Think Python

How to think like a computer scientist

Allen B. Downey

6 de agosto de 2021

Conteúdo

1

The Way of the Program

> *"The single most important skill for a computer scientistis **problem solving**. That is the ability to formulate problems, think creatively about solutions, and express a solution clearly and accurately"*
>
> – página 1

1.1 O que é um Programa?

Um **programa** é uma sequencia de instruções que especifica como fazer um determinado computation. Um **computation** [1] é qualquer tipo de cálculo que inclua passos artiméticos (computação matemática) e não artiméticos (computação simbólica) e que segue uma determinada ordem.
Algumas características comuns em qualquer linguagem de programação:

- input → dados de teclado, arquivo, rede, etc.
- output → mostrar dados na tela, salvar em um arquivo, enviar na rede, etc.
- math → operações matemáticas
- conditional execution → só executar um pedaço do código após certas condições
- repetition → reexecutar um pedaço do código com alguma variação

1.2 Executando o Python

O **interpretador** do Python é um programa que lê e executa códigos escritos em Python. Tem duas maneiras de se executar códigos em Python a

[1] https://en.wikipedia.org/wiki/Computation

primeira é no modo **interativo** onde podemos enviar as ordens direto pro interpretador e a segunda em em modo **script** onde o interpretador analisa o código inteiro antes de começar o processamento.

1.3 Operadores Aritméticos

```
# Soma
40 + 1

# Subtracao
10 - 1

# Multiplicacao
20 * 10

# Divisao
11 / 1

# Exponenciacao
6 ** 2
```

Listing 1.1: Operadores Básicos

1.4 Linguagens Formais e Naturais

As **linguagens naturais** são as linguagens que as pessoas usam para se comunicar no dia a dia. São criadas e se modificam organicamente com o passar do tempo.
Por outro lado, as **linguagens formais** são criadas por pessoas e possuem uma aplicação específica. As linguagens de computação são linguagens formais desenvolvidas para expressar computações.
As linguagens formais possuem regras estritas que chamamos de **regras de sintaxe**.

1.4.1 Regras de Sintaxe

As regras de sintaxe possuem dois sentidos gerais:

- tokens → são os elementos da linguagem
- estrutura → é a regra de combinação dos tokens

Quando o interpretador lê um código, ele faz um processo chamado **parsing** que é justamente entender a estrutura de combinação entre os diferentes tokens presentes no script. No caso do Python a identação faz um papel importante na estrutura.

1.5 Debugging

Os erros em um programa são chamados de **bugs** e o processo de resolução desses erros é chamado de **debugging**.

1.6 Glossário

- problem solving → processo de formular um problema, achar uma solução e expressá-la
- high-level language → lingaugem desenhada para ser fácil para um humano ler e escrever
- low-level language → lingaugem desenhada para ser fácil para um computador ler. AKA "machine language"ou "assembly language"
- portability → capacidade de um programa rodar em mais de um tipo de computador
- interpreter → um programa que lê outro programa e o executa
- prompt → caracteres no prompt que indicam que ele está pronto para receber input direto do usuário
- program → um conjunto de instruções que especifica uma computation
- print statement → uma instrução que diz para o interpretador mostrar caracteres na tela
- operator → um caracter especial que representa uma computação simples (+, -, *, /, **)
- value → uma das unidades básica de dados
- type → a categoria em que um valor pertence
- interger → numero inteiros
- floating-point → números fracionados
- string → sequencia de caracteres
- natural language → linguagem humana
- formal language → linguagem com objetivo e regras específicos
- token → um dos elementos básicos de uma linguagem formal, análogo à palavra na linguagem natural
- syntax → as regras de estrutura de um programa

- parse → processo de examinar um programa e analisar sua estrutura
- bug → um erro no programa
- debugg → o processo de resolver um erro

2

Varibles, Expressions and Statements

> *"One of the most powerful features of a programming language is the ability to manipulate* ***variables****. A variable is a name that refers to a value"*
>
> – página 11

2.1 Statements de atribuição

```
1 message = 'acesse www.economiamainstream.com.br'
2
3 n = 12
4
5 pi = 3.1415
```

Listing 2.1: Atribuições para diferentes classes

3

Functions

4

Case Study: Interface Design

5

conditionals and Recursion

6

Fruitful Funtions

7

Iteration

8

Strings

9

Case Study: Word Play

10

Lists

11

Dictionaries

12

Tuples

13

Case Study: Data Structure Selection

14

Files

15

Classes and Objects

16

Classes and Functions

17

Classes and Methods

18

Inheritance

19

The Goodies

20

Debugging

21

Analysis of Algorithms