

Conceitos básicos de programação

Alex Sandro Costa e Everton Freitas Godinho

Exemplo concreto de um programa

O programa a seguir tem o intuito de calcular o índice de massa corporal do usuário

```
nome = input("Informe o seu nome: ")
```

```
idade = int(input("Informe a sua idade: "))
```

```
altura = float(input("Informe a sua altura: "))
```

```
peso = float(input("Informe o seu peso: "))
```

```
imc = peso / (altura * altura)
```

```
msg = nome + ", " + str(idade) + " anos, o seu índice de massa corporal é: %.2f" % imc + " kg/m²"
```

```
print(msg)
```

Fluxo de execução

- O Python irá inicializar lendo a primeira linha de código
- A partir dela, as linhas posteriores serão lidas uma a uma, sequencialmente, de cima para baixo
- Existem situações em que o programa pode sofrer um desvio na sua execução
- Nesses casos, podemos criar blocos de comandos indentados

Comentários

- Comentários são um recurso nas linguagens de programação para que o desenvolvedor do código fonte possa dar informações sobre o código
- Comentários em Python são feitos iniciando a linha com o símbolo # e tudo descrito após esse símbolo é ignorado pelo Python
- Comentários são muito utilizados para explicar como um bloco de comando funciona ou até mesmo como o programa em si funciona

Ilustrando esses conceitos no exemplo

FLUXO DE EXECUÇÃO

```
1  # O programa a seguir tem o intuito de calcular o índice de massa corporal do usuário } COMENTÁRIO
2
3  nome = input("Informe o seu nome: ")
4
5  idade = int(input("Informe a sua idade: "))
6
7  altura = float(input("Informe a sua altura: "))
8
9  peso = float(input("Informe o seu peso: "))
10
11 imc = peso / (altura * altura)
12
13 msg = nome + ", " + str(idade) + " anos, o seu índice de massa corporal é: %.2f" % imc + " kg/m²"
14
15 print(msg)
```

Variáveis e atribuição

- Em programação, uma variável é um nome no qual atribuímos um determinado valor
- Tal valor pode mudar ao longo da execução de um programa
- O símbolo de atribuição em Python é o símbolo =
- Declaramos variáveis da seguinte forma:
 variável = valor
- É comum dizermos que variável recebe valor

Nome de variáveis

- Existem algumas regras para criar o nome de uma variável:
 - O nome só pode conter letras, algarismos e sublinhas, e nenhum outro símbolo é permitido, nem mesmo espaços e acentos
 - Os nomes são sensíveis a caixa, ou seja, existe diferença entre letras maiúsculas e minúsculas; variavel é diferente de VARIABEL, Variavel e assim por diante
 - Os nomes não podem ser palavras reservadas da linguagem como print, input, def, etc
 - Os nomes devem ser significativos para o melhor entendimento do programa

Ilustrando esses conceitos no exemplo

ATRIBUIÇÃO

nome ⊕ input("Informe o seu nome: ")

NOME VARIÁVEL

VALOR

ATRIBUIÇÃO

idade ⊕ int(input("Informe a sua idade: "))

NOME VARIÁVEL

VALOR

ATRIBUIÇÃO

altura ⊕ float(input("Informe a sua altura: "))

NOME VARIÁVEL

VALOR

ATRIBUIÇÃO

peso ⊕ float(input("Informe o seu peso: "))

NOME VARIÁVEL

VALOR

ATRIBUIÇÃO

imc ⊕ peso / (altura * altura)

NOME VARIÁVEL

VALOR

ATRIBUIÇÃO

msg ⊕ nome + ", " + str(idade) + " anos, o seu índice de massa corporal é: %.2f" % imc + " kg/m²"

