

Aula 1 – Fundamentos de Python e Condicionais

Desvendando os Primeiros Passos na Programação

1. Introdução ao Python: Sua Jornada Começa Aqui

Python é uma linguagem de programação notavelmente **simples de aprender** e **incrivelmente poderosa**, ideal para quem está começando e para desenvolvedores experientes. Sua sintaxe clara e legível permite que você se concentre na **lógica do seu código**, minimizando complexidades.

Essa versatilidade é o que impulsiona o uso de Python em diversas áreas:



Desenvolvimento Web

Crie desde protótipos rápidos até sistemas robustos e escaláveis para a internet.



Análise de Dados

Processe, visualize e extraia insights valiosos de grandes volumes de informações.



Inteligência Artificial

Desenvolva modelos de aprendizado de máquina, redes neurais e sistemas inteligentes.



Automação de Tarefas

Simplifique rotinas repetitivas, economizando tempo e aumentando sua produtividade.

Nesta aula, nosso foco será nos **fundamentos essenciais** que servirão como uma base sólida para qualquer caminho que você deseje seguir no mundo da programação. Prepare-se para codificar!

2. Variáveis e Tipos de Dados: Os Blocos Fundamentais

Em Python, uma **variável** é como um rótulo ou nome dado a um espaço na memória do computador. Esse espaço é usado para **armazenar valores** que seu programa pode usar e manipular. Pense nelas como caixas onde você guarda diferentes tipos de informações.

Exemplos Práticos:

`idade = 26` # **Inteiro** (int) – Armazena números sem casas decimais.

`altura = 1.75` # **Decimal** (float) – Armazena números com casas decimais.

`nome = "Ana"` # **Texto** (string, str) – Armazena sequências de caracteres.

`estudante = True` # **Booleano** (bool) – Armazena valores lógicos: Verdadeiro ou Falso.



Tipos de Dados Essenciais no Python:

1

Inteiro (int)

Representa números **sem frações**, como **10**, **-5** ou **0**.

10

Decimal (float)

Representa números com **casas decimais**, como **3.14** ou **2.5**.



Texto (str)

Representa **sequências de caracteres**, como **"Python"** ou **"Olá"**.



Booleano (bool)

Representa valores lógicos: **True** (verdadeiro) ou **False** (falso).

Conversão de Tipos:

Muitas vezes, você precisará converter dados de um tipo para outro. Use funções como `int()`, `float()`, `str()` ou `bool()`:

```
numero_texto = "10"
```

```
numero_inteiro = int(numero_texto) # Transforma a string "10" no inteiro 10.
```

```
print(type(numero_inteiro)) # Saída: <class 'int'>
```

3. Entrada e Saída de Dados: Comunicando com o Usuário

Para que seus programas sejam interativos, eles precisam ser capazes de exibir informações e receber dados do usuário. Python oferece funções diretas para isso:

Exibindo Dados: print()

A função `print()` é sua principal ferramenta para **exibir mensagens**, variáveis ou resultados diretamente na tela do console. Ela faz com que seu programa "fale" com quem o está usando.

```
print("Olá, mundo!")  
  
mensagem = "Bem-vindo ao universo Python!"  
  
print(mensagem)
```

Entrada de Dados: input()

A função `input()` permite que seu programa **receba informações** digitadas pelo usuário através do teclado. É importante lembrar que o valor retornado por `input()` é sempre uma **string**, mesmo que o usuário digite um número!

```
nome = input("Digite seu nome: ")  
  
print(f"Olá, {nome}! Que bom te ter por aqui.")
```

Atividade Prática: Apresentação Personalizada

Crie um pequeno programa que use `input()` para perguntar ao usuário seu **nome completo**, **idade** e **altura**. Utilize `print()` e **formatação de strings (f-strings)** para exibir uma mensagem personalizada, apresentando todas as informações coletadas. Não se esqueça de converter os tipos de dados quando necessário!

4. Operadores: Manipulando e Comparando Dados

Operadores são símbolos especiais em Python que realizam operações em valores e variáveis. Eles são a espinha dorsal de qualquer lógica de programação, permitindo cálculos, comparações e avaliações.

