

[Modo de Exibição Acessibilidade](#)

Compartilhar

Trabalho Prático | DGT2817 LÓGICA,
ALGORITMOS E PROGRAMAÇÃO DE
COMPUTADORES

Material de **orientações** para desenvolvimento do **Trabalho Prático** da disciplina DGT2817 Lógica, Algoritmos e Programação de Computadores..

ATENÇÃO: a nota deste trabalho prático representa 100% da nota desta disciplina. Esta disciplina NÃO tem simulados ou alternativas de recuperação de nota. Desta forma, interaja com o tutor via fórum da SAVA quantas vezes forem necessárias para você obter o entendimento necessário para o desenvolvimento deste trabalho. Fique atento ao prazo de entrega.

Objetivos da prática

- Descrever a utilização das estruturas de condição if e else em Python;
- Descrever a utilização da estrutura de condição else if (elif) em Python;
- Descrever a utilização da estrutura de repetição while em Python;
- Descrever a utilização da estrutura de repetição for em Python;
- Descrever a utilização de funções em Python;
- Descrever a utilização argumentos de funções no Python;
- Refazer a calculadora utilizando estrutura condicional e funções;

Entrega e Progresso

- As microatividades irão dar suporte para o desenvolvimento do Trabalho Prático. Elas têm apoio/gabarito para resolução no próprio documento;
- Teremos uma entrega intermediária das Microatividades 1-2-3 e a entrega final da disciplina que é o Trabalho Prático;
- Após concluir as microatividades, compartilhe com o professor o link do GITHUB
- **ATENÇÃO: a nota deste trabalho representa 100% da nota desta disciplina.**

- Esta disciplina NÃO tem simulados ou alternativas de recuperação de nota.
- Desta forma, interaja com o tutor via fórum da SAVA quantas vezes forem
- necessárias para você obter o entendimento necessário para o
- desenvolvimento deste trabalho. Fique atento ao prazo de entrega.

Atividades práticas

Microatividade 1: Descrever a utilização das estruturas de condição if e else em Python

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

- Procedimentos

1. Abra a IDE VS Code;
2. No menu File, selecione a opção “Open Folder”;
3. Crie uma nova pasta em seu computador para armazenar os códigos desse conjunto de microatividades e a selecione a partir do VS Code;
4. No VS Code, menu Explorer, clique na pasta do projeto e crie um novo arquivo/script chamado “estruturas_condicao1.py”;
5. No script criado:
 1. Crie uma variável chamada temperatura e atribua a ela o valor 29;
 2. Crie uma verificação, utilizando a condição if, para checar se o valor da
 3. variável temperatura é menor que 30;
 4. Caso positivo, imprima na tela o texto ‘A temperatura hoje está amena’;
 5. Caso contrário, e utilizando a condição else, imprima na tela o texto ‘Hoje
 6. está fazendo calor’;
6. Salve o arquivo/script;
7. Na barra superior direita você verá um ícone no formato de um triângulo deitado, semelhante à imagem abaixo:
8. Clique no ícone. A seguir, deverá ser aberta uma janela, na parte inferior do

VS Code, exibindo o terminal e a saída dos comandos acima, que é a exibição da frase digitada após o comando “print”;

9. Caso ocorra algum erro na execução do script, verifique o seu código, corrigindo eventuais erros, e tente executá-lo novamente;
10. Ainda no script, altere o valor da variável temperatura para 31, salve a alteração e execute novamente o script.

- Resultados esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos sobre as estruturas de condição if e else e se é capaz de criar instruções simples utilizando-as.

Microatividade 2: Descrever a utilização da estrutura de condição else if (elif) em Python

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

- Procedimentos:

1. Abra a IDE VS Code;
2. Na mesma pasta utilizada na microatividade anterior, crie um novo arquivo/
3. script chamado “estruturas_condicao2.py”;
4. Nesse novo script:
 1. Crie uma variável chamada tempoExperiencia e atribua a ela o valor 5;
 2. Crie uma verificação, utilizando a condição if, para checar se o valor da
 3. variável tempoExperiencia é menor que 2;
 4. Caso positivo, imprima na tela o texto ‘Nível de conhecimento júnior.’;
 5. Após as instruções acima, crie uma outra condição utilizando elif (else if)
 6. para verificar se o valor da variável tempoExperiencia é maior que 2 e
 7. menor que 5. Em caso positivo, imprima o texto ‘Nível de conhecimento
 8. pleno.’

9. Por fim, crie uma condição else e imprima o texto 'Nível de conhecimento
10. sênior.';
5. Salve o arquivo/script e o execute;
6. Altere o script, modificando o valor da variável tempoExperiencia para 1.
7. Salve e execute;
8. Por fim, altere novamente o script, modificando o valor da variável
9. tempoExperiencia para 3. Salve e execute.

- Resultados esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos sobre as estruturas de condição if, else if e else e se é capaz de criar instruções simples utilizando-as.

