Capitulo 1

O capitulo 1 busca desmistificar a programação e motivar o leitor, utilizando de analogias e situações cotidianas para compreendermos a lógica de programação, como acertar um relógio ou se guiar por um mapa, entender a programar é saber como seguir instruções passo a passo em uma ordem específica, e entender a linguagem de programação é como aprender um novo idioma.

Neste capitulo também é introduzido a linguagem a ser trabalhada a linguagem Python que segundo o livro é uma linguagem simples e poderosa, é também um software livre, compatível com uma enorme variedade de arquiteturas de computadores disponíveis no mercado. O uso do python é diverso podendo ser utilizado para animações 3D quanto para automação e machine learning. Outra vantagem do python é a facilidade de compreender os códigos por não ter uma poluição visual muito grande de marcadores e sua extensa quantidade de bibliotecas disponíveis permitindo que o desenvolvedor tenha o foco exatamente no que precisa

(Listagem 1.1)

Nesse primeiro programa utilizamos apenas a função \*\*\_print\_\*\* que é utilizada para imprimir os dados na tela. Tudo que for escrito entre aspas após o codigo \*\*\_print\_\*\* será impresso na tela, as aspas tem função apenas de limitar o início e o fim dos dados impressos, pois sem elas o computador não conseguirá interpretar o comando e irá gerar um erro.

Capitulo 2

Neste capitulo conhecemos o interpretador Python que é responsável por “traduzir” o nosso codigo para que o computador possa compreende-lo, o interpretador indica erros (se houverem) e executa os comandos linha a linha

O livro recomenda a utilização do Python 3.4 pois sofreu diversas alterações entre a versão 2 e 3, nessa etapa o livro nos ensina a baixar e instalar o Python nos sistemas operacionais mais populares (Windows, Linux e Mac), neste trabalho utilizo a versão 3.10 e o interpretador disponível na biblioteca do aplicativo \_Vs Code\_.

Para prosseguirmos precisamos de um editor de textos especial que suporte a edição em UTF-8 (é um tipo de codificação binaria ou Unicode com suporte a ASCII) o editor utilizado neste trabalho é o já mencionado \_Vs Code\_, estes editores tornam a compreensão facilitada destacando por cores o codigo indicando: funcoes, variáveis, linhas de erro (se houverem) e permitindo a utilização de acentos, ao salvar adicionamos a extinção que e caracteriza pelo final \*\_.py\_\* todos os arquivos Python devem ser salvos neste formato como: \*\_Exemplo.py\_\*. Ao terminar seu codigo podemos executá-lo ou testar apertando F5 (no editor idle) ou selecionando a opção \_run\_ em outros editores Python (lembre-se de salvar antes de executar o programa).

No capítulo 2.4 temos uma listagem de cuidados ao digitamos um programa que resumidamente são:

\* Letras maiúsculas e minúsculas são interpretadas de formas diferentes o que pode gerar erro se utilizamos ou não a capitalização (Ou maiusculização) em alguma função que deveria ser escrita inteiramente com letras minúsculas

\* Aspas “” são essenciais, ao abri-las lembre-se sempre de fecha-las

\* Parênteses são obrigatórios nunca os remova! Assim como as aspas sempre se lembre de fechá-los

\* Atenção com espaços e tabulação, os espaços são tão importantes quanto os caracteres digitados um único espaço (ou a falta dele) pode ser a diferença entre o programa funcionar ou não

Os parênteses em operações com Python funcionam da mesma forma que em uma operação convencional, o Python também respeita a prioridade das operações (\*\*PEMDAS\*\*).

Conceito de variáveis e atribuições

Variáveis são uma forma de armazenar um dado na memória do computador normalmente se utiliza variáveis paraqresover equações com uma complexidade elevada, para declarar uma variável se utiliza o símbolo da igualdade (=) seguido pelo valor que será atribuído a variável

>essa ação pode ser chamada de atribuição

Exemplo

A = 6

B = 4

>A recebe 6

>B recebe 4

Toda variável precisa ter um nome

Em seguida podemos “chamar” a variável para realizar operações

Listagem 2.8

Listagem 2.11

Primeiro declaramos as variáveis e atribuímos um valor para elas, em seguida escrevemos a nossa função onde o aumento multiplica o salário e o resultado é dividido por 100 para somar ao valor inicial do salário. Variáveis são uteis para facilitar o desenvolvimento e para aumentar a clareza do que é feito no programa sendo mais fácil de escrever e ler o que está sendo feito.

Exercicios 2.3 a 2.6

CAPITULO 3

Variáveis e entrada de dados

Agora vamos aprofundar o conceito de variáveis e entrada de dados, começando com as nomenclaturas.

(nessa seção o livro nos fornece uma tabela que irei reproduzir)

De forma breve variáveis não podem ter espaços em branco e não podem começar com números,

>versões anteriores ao Python 3 não suportam acentuação.

Variáveis numéricas

Neste ponto teremos dois tipos principais de variáveis variáveis inteiras e variáveis de ponto flutuante, a diferença e notável visto que com inteiros não teremos nenhuma casa decimal após a virgula, já em uma variável de ponto flutuante(float) sempre teremos a parte decimal após a virgula mesmo que o seu valor seja 0

Exercício 3.1

Representações numéricas