Aritméticos

Utilizados para realizar **cálculos matemáticos** básicos e avançados:

- **+** (soma)
- **-** (subtração)
- ***** (multiplicação)
- **/** (divisão **decimal**)
- **//** (divisão **inteira**)
- **%** (resto da divisão)
- ****** (potência)

```
print(10 + 5) # Saída: 15
```

```
print(10 % 3) # Saída: 1 (resto de  
10 dividido por 3)
```

```
print(2 ** 3) # Saída: 8 (2 elevado  
à 3ª potência)
```

Comparação

Comparam dois valores, retornando **True** (verdadeiro) ou **False** (falso):

- **==** (igual a)
- **!=** (diferente de)
- **>** (maior que)
- **<** (menor que)
- **>=** (maior ou igual a)
- **<=** (menor ou igual a)

```
print(5 > 3)
```

```
# Saída: True
```

```
print(5 == 3)
```

```
# Saída: False
```

Lógicos

Combinam ou invertem expressões booleanas, cruciais para a tomada de decisões:

- **and**: Verdadeiro se **TODAS** as condições forem verdadeiras.
- **or**: Verdadeiro se **PELO MENOS UMA** condição for verdadeira.
- **not**: **Inverte** o valor lógico (True vira False, False vira True).

```
print(10 > 5 and 8 > 3)
```

```
# Saída: True
```

```
print(10 > 5 or 8 < 3)
```

```
# Saída: True
```

```
print(not True)
```

```
# Saída: False
```

Atividade Prática: Calculadora de Área de Retângulo

Desenvolva um programa que peça ao usuário a **base** e a **altura** de um retângulo. Em seguida, utilize o operador aritmético correto para **calcular a área** (Base * Altura) e exiba o resultado na tela de forma clara.

5. Estruturas Condicionais: Tomando Decisões no Código

As estruturas condicionais permitem que seu programa se comporte de maneiras diferentes com base em **condições específicas**. Elas são essenciais para criar programas dinâmicos que podem responder a diversas situações e entradas do usuário.

Como funciona: if, elif e else

if: A Condição Inicial

```
if idade <= 10:  
    print("Criança")
```

O bloco de código indentado sob o if é executado **somente se a condição for verdadeira**.

elif: Condições Alternativas

```
elif idade <= 18:  
    print("Adolescente")
```

O elif (else if) testa uma **nova condição** se todas as condições if e elif anteriores forem **falsas**. Você pode usar quantos elif forem necessários.

else: Condição excepcional

```
else:  
    print("Adulto")
```

O bloco de código sob o else é executado **se todas as condições** if e elif anteriores forem **falsas**. Ele captura todos os outros casos.

Atividade Prática: Verificador de Habilitação

Crie um programa que determine se uma pessoa pode dirigir. O programa deve pedir a **idade** do usuário e, em seguida, perguntar se ele **possui carteira de motorista** (o usuário deve responder "sim" ou "não"). Use estruturas condicionais (if, elif, else) e operadores lógicos para verificar se a pessoa é **maior de idade (18 anos ou mais)** e se possui carteira para exibir uma mensagem apropriada.

6. Exercícios Práticos: Solidificando seu Aprendizado

Agora que você explorou os fundamentos de Python, variáveis, operadores e condicionais, é hora de colocar a mão na massa! Estes exercícios foram desenhados para que você possa **praticar e consolidar seus conhecimentos.**

1 Verificador de Número Par ou Ímpar

Peça um número inteiro ao usuário e use uma condição para determinar e exibir se ele é **par** ou **ímpar**. (Dica: o operador de resto `%` será muito útil aqui!).

2 Sistema de Média Escolar

Desenvolva um programa que receba duas notas de um aluno. Calcule a **média** dessas notas. Em seguida, usando condicionais, informe se o aluno está **Aprovado** (média maior ou igual a 6) ou **Reprovado**.

3 Mini-Calculadora Interativa

Crie um programa que receba dois números e pergunte ao usuário qual **operação matemática** (soma, subtração, multiplicação ou divisão) ele deseja realizar. Utilize condicionais para executar a operação escolhida e, por fim, **exiba o resultado.**

A programação é uma jornada de constante aprendizado e descobertas!