Microatividade 3: Descrever a utilização da estrutura de repetição while em Python

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

- Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta utilizada nas microatividades anteriores, crie um novo
3. arquivo/script chamado "estruturas_repeticao1.py";5
4. Nesse novo script:
 1. Crie uma variável chamada entrada_idade e atribua a ela o valor "";
 2. Crie uma instrução while que verifique se o valor atribuído à variável
 3. entrada_idade é diferente de 0 (como o valor inicial atribuído à variável é
 4. "", isso a definiu como tipo string. Logo, a verificação no While deve ser
 5. feita com auxílio da instrução str);
 6. No escopo da instrução while, atribua à variável entrada_idade um input
 7. de entrada de dados com o texto 'Digite um número qualquer ou 0 para

8. sair: ‘;
 9. Imprima, na tela, o número digitado pelo usuário precedido do texto
 10. ‘Número digitado: ‘;
5. Salve o arquivo/script e o execute;
 6. Teste diferentes valores como entrada de dados, incluindo o número 0 - que
 7. deverá fazer com que a execução do programa seja interrompida. Caso isso
 8. não ocorra, verifique seu código – sobretudo a comparação na instrução
 9. While.

- Resultados esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos sobre a estrutura de repetição while, sendo capaz de utilizá-la num programa simples.

Microatividade 4: Descrever a utilização da estrutura de repetição for em Python

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

- Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta utilizada nas microatividades anteriores, crie um novo
3. arquivo/script chamado “estruturas_repeticao2.py”;
4. Nesse novo script:
 1. Crie uma variável chamada texto e atribua a ela o valor ‘Olá, laço for.’;
 2. Crie uma instrução for que itere sobre a variável texto atribuindo cada um
 3. de seus caracteres a uma variável chamada item;
 4. Imprima, na tela, dentro do escopo do laço for, o valor da variável item
 5. precedido do texto ‘Caractere: ‘;

5. Salve o arquivo/script e o execute;
6. Crie, no mesmo script, uma nova instrução for que:
 1. Itere sobre um intervalo numérico entre 1 e 10 (dica: use a instrução
 2. range);
 3. Imprima, na tela, dentro do escopo do laço for, o valor de cada número no
 4. intervalo acima precedido do texto 'Número do intervalo: ' ;
 5. Lembre-se de utilizar a instrução str para concatenar o valor inteiro com a
 6. string no momento de imprimir o valor pedido na tela.

- Resultados Esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos sobre a estrutura de repetição for e se é capaz de usá-la em diferentes cenários.

Microatividade 5: Descrever a utilização de funções em Python

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

- Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta utilizada nas microatividades anteriores, crie um novo
3. arquivo/script chamado "funcoes1.py";
4. No script recém criado:
 1. Defina uma função chamada "imprimir_variavel". Tal instrução não
 2. receberá parâmetros;
 3. No escopo da função acima, crie uma variável chamada texto e atribua a

4. ela o valor 'Olá, funções em Python';
 5. A seguir, ainda no escopo da função, imprima na tela o valor da variável
 6. texto;
 7. Por último, fora do escopo da função, faça a chamada da mesma.
5. Salve o arquivo/script e o execute. Você deverá ver na tela o conteúdo da
 6. variável texto.

- Resultados esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos básicos sobre a declaração e chamada de funções na linguagem Python.

Microatividade 6: Descrever a utilização

argumentos de funções no Python

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

- Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta utilizada nas microatividades anteriores, crie um novo
3. arquivo/script chamado "funcoes2.py";
4. No script acima:
 1. Defina uma função chamada "loginUsuario". Tal instrução receberá cini
 2. parâmetro a variável perfil;
 3. No escopo da função, verifique se o valor do parâmetro perfil é igual a
 4. 'admin'. Dica: considere que o usuário poderá digitar letras maiúsculas e/
 5. ou minúsculas na entrada de dados. Portanto, utilize a instrução lower no
 6. momento de fazer a verificação;
 7. Caso o valor do parâmetro seja igual a 'admin', imprima na tela: 'Bem-

8. vindo, Administrador'. Do contrário, imprima: 'Bem-vindo, Usuário';
9. Por último, fora do escopo da função, faça a chamada da mesma
10. passando diferentes valores como parâmetro. Ex:
 1. Admin
 2. admin
 3. User
 4. usuário
 5. etc.
5. Salve o arquivo/script a cada alteração no valor do parâmetro na chamada da
6. função e o execute.

- Resultados esperados

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos básicos sobre a declaração e chamada de funções com a utilização de parâmetros na linguagem Python.

Trabalho Prático

Através dessa atividade o aluno irá refatorar o código de uma calculadora, acrescentando ao mesmo alguns dos recursos da linguagem Python vistos ao longo das microatividades.

