

Universidade Federal do ABC
Curso de Especialização em Geoprocessamento
Disciplina de Introdução a programação para geoprocessamento

Atividade complementar 1

Prof. Alexandro Schafer

Seja bem-vindo(a) à primeira atividade complementar da disciplina **Introdução à Programação para Geoprocessamento!**

Nesta etapa, você terá a oportunidade de aplicar os conceitos básicos que estudamos até agora, reforçando seu aprendizado e desenvolvendo suas habilidades em programação voltadas para geoprocessamento.

A atividade complementar 1 não compõe o conceito final da disciplina.

Lembre-se de seguir as instruções com atenção e, se surgir qualquer dúvida, não hesite em procurar o tutor do seu polo para orientação.

Instruções Gerais:

1. *Criação do Notebook:* Você deve criar um *notebook* no Google Colab para resolver os exercícios listados.
2. *Envio do Notebook:* Após concluir os exercícios, o notebook deve ser enviado pelo *Moodle*, dentro do prazo estipulado no *mapa de atividades* da disciplina.

Desejo a você um bom trabalho e sucesso na realização da atividade!

Prof. Alexandro Schafer

Exercício 1: Variáveis e Impressão de Dados

Neste exercício, você deve criar uma variável e exibir o seu conteúdo na tela. Para tanto, siga as seguintes etapas:

- a) Crie uma variável chamada `cidade` e atribua o nome de um município a ela.
- b) Mostre na tela o valor que está dentro da variável que você criou. Para isso, use a função `print()`.

Exercício 2: Entrada de Dados

Neste exercício, você deve pedir a um usuário para inserir um valor e exibi-lo na tela. Para tanto, siga as seguintes etapas:

- a) Crie a variável `area`. O texto que deve aparecer em tela para o usuário é "Digite área do município em km2:". Use a função `input()` nesta linha de código.
- b) Mostre na tela o valor que foi informado pelo usuário. Para isso, use a função `print()`.

Exercício 3: Tipos de Dados – Inteiros e Floats

Neste exercício, você deve criar duas variáveis para armazenar dados de um município e, em seguida, exibi-las na tela. Para tanto, siga as seguintes etapas:

- a) Crie a variável `habitantes` para armazenar o número de habitantes do município (um número inteiro). O texto que deve aparecer na tela para o usuário é "Digite o número de habitantes do município:". Use a função `input()` nesta linha de código.

b) Crie a variável `altitude` para armazenar a altitude média da cidade em metros (um número decimal). O texto que deve aparecer na tela para o usuário é "Digite a altitude média da cidade em metros:". Use a função `input()` nesta linha de código.

c) Mostre na tela os valores que foram informados pelo usuário. Para isso, use a função `print()`.

Exercício 4: Operadores Aritméticos

Neste exercício, você deve criar variáveis para armazenar dados da população e calcular a população total do município. Para tanto, siga as seguintes etapas:

a) Crie a variável `populacao_urbana` para armazenar o número de habitantes da área urbana. O texto que deve aparecer na tela para o usuário é "Digite o número de habitantes da área urbana:". Use a função `input()` nesta linha de código.

b) Crie a variável `populacao_rural` para armazenar o número de habitantes da área rural. O texto que deve aparecer na tela para o usuário é "Digite o número de habitantes da área rural:". Use a função `input()` nesta linha de código.

c) Calcule a população total do município somando as duas variáveis que você criou (`populacao_urbana` e `populacao_rural`) e armazene o resultado em uma nova variável chamada `populacao_total`.

d) Mostre na tela o valor da população total. Para isso, use a função `print()`.

Exercício 5: Booleanos – Comparação de Números

Neste exercício, você deve criar duas variáveis para armazenar áreas e verificar a relação entre elas. Para tanto, siga as seguintes etapas:

- a) Crie a variável `area1` com o valor de 250 km² e a variável `area2` com o valor de 150 km².
- b) Verifique se a área armazenada em `area1` é maior que a área armazenada em `area2`, utilizando o operador de comparação `>`.
- c) Mostre na tela o resultado da comparação (True ou False). Para isso, use a função `print()`.

Exercício 6: Listas – Criando e Acessando Elementos

Neste exercício, você deve criar uma lista de biomas e exibir um dos itens da lista. Para tanto, siga as seguintes etapas:

- a) Crie uma lista chamada `biomas` que contenha três nomes de biomas brasileiros: "Amazônia", "Cerrado" e "Pantanal".
- b) Acesse e mostre na tela o segundo bioma da lista. Lembre-se de que a contagem de índices no Python começa a partir de 0. Use a função `print()` para exibir o valor.

Exercício 7: Usando `append()` para Adicionar um Novo Bioma

Neste exercício, você deve adicionar um novo bioma à lista criada no exercício anterior. Para tanto, siga as seguintes etapas:

- a) Use a função `append()` para adicionar o bioma "Mata Atlântica" à lista de biomas criada no exercício anterior.
- b) Mostre na tela a lista completa de biomas, incluindo o novo bioma que foi adicionado. Use a função `print()` para exibir a lista.

Exercício 8: Contando Elementos em uma Lista

Neste exercício, você deve verificar quantos biomas estão armazenados na lista que foi criada anteriormente. Para tanto, siga as seguintes etapas:

- a) Use a função `len()` para contar quantos elementos existem na lista `biomas` criada nos exercícios anteriores.
- b) Mostre na tela o número de elementos presentes na lista. Para isso, use a função `print()`.

Exercício 9: Armazenamento de Coordenadas Geográficas em uma Tupla

Neste exercício, você deve armazenar as coordenadas geográficas de Brasília em uma tupla. Para tanto, siga as seguintes etapas:

- a) Crie uma tupla chamada `coordenadas` que armazene a latitude e a longitude de Brasília. As coordenadas são aproximadamente: *latitude -15.7939* e *longitude -47.8828*.

b) Mostre na tela a tupla contendo as coordenadas geográficas. Para isso, use a função `print()`.

Exercício 10: Criação e Manipulação de um Dicionário

Neste exercício, você deve criar e manipular um dicionário para armazenar informações sobre um município. Para tanto, siga as seguintes etapas:

a) Crie um dicionário chamado `dados_municipio` para armazenar as seguintes informações:

- **nome:** "São Paulo"
- **populacao:** 12325232
- **area_km2:** 1521.11

b) Adicione um novo par chave-valor ao dicionário chamado `densidade_demografica`. O valor dessa chave deve ser calculado dividindo a população pela área (população/área).

c) Mostre na tela o dicionário completo com todos os pares chave-valor. Para isso, use a função `print()`.