



Universidade de Brasília

Departamento de Ciência da Computação



Semana 02

Lógica de Programação

CIC0004

Algoritmos e Programação de Computadores

Prof. Pedro Garcia Freitas

<https://pedrogarcia.gitlab.io/>

pedro.garcia@unb.br

Brasília



Este conjunto de slides não deve ser utilizado ou republicado sem a expressa permissão do autor.

This set of slides should not be used or republished without the author's express permission.



1. Breve Introdução

1.0. Quem sou eu

- +15 anos trabalhando com programação:
 - Desenvolvimento de sistemas de informação;
 - Relatórios e controle financeiro;
 - Gestão de documentos;
 - *Smartgrids*;
 - *Codecs*;
 - Ciência de dados;
 - Pesquisa e desenvolvimento de produtos móveis;
 - IA/ML;
 - Câmera;
 - *Smartwatches*;



1. Breve Introdução

1.0. Quem sou eu

- Diversas linguagens
 - Python
 - Fortran
 - C
 - C++
 - Javascript
 - PHP
 - Java/Scala/Groovy
 - Etc




1. Breve Introdução


1.0. Quem sou eu



- Diversos escopos
 - Academia (TCC/Mestrado/Doutorado/UnB/TuDelft)
 - Governo (ABEn, MCTI, MPOG)
 - Software livre (FreBSD project)
 - Indústria (Samsung)
 - Comitês internacionais de padronização (JPEG/MPEG)







1. Breve Introdução





 <https://gitlab.com/wg1/jpeg-pleno-refsw>

 Search GitLab



 JPEG >  JPEG Pleno Reference Software



 **JPEG Pleno Reference Software** 
Project ID: 16452603 






 Star 3




 1,529 Commits  1 Branch  3 Tags  3.6 MB Project Storage



JPEG Pleno Light Field reference software model (4DTM tools)

 Merge branch 'pedrogarcia-master-patch-30886' into 'master' 
Pedro Garcia Freitas authored 1 year ago

 251eae8e 

master  jpeg-pleno-refsw  Find file   Clone 

 README  BSD 3-Clause "New" or "Revised" License  CONTRIBUTING

Name	Last commit	Last update
 cfg/part2	Update cfg/part2/4DTransformMode/Bikes/IO1_Bikes_22...	1 year ago
 cmake	Added support to hashdigs on EncoderRuntimeStatis...	2 years ago



1. Breve Introdução

1.1. Objetivos

- Desenvolver um “pensamento computacional”:



1. Breve Introdução

1.1. Objetivos

- Desenvolver um “pensamento computacional”:
 - Compreender a organização básica de um computador.



1. Breve Introdução

1.1. Objetivos

- Desenvolver um “pensamento computacional”:
 - Compreender a organização básica de um computador.
 - Adquirir competência para representar a resolução de problemas por meio de algoritmos.



1. Breve Introdução

1.1. Objetivos

- Desenvolver um “pensamento computacional”:
 - Compreender a organização básica de um computador.
 - Adquirir competência para representar a resolução de problemas por meio de algoritmos.
 - Adquirir conhecimentos básicos em Linguagem de programação Python



1. Breve Introdução

1.1. Objetivos

- Desenvolver um “pensamento computacional”:
 - Compreender a organização básica de um computador.
 - Adquirir competência para representar a resolução de problemas por meio de algoritmos.
 - Adquirir de conhecimentos básicos em Linguagem de programação Python: não será um curso Python.



1. Breve Introdução

1.1. Objetivos

- Destinado a alunos que têm **pouca ou nenhuma experiência pregressa** com programação.



1. Breve Introdução

1.1. Objetivos

- Destinado a alunos que têm **pouca ou nenhuma experiência pregressa com programação**.
 - Além disso, o aluno deverá ser capaz de escrever e ler códigos escritos em linguagem Python (ou aprender qualquer outra linguagem autodidaticamente).



1. Breve Introdução

1.1. Objetivos

- Destinado a alunos que têm **pouca ou nenhuma experiência pregressa** com programação.
 - Além disso, o aluno deverá ser capaz de escrever e ler códigos escritos em linguagem Python (ou aprender qualquer outra linguagem autodidaticamente).
- Ao final do curso o aluno deverá ter adquirido confiança em sua habilidade de propor e interpretar algoritmos.



1. Breve Introdução

1.1. Objetivos

- Destinado a alunos que têm **pouca ou nenhuma experiência pregressa** com programação.
 - Além disso, o aluno deverá ser capaz de escrever e ler códigos escritos em linguagem Python (ou aprender qualquer outra linguagem autodidaticamente).
- Ao final do curso o aluno deverá ter adquirido confiança em sua habilidade de propor e interpretar algoritmos.
- Programação requer prática!