Contextualização

Após o lançamento da primeira versão da calculadora (desenvolvida por você anteriormente), será necessário refatorar a aplicação, modificando e otimizando fragmentos do código a partir do uso de recursos como estruturas de condição, estruturas de repetição e funções. Todos os requisitos para essa tarefa podem ser encontrados no Roteiro de Prática.

Roteiro de prática

- Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

- Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta onde criou os scripts utilizados nas microatividades, crie um
3. novo script chamado "calculadora_v2.py";
4. No script:
 1. Crie uma variável chamada `saida` e atribua a ela o valor "";
 2. Crie uma função chamada `adicao`. Tal função deverá receber dois
 3. parâmetros e retornar a soma entre ambos;
 4. Crie uma função chamada `subtracao`. Tal função deverá receber dois
 5. parâmetros e retornar a subtração entre ambos;
 6. Crie uma função chamada `multiplicacao`. Tal função deverá receber dois
 7. parâmetros e retornar a multiplicação entre ambos;
 8. Crie uma função chamada `divisao`. Tal função deverá receber dois
 9. parâmetros, verificar se um deles é igual a 0. Em caso positivo, deverá
 10. retornar a mensagem "Não foi possível realizar a divisão por 0". Em caso
 11. negativo, deverá retornar a divisão entre ambos;
 12. Crie uma função chamada `calculadora`. Tal função deverá receber três
 13. parâmetros, sendo eles: os dois números que serão usados para os
 14. cálculos e a operação matemática que se deseja realizar. Sobre esse
 15. último parâmetro, você poderá utilizar tanto o sinal da operação quanto o
 16. seu nome;
 17. No corpo da função `calculadora` você deverá verificar qual a operação
 18. desejada pelo usuário, checando o valor do parâmetro correspondente.
 19. Utilize estruturas de condição para isso e, dependendo da operação
 20. desejada, você deverá chamar a função relativa a ela, passando as
 21. variáveis contendo os dois números para serem utilizados no cálculo.
 22. Armazene o resultado da chamada às funções de cálculo numa variável
 23. chamada `resultado`. Ao final da função `calculadora` você deverá retornar a

24. variável resultado;
 25. Crie um laço while e, como condição do mesmo, verifique se o valor da
 26. variável saída é diferente de n. Lembre-se de que o usuário poderá inserir
 27. tanto N quanto n;
 28. No escopo do laço while peça ao usuário para digitar o primeiro número e
 29. armazene seu valor numa variável. Faça o mesmo para o segundo número
 30. e para a operação matemática. Passe essas três variáveis para o método
 31. calculadora, armazenando o retorno dessa chamada numa variável
 32. também chamada resultado. Imprima na tela o valor da variável resultado
 33. precedido pelo texto 'Resultado da operação: '. Por fim, pergunte ao
 34. usuário se ele deseja continuar ou não executando o programa. Armazene
 35. tal input na variável saída;
 36. Tome cuidado com a condição de verificação do laço for em relação à
 37. entrada do usuário armazenada na variável saída. Em outras palavras,
 38. deixe claro para o usuário as respostas possíveis para a pergunta se ele
 39. deseja sair. Use, por exemplo, S/N. Com isso você poderá considerar um
 40. desses dois valores na verificação do laço para saber se deve continuar
 41. executando o programa ou se deve encerrá-lo.
5. Salve as alterações no script e o execute via VS Code;
 6. Teste o aplicativo interagindo com ele através do prompt, fornecendo os
 7. dados necessários para a sua execução.

- Resultados esperados ✨

Os passos anteriores visam a construção de um aplicativo completo com a estrutura básica do Flutter, garantindo que as interfaces de usuário sejam responsivas e acessíveis em dispositivos móveis, tablets e desktops. Essa plataforma permitirá que os clientes compartilhem avaliações e comentários sobre suas experiências de viagem. Além disso, a Agência de viagens utilizará widgets interativos do Flutter para exibir as avaliações dos clientes e interagir com eles de forma envolvente, compartilhando suas opiniões e proporcionando


uma experiência interativa e informativa aos usuários do site. A imagem a seguir ilustra um exemplo de aplicativo desenvolvido. Adapte os códigos apresentados para que você possa criar a sua solução.

Referências


Não foram utilizadas referências bibliográficas para a elaboração das atividades.


Entrega da prática

Chegou a hora, gamer!

 Armazene o projeto em um repositório no GIT.

 Anexar a documentação do projeto (PDF) no GIT.

 Compartilhe o link do repositório do GIT com o seu tutor para correção da prática, por meio da **Sala de Aula Virtual**, na aba "**Trabalhos**" do respectivo nível de conhecimento.

 **Ei, não se esqueça de entregar este trabalho na data estipulada na aba TRABALHOS da SAVA!**

[Feito com o Microsoft Sway](#)

Crie e compartilhe apresentações, histórias pessoais, relatórios interativos e muito mais.

Introdução

Exibir mais

ao rolar para baixo ou passar o dedo.