Для встановлення бібліотеки shapely необхідно використати команду

1. !pip install shapely
2. pip install shapely
3. pip instal shapely
4. install shapely
5. !install shapely

Виберіть рядок коду в якому демонстується функція для створення точки бібліотеки shapely

1. point = Point(1, 1)
2. point = Dot(1, 1)
3. point = LineString([(1, 1), (2, 2)])
4. point = Polygon([(0, 0), 0, 1), (1, 1), (1, 0)])
5. point = Punctum(1, 1)

Виберіть рядок коду в якому демонстується функція для створення Лінії бібліотеки shapely

1. line = LineString([(1, 1), (2, 2)])
2. line = Line([(1, 1), (2, 2)])
3. line = Polygon([(0, 0), 0, 1), (1, 1), (1, 0)])
4. line = branch([(1, 1), (2, 2)])
5. line = Punctum(1, 1)

Виберіть рядок коду в якому демонстується функція для створення Полігону бібліотеки shapely

1. polygon = Polygon([(0, 0), 0, 1), (1, 1), (1, 0)])
2. polygon = polygon1.union(polygon2)
3. polygon = PolygonString([(0, 0), 0, 1), (1, 1), (1, 0)])
4. polygon = Line([(1, 1), (2, 2)])
5. polygon = Point(1, 1)

Виберіть рядок коду в якому демонстується функція для створення Обєднання полігонів бібліотеки shapely

1. union\_polygon = polygon1.union(polygon2)
2. union\_polygon = MultiPolygon([polygon1, polygon2])
3. union\_polygon = PolygonString([(0, 0), 0, 1), (1, 1), (1, 0)])
4. union\_polygon = LineString([(1, 1), (2, 2)])
5. union\_polygon = Point(1, 1)

Недоліки бібліотеки shapely

1. Не має функцій відображення геометрій
2. Не має можливостей для роботи проекції та координатні системи
3. При кожній операції з растровими даними, такі як збільшення, зменшення, обрізання, розмивання або фільтрування, може відбуватися погіршення якості даних
4. Дозволяють проводити складний геоінформаційний аналіз, використовуючи різноманітні аналітичні методи
5. Використовується для виконання операції з колекціями, такі як об'єднання, перетин та інші

Виберіть рядок коду в якому демонстується функція для створення Обєднання полігонів бібліотеки shapely

1. multi\_polygon = MultiPolygon([polygon1, polygon2])
2. multi\_polygon = LineString([(1, 1), (2, 2)])
3. multi\_polygon = PolygonString([(0, 0), 0, 1), (1, 1), (1, 0)])
4. multi\_polygon = Point(1, 1)
5. multi\_polygon = polygon1.union(polygon2)

На яких бібліотеках базується робота Geopandas

1. Pandas, Shapely та Fiona
2. Pandas, GDAL, Rasterio
3. GDAL, Rasterio, Fiona
4. Shapely, Rasterio, Fiona
5. Pandas, Fiona, GDAL

З якими форматами дозволяє читати бібліотека Geopandas

1. CSV, GeoJSON, Shapefile
2. KML, GML, Shapefile
3. Fiona, GeoJSON, Shapefile
4. CSV, KML, GML
5. GeoJSON, GML, Shapefile

Виберіть рядок в якому зазначено команду для читання Shapefile файлу

1. shapefile = gpd.read\_file('shapefile.shp')
2. shapefile.read(window[cut.width; cut.height])
3. shapefile.window[read]
4. shapefile. gpd.read\_file('shapefile.shp')
5. shapefile. read\_file('shapefile.shp')

Виберіть рядок в якому зазначено команду для збереження даних в Shapefile файл

1. shapefile.to\_file('new\_shapefile.shp', driver='ESRI Shapefile')
2. shapefile = gpd.read\_file('shapefile.shp')
3. shapefile. gpd.read\_file('shapefile.shp')
4. shapefile. read\_file('shapefile.shp')
5. shapefile = to\_file('new\_shapefile.shp', driver='ESRI Shapefile')

Виберіть рядок в якому зазначено команду для обєднання двох шейпфайлів

1. union\_shapefile = gpd.overlay(shapefile1, shapefile2, how='union')
2. union\_shapefile = shapefile.attribute\_table()
3. union\_shapefile = gpd.read\_file('shapefile.shp')
4. union\_shapefile = shapefile.to\_file('new\_shapefile.shp', driver='ESRI Shapefile')
5. union\_shapefile = shapefile.area

Виберіть рядок в якому зазначено команду для обчислення площи полігону

1. area = shapefile.area
2. area = shapefile.length
3. shapefile.area(‘’)
4. shapefile.length()
5. area = shapefile.area(‘’)

Виберіть рядок в якому зазначено команду для обчислення периметра полігону

1. Perimeter = shapefile.area
2. Perimeter = shapefile.length
3. shapefile.area(‘’)
4. shapefile.length()
5. perimeter = shapefile.length()

Виберіть рядок в якому зазначено команду для створення множини ліній

1. lines = gpd.GeoSeries([line1, line2])
2. point = Point(1, 0)
3. nearest\_point, \_ = lines.geometry[nearest].nearest\_points(point)
4. result = gdf.groupby('column\_name').mean()
5. gdf = gpd.read\_file('file.shp')

Виберіть рядок в якому зазначено команду для групування та обчислення середнього значення

1. result = gdf.groupby('column\_name').mean()
2. lines = gpd.GeoSeries([line1, line2])
3. gdf = gpd.read\_file('file.shp')
4. polygon = point.buffer(1)
5. data = gpd.read\_file('data.shp')

Виберіть рядок в якому зазначено команду для завантаження геоданих

1. data = gpd.read\_file('data.shp')
2. lines = gpd.GeoSeries([line1, line2])
3. result = gdf.groupby('column\_name').mean()
4. Perimeter = shapefile.area
5. gdf = gpd.read\_file('file.shp')

Виберіть рядок в якому зазначено команду знаходження найближчого сусіда до точки з множини ліній

1. nearest\_point, \_ = lines.geometry[nearest].nearest\_points(point)
2. lines = gpd.GeoSeries([line1, line2])
3. result = gdf.groupby('column\_name').mean()
4. Perimeter = shapefile.area
5. gdf = gpd.read\_file('file.shp')

Виберіть рядок в якому зазначено команду для відображення геоданих на карті (import geopandas as gpd)

1. gdf.plot()
2. gdf = gpd.read\_file('file.shp')
3. point = Point(2.2, 4.2)
4. result = gdf[gdf.contains(point)]
5. lines = gpd.GeoSeries([line1, line2])

Виберіть рядок в якому зазначено команду для знаходження полігонів що містять точку

1. result = gdf[gdf.contains(point)]
2. lines = gpd.GeoSeries([line1, line2])
3. gdf = gpd.read\_file('file.shp')
4. polygon = point.buffer(1)
5. data = gpd.read\_file('data.shp')