1. Breve Introdução

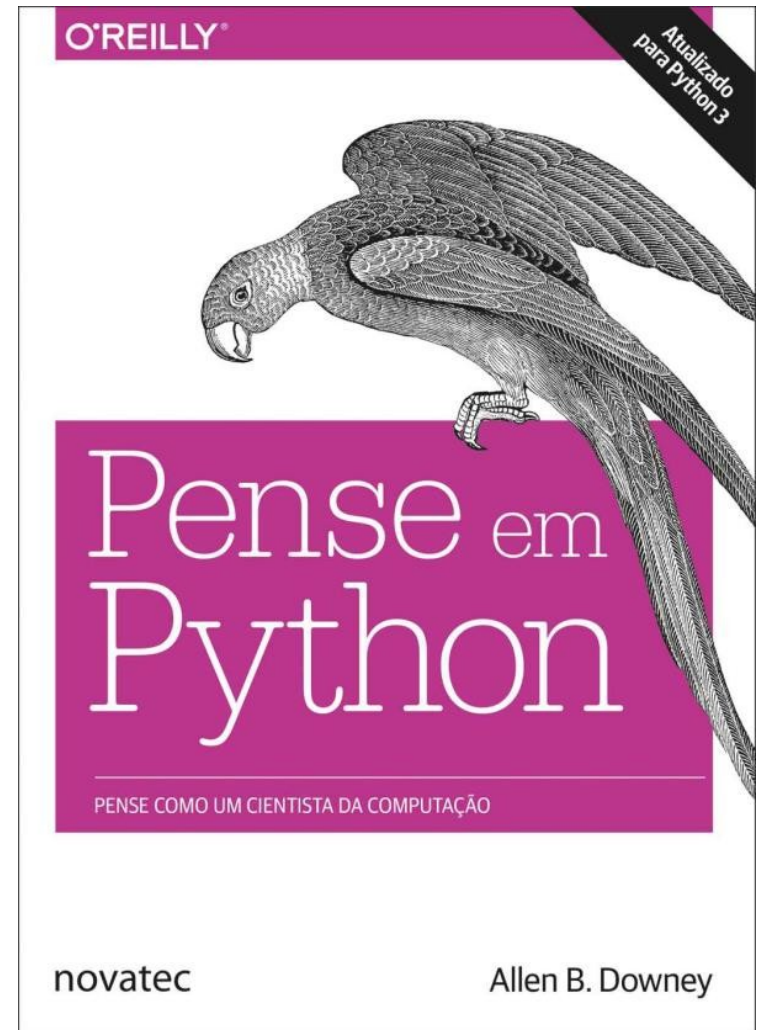
1.1. Objetivos

- Destinado a alunos que têm **pouca ou nenhuma experiência pregressa** com programação.
 - Além disso, o aluno deverá ser capaz de escrever e ler códigos escritos em linguagem Python (ou aprender qualquer outra linguagem autodidaticamente).
- Ao final do curso o aluno deverá ter adquirido confiança em sua habilidade de propor e interpretar algoritmos.
- Programação requer prática!
 - Ou seja, a **presença** do aluno nos laboratórios é essencial.



1. Breve Introdução

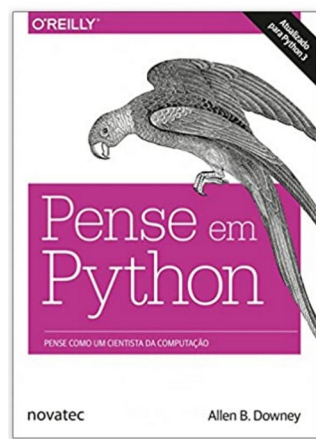
1.2. Bibliografia adotada





1. Breve Introdução

1.2. Bibliografia adotada



Ver todas as 2 imagens

Ler amostra

Pense em Python: Pense Como um Cientista da Computação



Capa comum – 7 junho 2016

Edição Português | por Allen B. Downey (Autor)

★★★★★ 401 avaliações de clientes

Ver todos os formatos e edições

Kindle
R\$ 70,11

Leia com nossos apps gratuitos

Capa Comum
R\$ 75,70

Outros novo a partir de
R\$ 75,70

Se você quer aprender como programar, usar Python é uma ótima forma de começar. Este guia prático apresenta a linguagem passo a passo, começando com conceitos de programação básicos antes de chegar a funções, recursividade, estruturas de dados e design orientado a objeto. Esta edição e seu código de apoio foram atualizados para o Python 3.

Com os exercícios em cada capítulo, você testará conceitos de programação conforme os aprende. Pense em Python é ideal para estudantes de ensino médio e universitários ou para autodidatas, estudantes educados em casa e profissionais que precisam aprender fundamentos de programação. Os principiantes que quiserem começar por uma seção básica, podem começar com Python

▼ Leia mais

ISBN-10



ISBN-13



Edição



Editora



Data da publicação



Idioma



Comprar
novo:

R\$ 75,70

De: R\$ 95,00

Você economiza: R\$ 19,30 (20%)

Entrega GRÁTIS: Quinta-feira, 6 de Abril em produtos enviados pela Amazon a partir de R\$129 em produtos elegíveis

