некий “контейнер” контента, которому можно менять задний фон, изменять размеры при помощи классовой/id системы и тегов css  
На картинке показаны области внешних div’ов с их ID  
Также стоит заметить, что вся страница принадлежит div’у с id “main”  
Остальные менее значительны и отвечают за расположение функционала:  
все id(#) и классы(.) главной страницы(редактируются в файле style.css):  
а) .main text – расположение основного контента “About” относительно контейнера div #main-text, цвет текста  
б) .main-png – класс картинки(лого) с размерами,  
в) .btn\_nav .btn-blue .dropbtn .dropdown-content – класс ссылок(кнопок) и dropdown-контента, находящиеся в верхней части сайта контейнера div #top  
г) .information -настройка шрифта основной информации div’a #main-text