**Atividade de Revisão - Tipos de Dados, Coleções, Estruturas Condicionais, Estruturas de Repetição.**

**Instruções:**

Para realizar a atividade, você deve seguir as seguintes instruções:

1. Crie um novo repositório no GitHub para a atividade.
2. Formem duplas e decidam quem será o piloto (P) e o observador (O).
3. O professor sinalizará para que as funções sejam invertidas a cada 10 minutos.
4. Cada questão deve ser resolvida em um arquivo .py separado.
5. Armazenem os arquivos no repositório do GitHub da dupla.
6. Enviem o link do repositório para "tarikponciano@gmail.com" até o final da aula.
7. A premiação será concedida à dupla que realizar todas as questões corretamente ou ao maior número de questões realizadas até o fim da aula.
8. Duplas que completarem pelo menos 2 questões de cada conteúdo receberão um chocolate bis por aluno.
9. Pesquisa e uso de ferramentas adequadas são permitidos.anças para o repositório remoto no GitHub usando o comando git push origin main.
10. Verifique se as respostas foram enviadas corretamente no repositório remoto no GitHub.

Lembre-se de utilizar boas práticas de programação, como comentar o código e usar nomes de variáveis descritivos. Além disso, é importante responder, quando possível, pelo menos 3 questões de cada seção para demonstrar seu conhecimento em diferentes áreas da programação em Python. Boa sorte!

**Checklist de Conhecimentos:**

Utilize a lista de conhecimentos abaixo para identificar aspectos das aulas anteriores que precisam ser revisados.

**Tipos de Dados:**

* Entender os tipos básicos de dados em Python (int, float, bool, str)
* Conhecer os tipos compostos de dados em Python (list, tuple, set, dict)
* Compreender as diferenças entre tipos mutáveis e imutáveis em Python
* Saber converter entre tipos de dados em Python

**Coleções:**

* Saber criar e manipular listas, tuplas, sets e dicionários
* Entender como funcionam as operações de acesso, inserção e remoção de elementos em cada tipo de coleção
* Compreender as diferenças entre os tipos de coleções em Python
* Saber como usar a compreensão de listas e dicionários em Python

**Estruturas Condicionais:**

* Saber criar e usar as estruturas condicionais if/else e elif em Python
* Entender como usar operadores lógicos (and, or, not) em Python
* Saber como usar operadores de comparação (==, !=, <, >, <=, >=) em Python

**Estruturas de Repetição:**

* Saber criar e usar as estruturas de repetição for e while em Python
* Entender como usar a instrução "break" para interromper um loop em Python
* Saber como usar a instrução "continue" para pular para a próxima iteração de um loop em Python

Lembre-se que esta é apenas uma lista básica de conhecimentos e que há muitos outros conceitos importantes em Python que podem ser explorados!

**Lista de Questões:**

**Tipos de Dados**

**Questão 1:**

Escreva um programa que solicite ao usuário um número inteiro e um número de ponto flutuante. Depois, exiba a soma desses dois números.

**Questão 2:**

Crie um programa que peça o nome e a idade do usuário e exiba uma mensagem personalizada de boas-vindas.

**Questão 3:**

Desenvolva um programa que solicite ao usuário uma palavra e exiba quantas letras a palavra possui.

**Questão 4:**

Escreva um programa que peça ao usuário seu ano de nascimento e calcule quantos anos ele terá em 2030.

**Questão 5:**

Crie um programa que leia um número inteiro e verifique se ele é par ou ímpar. Exiba uma mensagem apropriada.

**Estruturas Condicionais**

**Questão 1:**

Elabore um programa que peça ao usuário uma temperatura em Celsius e a converta para Fahrenheit utilizando a fórmula: F = C \* 9/5 + 32.

**Questão 2:**

Crie um programa que solicite a idade do usuário e determine se ele é uma criança, adolescente, adulto ou idoso.

**Questão 3:**

Escreva um programa que verifique se um número digitado pelo usuário é positivo, negativo ou zero.

**Questão 4:**

Faça um programa que peça para o usuário inserir dois números, pergunte se ele quer realizar a operação de adição ou de subtração e, que a partir desta escolha, mostre o resultado na tela.

**Questão 5:**

Crie um programa que peça ao usuário um número inteiro e verifique se ele é divisível por 3 e 5 ao mesmo tempo.

**Estruturas de Repetição**

**Questão 1:**

Elabore um programa que exiba a sequência de números pares de 0 a 20 utilizando um loop for.

**Questão 2:**

Crie um programa que solicite ao usuário um número inteiro e exiba a tabuada desse número utilizando um loop while.

**Questão 3:**

Escreva um programa que calcule a média de 5 números digitados pelo usuário utilizando um loop for.

**Questão 4:**

Desenvolva um programa que peça ao usuário um número e imprima todos os números primos até esse número utilizando um loop while.

**Questão 5:**

Elabore um programa que simule um jogo de adivinhação. O programa deve sortear um número entre 1 e 100 e permitir que o usuário tente adivinhar até acertar.

**Coleções**

**Questão 1:**

Crie um programa que solicite ao usuário três nomes de frutas e armazene-os em uma lista. Em seguida, exiba os nomes das frutas na tela.

**Questão 2:**

Desenvolva um programa que peça ao usuário cinco números inteiros e armazene-os em uma lista. Depois, exiba os números em ordem inversa.

**Questão 3:**

Escreva um programa que simule uma lista de compras. O programa deve permitir que o usuário adicione itens à lista e, no final, exiba todos os itens da lista.

**Questão 4:**

Elabore um programa que utilize um dicionário para armazenar o nome e a idade de três amigos. Depois, exiba as informações de cada amigo na tela.

**Questão 5:**

Crie um programa que leia o nome de cinco cidades digitadas pelo usuário e armazene-as em uma lista. Em seguida, exiba as cidades em ordem alfabética.

**Dicas**

**Tipos de Dados:**

Verifique os tipos de dados necessários para cada entrada do usuário (int, float, str).

Certifique-se de converter os tipos de dados adequadamente ao ler e exibir informações.

**Estruturas de Condição:**

Use instruções "if", "elif" e "else" para verificar as condições e tomar decisões baseadas nos valores.

Lembre-se de indentar corretamente o código dentro dos blocos de condição.

**Estruturas de Repetição:**

Use loops "for" e "while" para repetir a execução do código.

Certifique-se de inicializar variáveis de controle antes de usar em loops.

**Listas e Dicionários:**

Use métodos como "append()" para adicionar elementos a listas.

Utilize índices corretos para acessar elementos de listas.

Para dicionários, lembre-se da estrutura chave-valor e como acessar valores usando chaves.

Utilize laços para percorrer listas e dicionários para processar informações.

**Dicas Gerais:**

Utilize nomes de variáveis descritivos para facilitar a compreensão do código.

Quebre o problema em partes menores e resolva cada parte antes de avançar.

Comente seu código para explicar partes complexas ou detalhes importantes.

Teste seu código com diferentes entradas para garantir que ele funcione corretamente.