Ou Entrega R\$ 8,90: Quinta-feira, 6 de Abril

Ou entrega mais rápida: Terça-feira, 4 de Abril. Se pedir dentro de 1 h 53 mins

Enviar para Pedro - Campinas 13098401

Em estoque

Quantidade: 1

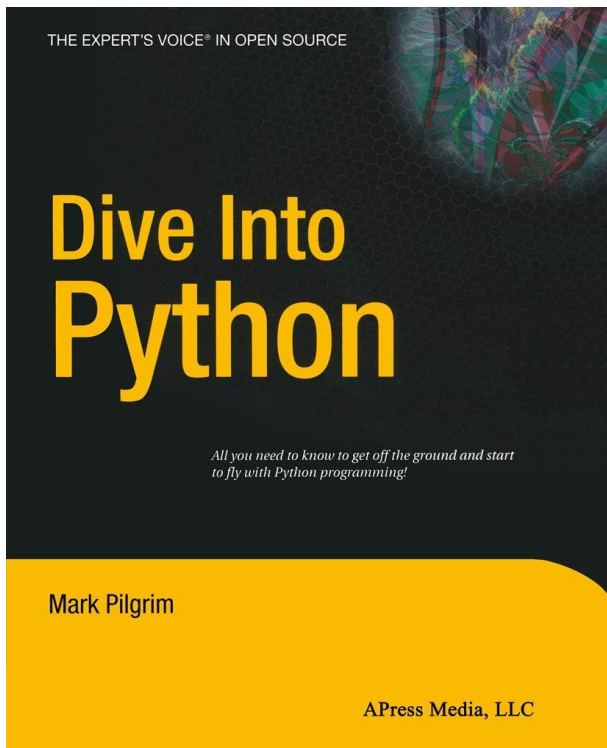
Adicionar ao carrinho

Comprar agora



1. Breve Introdução

1.3. Bibliografia adicional sugerida

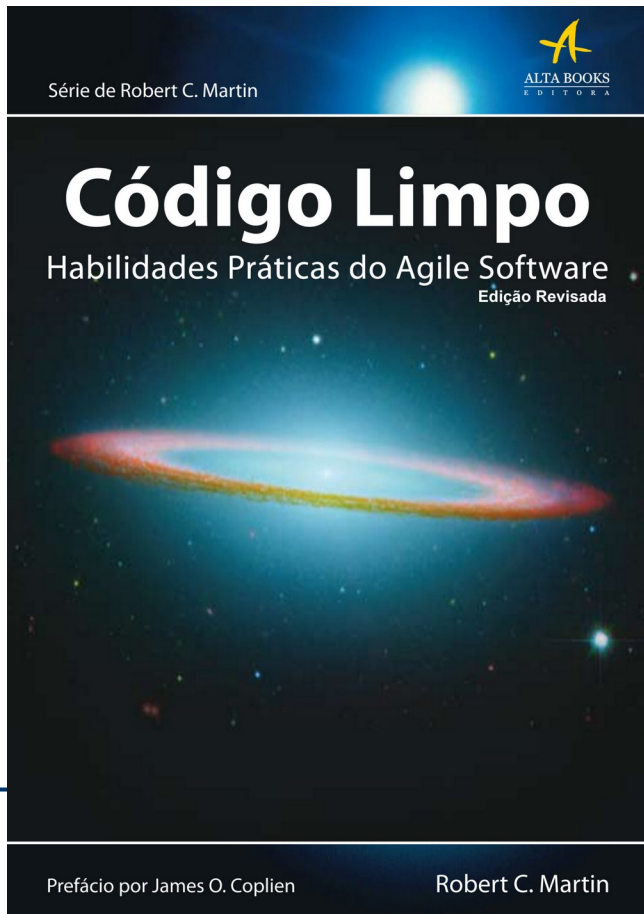


- Pilgrim, Mark, and Simon Willison. **“Dive into Python 3”**. Vol. 2. New York, NY, USA: Apress, 2009.
- Impresso: R\$ 231,37 na Amazon.
- Digital: Totalmente grátis em
 - <https://diveintopython3.net/table-of-contents.html>



1. Breve Introdução

1.3. Bibliografia adicional sugerida



- Martin, Robert C. “Código limpo: habilidades práticas do Agile software”. Alta Books, 2019.
- Impresso: R\$ 76,99 na Amazon.



2. Introdução à Lógica de Programação

2.1. Lógica



2. Introdução à Lógica de Programação

2.1. Lógica

- A lógica de programação é necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas, ela permite definir a sequência lógica para o desenvolvimento.



2. Introdução à Lógica de Programação

2.1. Lógica

- A lógica de programação é necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas, ela permite definir a sequência lógica para o desenvolvimento.
- Mas o que é lógica de programação?



2. Introdução à Lógica de Programação

2.1. Lógica

- A lógica de programação é necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas, ela permite definir a sequência lógica para o desenvolvimento.
- Mas o que é lógica de programação?
 - **Lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo.**



2. Introdução à Lógica de Programação

2.2. Sequência Lógica

Estes pensamentos, podem ser descritos como uma **sequência de instruções**, que devem ser seguidas para se cumprir uma determinada tarefa.



