

PYTHON PARA QAs



REVISÃO AULA 2

- **Tipos de Dados:** int, float, str, bool.
- **Operadores:**
 - **Aritméticos** (+, -, *, /)
 - **Comparativos** (>, <, ==)
 - **Lógicos** (and, or, not)
- **Estruturas Condicionais:** Como decidir o fluxo do código (if, elif, else).
- **Manipulação de Strings:** Métodos úteis como upper(), lower(), strip(), replace().

MÓDULO 2 - ESTRUTURAS DE CONTROLE E COLEÇÕES

1. Estruturas de Decisão (Simples, Aninhadas e Compostas);
2. Estruturas de Repetição (Loops: for, while);
3. Coleções:
 - Listas;
 - Tuplas;
 - Conjuntos;
 - Dicionários.
4. Compreensão de listas e iterações avançadas.

AULA 3 - ESTRUTURAS DE CONTROLE E COLEÇÕES

Objetivos da aula:

- Compreender e aplicar estruturas de decisão (if, elif, else)
- Utilizar estruturas de repetição (for e while)
- Explorar coleções de dados em Python:
 - Listas
 - Tuplas
 - Dicionários
 - Conjuntos

ESTRUTURAS DE DECISÃO

- As estruturas condicionais permitem que o código tome decisões com base em condições.
- Exemplo de Fluxo de Decisão:
 - Entrada: Usuário digita uma senha
 - Processamento: Se a senha for correta, permitir acesso senão não permitir acesso
 - Saída: "Acesso concedido" ou "Acesso negado"

ESTRUTURAS DE REPETIÇÃO

- As estruturas de repetição (ou loops) são usadas para executar um bloco de código múltiplas vezes, economizando esforço e evitando repetições desnecessárias.
- Tipos:
 - **for**: Usado quando sabemos quantas vezes queremos repetir um bloco de código ou quando queremos percorrer elementos de uma coleção como listas, tuplas, dicionários e conjuntos.
 - **while**: Usado quando não sabemos exatamente quantas vezes o código será repetido, mas queremos que ele continue executando enquanto uma condição for verdadeira.

QUANDO USAR CADA UM?

Cenário	Melhor Opção
Percorrer listas, strings, dicionários	<code>for</code>
Repetir algo um número específico de vezes	<code>for</code> com <code>range()</code>
Executar até que uma condição seja atendida	<code>while</code>
Criar menus interativos ou loops de espera	<code>while</code>

INTRODUÇÃO ÀS COLEÇÕES

- As coleções armazenam múltiplos valores em uma única variável.
- Tipos:
 - **Listas:** Mutáveis e ordenadas.
 - **Tuplas:** Imutáveis e ordenadas.
 - **Conjuntos:** Não permitem valores duplicados.

INTRODUÇÃO ÀS COLEÇÕES - LISTAS (list)

- As listas são estruturas de dados mutáveis que armazenam uma coleção ordenada de elementos. Elas permitem valores duplicados e podem conter diferentes tipos de dados (inteiros, strings, floats, booleanos, etc.).
- Exemplo Lista de alunos: ["Ana", "João", "Pedro"]
- As listas mantêm a ordem dos elementos.
- Podemos adicionar, remover e modificar os itens.

INTRODUÇÃO ÀS COLEÇÕES - TUPLAS (tuple)

- As tuplas são estruturas de dados imutáveis, ou seja, não podem ser modificadas após sua criação. São utilizadas quando queremos armazenar dados que não devem ser alterados.
- Exemplo Coordenadas de um ponto: (10, 20)
- Não podemos adicionar ou remover elementos.
- Mais eficientes do que listas em termos de desempenho.

INTRODUÇÃO ÀS COLEÇÕES - DICIONÁRIOS (dict)

- Os dicionários armazenam pares chave-valor, permitindo acesso rápido aos dados por meio da chave, em vez de um índice numérico.
- Exemplo de Informações de um usuário: {"nome": "Carlos", "idade": 30, "email": "carlos@email.com"}
- Cada elemento é um par "chave: valor".
- As chaves devem ser únicas e imutáveis (strings, números ou tuplas).

INTRODUÇÃO ÀS COLEÇÕES - CONJUNTOS (set)

- Os conjuntos armazenam valores únicos, sem duplicatas, e não garantem a ordem dos elementos. Eles são úteis quando precisamos verificar a existência de itens e realizar operações matemáticas como união e interseção.
- Exemplo de Linguagens de programação preferidas:
- {"Python", "Java", "C++", "Python"}
- O conjunto remove automaticamente valores duplicados.
- Muito útil para verificar itens únicos rapidamente.

RESUMO DAS DIFERENÇAS

Estrutura	Características	Permite Duplicatas?	Mutável?	Mantém Ordem?
Lista (list)	Coleção ordenada e mutável	✓ Sim	✓ Sim	✓ Sim
Tupla (tuple)	Coleção ordenada e imutável	✓ Sim	✗ Não	✓ Sim
Dicionário (dict)	Pares chave-valor, busca rápida	✓ Sim	✓ Sim	✓ Sim
Conjunto (set)	Coleção não ordenada e única	✗ Não	✓ Sim	✗ Não

QUANDO USAR CADA ESTRUTURA?

Cenário	Melhor Estrutura
Preciso armazenar uma lista ordenada de itens e poder modificar os elementos.	Lista (list)
Preciso armazenar um conjunto fixo de valores que não serão alterados .	Tupla (tuple)
Preciso armazenar informações associadas a uma chave , como dados de um usuário.	Dicionário (dict)
Preciso armazenar valores únicos e fazer operações matemáticas como união e interseção.	Conjunto (set)

REVISÃO

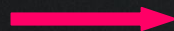
- **ESTRUTURAS DE DECISÃO**
 - **if**
 - **elif**
 - **else**
- **ESTRUTURAS DE REPETIÇÃO**
 - **for**
 - **while**
- **INTRODUÇÃO ÀS COLEÇÕES**
 - **Listas**
 - **Tuplas**
 - **Dicionários**
 - **Conjuntos**

PARA A PRÓXIMA AULA

- Desafio de hoje:
 - Criar um programa que:
 - Leia os nomes de 5 casos de teste.
 - Armazene-os em uma lista.
 - Simule a execução de cada caso, gerando aleatoriamente Pass ou Fail.

Podemos contar com o seu feedback?

Escaneie o QR Code ao lado e responda nossa Pesquisa de Avaliação.



OBRIGADO



/apsferreira



/apsferreira_



alura + FIA/P

PARA EMPRESAS

Copyright © 2019 | Professor (a) Nome do Professor

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.