NOME VARIÁVEL

VALOR

Ilustrando esses conceitos no exemplo

Exemplos de nomes de variáveis inválidas

`nome usuário = "Alex" # nome da variável possui espaço e acento`

`id@de = 22 # nome da variável possui o símbolo @ que é inválido`

`1altura = 1.74 # nome da variável não começa com letra ou sublinha`

`pe$o = 70 # nome da variável possui o símbolo $ que é inválido`

Tipos de dados

- . Toda variável possui um tipo associado a ela para que o Python seja capaz de realizar as operações corretas
- . Existem três tipos principais:
 - . Numéricos (inteiros, reais, etc)
3, -15, +1001, -2.71828, 3.14159, 141.5
 - . Textuais (caracteres e cadeia de caracteres)
“a”, “z”, “k”, “João Oliveira”, “8765”, “!@#\$%&”
 - . Lógicos (booleanos)
True e False
- . Python é capaz de identificar o tipo da variável através do contexto

Entrada de dados

- O Python utiliza da palavra reservada `input` para ler os dados de entrada
- É possível escrever um texto no qual será exibido ao usuário informando-o que é necessário digitar um valor

`input("texto")`

- O texto é opcional mas o uso de parênteses é obrigatório

Saída de dados

- O Python utiliza da palavra reservada `print` para imprimir os dados de saída

```
print()
```

- O uso de parênteses é obrigatório
- Se não haver nenhuma variável ou valor entre os parênteses, uma linha em branco é imprimida
- Caso haja um valor numérico, textual ou lógico, o comando imprime o valor informado
- Caso haja uma variável, o valor armazenado na variável é imprimido

Conversão de tipos

- O comando input sempre lê as entradas como uma cadeia de caracteres, e por vezes, os valores entrados são numéricos
- Para mudar o tipo de uma variável usamos a palavra reservada do tipo que se deseja converter e a variável entre os parênteses

`var = "1234" #suponha esse valor de entrada`

`int(var) #para valores inteiros`

`float(var) #para valores reais`

`str(var) #para retornar ao valor textual`

Ilustrando esses conceitos no exemplo

ENTRADA

```
nome = input("Informe o seu nome: ")
```

ENTRADA

```
idade = int(input("Informe a sua idade: "))
```

CONVERSÃO PARA INTEIRO

ENTRADA

```
altura = float(input("Informe a sua altura: "))
```

CONVERSÃO PARA REAL

ENTRADA

```
peso = float(input("Informe o seu peso: "))
```

CONVERSÃO PARA REAL

```
imc = peso / (altura * altura)
```

```
msg = nome + ", " + str(idade) + " anos, o seu índice de massa corporal é: %.2f" % imc + " kg/m²"
```

CONVERSÃO PARA STRING

```
print(msg)
```

SAÍDA

Erros comuns em programas

- Erros de sintaxe
 - São erros que ocorrem devido ao uso errado das estruturas da linguagem
 - O Python detecta e informa tais erros ao programador
- Erros de lógica
 - São erros onde o programa executa normalmente mas não fornece o resultado esperado

Ilustrando esses conceitos no exemplo

Erros de sintaxe

```
nome ○ input("Informe o seu nome: ") # falta o símbolo de atribuição =
```

```
idade = int(input("Informe a sua idade: ")○) # o parênteses não foi fechado
```

```
altura = floa○(input("Informe a sua altura: ")) # a palavra float está com um erro de digitação
```

```
peso = float(input(○Informe o seu peso:○ )) # falta aspas no texto
```

```
imc = peso / ○ # divisão por zero
```


Ilustrando esses conceitos no exemplo

Erro de lógica

```
msg = str(idade) + ", %.2f" % imc + " anos, o seu índice de massa corporal é: " + nome + " kg/m²"  
print(msg)
```

- Considerando as entradas como Alex, 22, 1.74, 70 temos a seguinte mensagem
22, 23.12 anos, o seu índice de massa corporal é: Alex kg/m²
- O trecho de código acima está logicamente errado pois não é o resultado esperado

Agora é sua vez

1. Escreva um algoritmo que leia o valor e quantidade de um produto e imprima o valor total a ser pago pelo produto
2. Escreva um algoritmo que leia o raio de uma círculo, calcule e imprima a área desse círculo.
3. Escreva um algoritmo que leia três notas de um aluno e imprima a média aritmética dele.
4. Uma empresa decidiu dar um aumento de 10% aos seus funcionários. Escreva um algoritmo que leia um salário e imprima o novo valor deste salário após o aumento dado.