2. Introdução à Lógica de Programação

2.2. Sequência Lógica

Estes pensamentos, podem ser descritos como uma sequência **de instruções**, que devem ser seguidas para se cumprir uma determinada tarefa.

Sequência Lógica são passos executados até atingir um objetivo ou solução de um problema.



2. Introdução à Lógica de Programação

2.2. Sequência Lógica

Sempre que decidimos fazer qualquer atividade em nosso dia a dia, acabamos seguindo uma **sequência lógica**. Na maior parte do tempo, fazemos isso de maneira tão natural que nem nos damos conta, mas, quando percebemos, conseguimos enxergar passos que levaram ao resultado final.



2. Introdução à Lógica de Programação

2.2. Sequência Lógica

Exemplo: cozinhar um bolo



2. Introdução à Lógica de Programação

2.2. Sequência Lógica

Exemplo: cozinhar um bolo



1. Selecionar os ingredientes da receita;
2. Selecionar tigela;
3. Colocar farinha, de acordo com a medida;
4. Selecionar ovos;
5. Colocar manteiga e açúcar a gosto;
6. Colocar leite;
7. Misturar todos os ingredientes na tigela;
8. Despejar a massa na forma;
9. Levar ao forno;
10. Aguardar 40 minutos;
11. Retirar do forno;

2. Introdução à Lógica de Programação

2.2. Sequência Lógica

Exemplo: cozinhar um bolo



- Note que cada pessoa pode definir a sequência de passos de maneira diferente, sendo que algumas **instruções** dependem da ordem e outras não.

2. Introdução à Lógica de Programação

2.2. Sequência Lógica

Exemplo: cozinhar um bolo



- Note que cada pessoa pode definir a sequência de passos de maneira diferente, sendo que algumas **instruções** dependem da ordem e outras não.
 - Trocar a ordem da inserção dos ingredientes antes da mistura pode não afetar o resultado final.

2. Introdução à Lógica de Programação

2.2. Sequência Lógica

Exemplo: cozinhar um bolo



- Note que cada pessoa pode definir a sequência de passos de maneira diferente, sendo que algumas **instruções** dependem da ordem e outras não.
 - Trocar a ordem da inserção dos ingredientes antes da mistura pode não afetar o resultado final.
 - Mas levar os ingredientes ao forno antes de realizar a mistura não irá trazer o resultado esperado.



2. Introdução à Lógica de Programação

2.2. Sequência Lógica

- Em atividades rotineiras, não costumamos prestar atenção quando seguimos uma mesma ordem para executar tarefas.



2. Introdução à Lógica de Programação

2.2. Sequência Lógica

- Em atividades rotineiras, não costumamos prestar atenção quando seguimos uma mesma ordem para executar tarefas.
- Porém, quando o assunto é **programar**, definir as **etapas** do que deve ser feito assume uma grande importância, uma vez que **instruir um computador** ainda é bem diferente do que instruir uma pessoa.



2. Introdução à Lógica de Programação

2.3. Instruções

- Na linguagem comum, são instruções “um conjunto de regras ou normas definidas para a realização ou emprego de algo”.



2. Introdução à Lógica de Programação

2.3. Instruções

- Na linguagem comum, são instruções **“um conjunto de regras ou normas definidas para a realização ou emprego de algo”**.
- Em programação, entretanto, **instrução** é a informação que indica a um computador **uma ação elementar a executar**.



2. Introdução à Lógica de Programação

2.3. Instruções

- Na linguagem comum, são instruções **“um conjunto de regras ou normas definidas para a realização ou emprego de algo”**.
- Em programação, entretanto, **instrução** é a informação que indica a um computador **uma ação elementar a executar**.
- Convém ressaltar que uma ordem isolada não permite realizar o processo completo, para isso é necessário um **conjunto de instruções** colocadas em **ordem sequencial lógica**.



2. Introdução à Lógica de Programação

2.3. Instruções

- Por exemplo, se quisermos fazer uma omelete, precisaremos colocar em prática uma **série de instruções**:



2. Introdução à Lógica de Programação

2.3. Instruções

- Por exemplo, se quisermos fazer uma omelete, precisaremos colocar em prática uma **série de instruções**:
 - Quebrar os ovos, bater os ovos, fritar os ovos, etc...



2. Introdução à Lógica de Programação

2.3. Instruções

- Por exemplo, se quisermos fazer uma omelete, precisaremos colocar em prática uma **série de instruções**:
 - Quebrar os ovos, bater os ovos, fritar os ovos, etc...
- Essas instruções tem que ser executadas em uma **ordem adequada** – não se quebra os ovos depois de fritá-los.



2. Introdução à Lógica de Programação

2.3. Instruções

- Por exemplo, se quisermos fazer uma omelete, precisaremos colocar em prática uma **série de instruções**:
 - Quebrar os ovos, bater os ovos, fritar os ovos, etc...
 - Essas instruções tem que ser executadas em uma **ordem adequada** – não se quebra os ovos depois de fritá-los.
 - Dessa maneira, uma instrução tomada em separado não tem muito sentido; para obtermos o resultado, precisamos colocar em prática o **conjunto de todas as instruções, na ordem correta**.
-



2. Introdução à Lógica de Programação

2.3. Instruções

Instruções são um conjunto de regras ou normas definidas para a realização ou emprego de algo. Em **informática**, é o que **indica a um computador uma ação elementar a executar**.

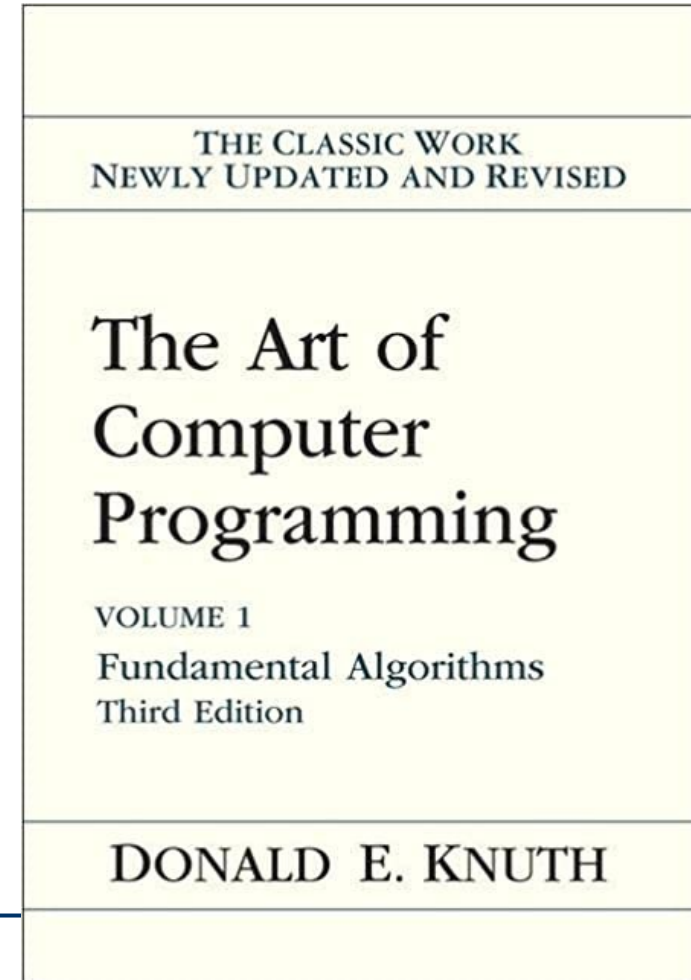


2. Introdução à Lógica de Programação

2.3. Instruções

Curiosidade: aprender a programar nos anos 60

- Naquela época, as sequências lógicas eram ensinadas com instruções muito básicas, próximo do que acontece no processador ainda hoje em dia;



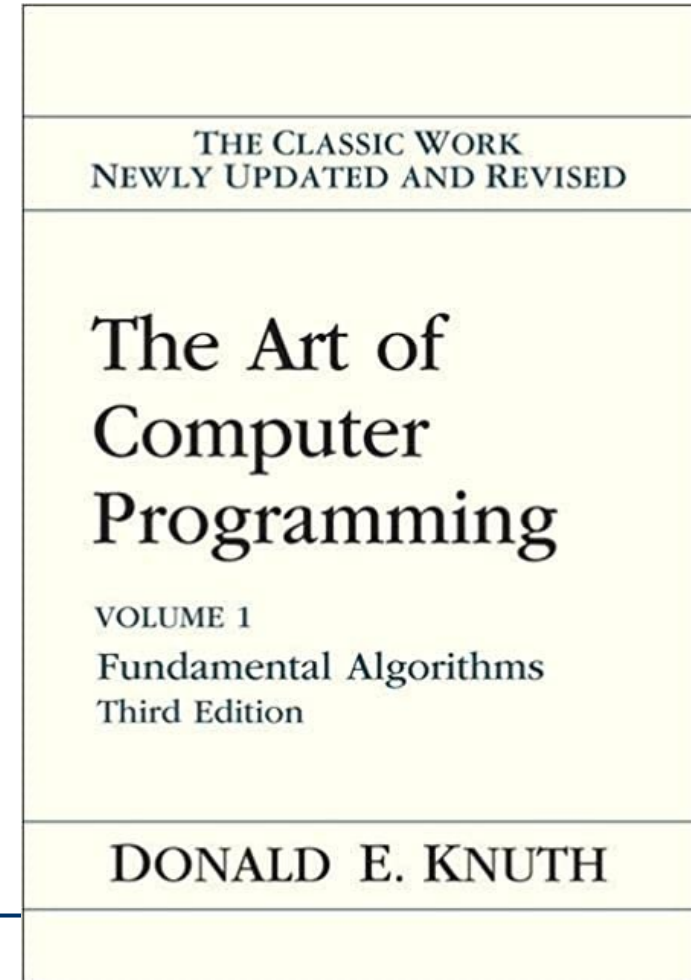


2. Introdução à Lógica de Programação

2.3. Instruções

Curiosidade: aprender a programar nos anos 60

- Naquela época, as sequências lógicas eram ensinadas com instruções muito básicas, próximo do que acontece no processador ainda hoje em dia;
- De fato, esse conceito de instrução vai aparecer em diversas matérias durante o curso de vocês;

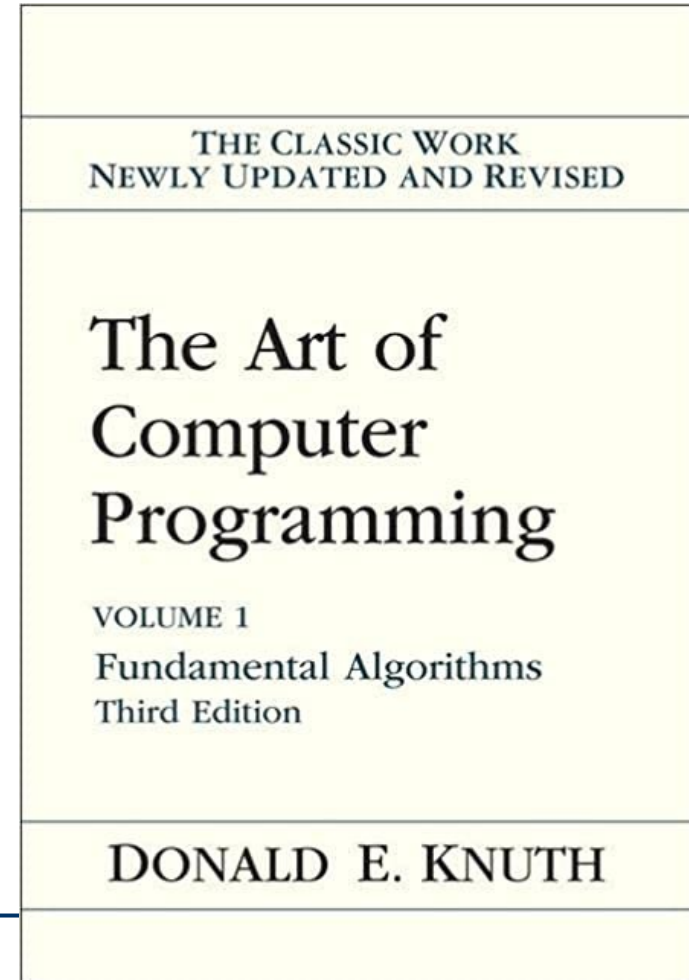




2. Introdução à Lógica de Programação

2.3. Instruções

- Quando programamos (definimos a sequência lógica de programação) utilizando diretamente as instruções que o processador “entende”, chamamos isso de *“programação de baixo nível”*.

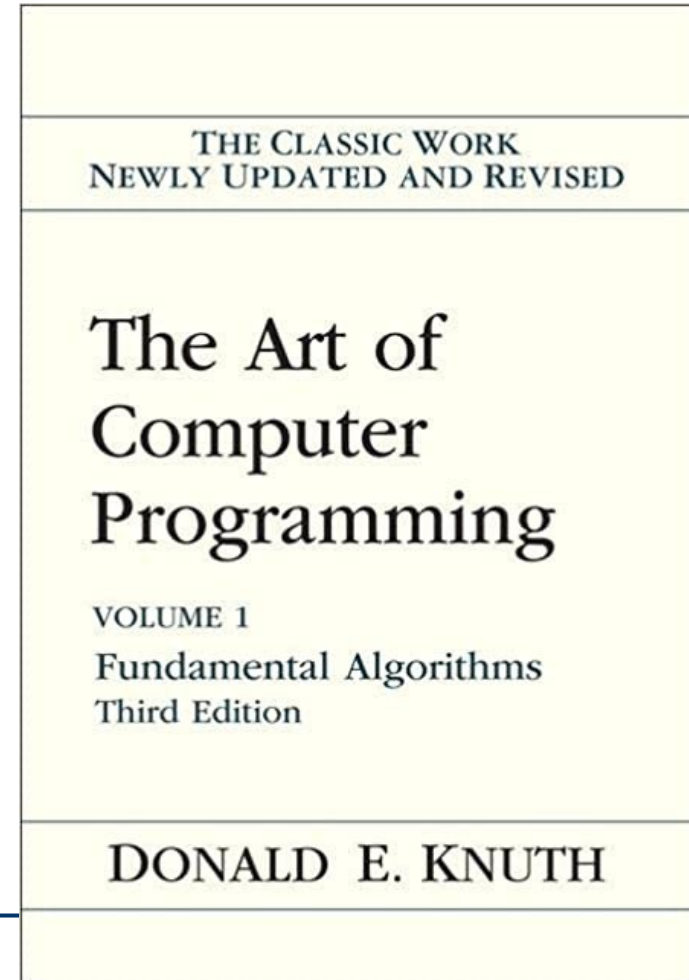




2. Introdução à Lógica de Programação

2.3. Instruções

- Quando programamos (definimos a sequência lógica de programação) utilizando diretamente as instruções que o processador “entende”, chamamos isso de *“programação de baixo nível”*.
- Quando utilizamos uma linguagem mais próxima da natural, chamamos de *“programação de alto nível”*.





2. Introdução à Lógica de Programação

2.3. Instruções

- Quando programamos (definimos a sequência lógica de programação) utilizando diretamente as instruções que o processador “entende”, chamamos isso de *“programação de baixo nível”*.
- Quando utilizamos uma linguagem mais próxima da natural, chamamos de *“programação de alto nível”*.
 - Neste curso, começaremos a estudar a programação de alto nível. Contudo, nos semestres vindouros, vocês irão estudar como funciona o computador em níveis mais baixos (i.e., próximos do hardware).

THE CLASSIC WORK
NEWLY UPDATED AND REVISED

The Art of Computer Programming

VOLUME 1

Fundamental Algorithms
Third Edition

DONALD E. KNUTH



2. Introdução à Lógica de Programação

2.3. Instruções

- Independente do nível, quando a necessidade é desenvolver um programa, precisamos deixar bem claro a sequência que deve ser seguida para atingir o resultado esperado.
- A esse encadeamento lógico na programação, chamamos de Lógica de Programação, e a **descrição de como fazer**, definimos como **Algoritmos**.



2. Introdução à Lógica de Programação

2.4. Algoritmos

- Um algoritmo é formalmente uma **sequência finita** de passos que levam a execução de uma tarefa com a **intenção de atingir um objetivo**.



2. Introdução à Lógica de Programação

2.4. Algoritmos

- Um algoritmo é formalmente uma **sequência finita** de passos que levam a execução de uma tarefa com a **intenção de atingir um objetivo**.
 - Pode ser desde atravessar uma rua, fazer um bolo ou definir qual critério usar para aprovar ou reprovar um aluno, por exemplo.



2. Introdução à Lógica de Programação

2.4. Algoritmos

- Um algoritmo é formalmente uma seqüência finita de passos que levam a execução de uma tarefa com a **intenção de atingir um objetivo**.
 - Pode ser desde atravessar uma rua, fazer um bolo ou definir qual critério usar para aprovar ou reprovar um aluno, por exemplo.
- Estas tarefas não podem ser redundantes nem subjetivas na sua definição, devem ser claras e precisas.



2. Introdução à Lógica de Programação

2.4. Algoritmos

- Um algoritmo é formalmente uma seqüência finita de passos que levam a execução de uma tarefa com a **intenção de atingir um objetivo**.
 - Pode ser desde atravessar uma rua, fazer um bolo ou definir qual critério usar para aprovar ou reprovar um aluno, por exemplo.
- Estas tarefas não podem ser redundantes nem subjetivas na sua definição, devem ser claras e precisas.
- Podemos pensar em algoritmo como uma receita.



2. Introdução à Lógica de Programação

2.4. Algoritmos

Como exemplos de algoritmos podemos citar os algoritmos das operações básicas (adição, multiplicação, divisão e subtração) de números reais decimais. Outros exemplos seriam os manuais de aparelhos eletroeletrônicos, como uma lavadora de roupas “*smart*”, que explicam passo a passo como, por exemplo, gravar um evento.



2. Introdução à Lógica de Programação

2.4. Algoritmos

- Até mesmo as coisas mais simples, podem ser descritas por sequências lógicas. Por exemplo:

"Chupar uma bala"

- Pegar a bala
- Retirar o papel
- Chupar a bala
- Jogar o papel no lixo



2. Introdução à Lógica de Programação

2.4. Algoritmos

“Somar dois números quaisquer”

- Escreva o primeiro número no retângulo A
- Escreva o segundo número no retângulo B
- Some o número do retângulo A com número do retângulo B e coloque o resultado no retângulo C





2. Introdução à Lógica de Programação

2.4. Algoritmos

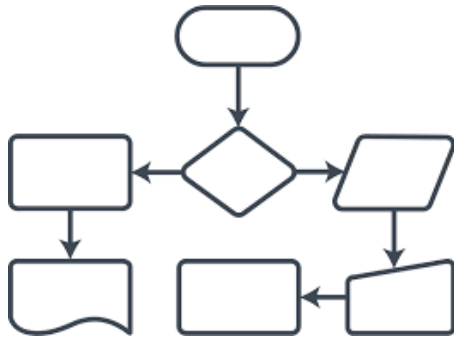
Como representar um algoritmo?



2. Introdução à Lógica de Programação

2.4. Algoritmos

Como representar um algoritmo?



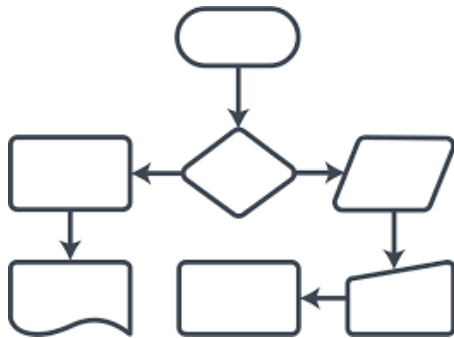
Fluxograma



2. Introdução à Lógica de Programação

2.4. Algoritmos

Como representar um algoritmo?



Fluxograma

```
Algoritmo CalculaMedia
  LER Nota1
  LER Nota2
  LER Nota3
  media = (Nota1+Nota2+Nota3)/3
  SE MEDIA >= 6 ENTÃO
    IMPRIME "Aprovado"
  SENÃO
    IMPRIME "Reprovado"
  FIM-SE
  IMPRIME o valor da média
Fim-Algoritmo
```

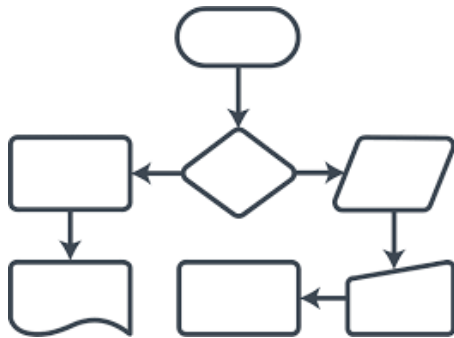
Pseudocódigo



2. Introdução à Lógica de Programação

2.4. Algoritmos

Como representar um algoritmo?



Fluxograma

```
Algoritmo CalculaMedia
  LER Nota1
  LER Nota2
  LER Nota3
  media = (Nota1+Nota2+Nota3)/3
  SE MEDIA >= 6 ENTÃO
    IMPRIME "Aprovado"
  SENÃO
    IMPRIME "Reprovado"
  FIM-SE
  IMPRIME "Media = " media
Fim-Algoritmo
```

Pseudocódigo

```
def compute_mean(s):
    return sum(s) / len(s)

n1 = int(input())
n2 = int(input())
n3 = int(input())

compute_mean([n1, n2, n3])
```

Linguagem de Programação



2. Introdução à Lógica de Programação

2.5. Programas

Os programas de computadores nada mais são do que **algoritmos** escritos numa **linguagem de computador** (Pascal, C, Cobol, Fortran, Visual Basic entre outras) e que são **interpretados e/ou executados** por uma máquina, no caso um computador.



Exercícios

- E1. Crie uma sequência lógica para tomar banho.**
- E2. Faça um algoritmo para somar dois números e multiplicar o resultado pelo primeiro número.**
- E3. Descreva com detalhes a sequência lógica para Trocar um pneu de um carro.**
- E4. Faça um algoritmo para trocar uma lâmpada.**



Exercícios

E1. Crie uma sequência lógica para tomar banho.

1. Tirar a roupa
2. Jogar a roupa no cesto
3. Entrar no banheiro
4. Entrar no box
5. Ligar chuveiro
6. Entrar na ducha
7. Ensaboar-se
8. Enxaguar-se
9. Desligar chuveiro
10. Secar-se
11. Sair do box
12. Vestir a roupa



Exercícios

E1. Crie uma sequência lógica para tomar banho.

1. Tirar a roupa
2. Jogar a roupa no cesto
3. Entrar no banheiro
4. Entrar no box
5. Ligar chuveiro
6. Entrar na ducha
7. Ensaboar-se
8. Enxaguar-se
9. Desligar chuveiro
10. Secar-se
11. Sair do box
12. Vestir a roupa

1. Entrar no banheiro
2. Entrar no box
3. Ligar chuveiro
4. Entrar na ducha
5. Ensaboar-se
6. Enxaguar-se
7. Desligar chuveiro
8. Secar-se
9. Sair do box
10. Sair do banheiro



Exercícios

E2. Faça um algoritmo para somar dois números e multiplicar o resultado pelo primeiro número.

```
Algoritmo SomaDoisNumeros
  LER Numero1
  LER Numero2
  soma = Numero1 + Numero2
  produto = Numero1 * soma
  IMPRIME "Soma = " soma
  IMPRIME "Multiplicacao = " produto
Fim-Algoritmo
```




Exercícios

E3. Descreva com detalhes a sequência lógica para Trocar um pneu de um carro.

1. Pare em um local seguro
2. Pegue o triângulo e sinalize a via atrás do veículo
3. Localize o estepe, macaco e chave de roda
4. Afrouxe os parafusos
5. Levante o veículo
6. Solte os parafusos e troque os pneus
7. Aperte os parafusos
8. Desça o carro
9. De volta para a pista

FONTE: <https://autoesporte.globo.com/servicos/noticia/2020/06/como-trocar-o-pneu-do-carro-sem-sustos.ghtml>



Exercícios

E4. Faça um algoritmo para trocar uma lâmpada.

Algoritmo TrocaLampada

Pega Escada

Monta Escada

Sobe Escada

Desatarraxa Lampada queimada

Desce Escada

Pega Lâmpada nova

Sobe Escada

Atarraxa Lampada nova

Desce Escada

Fim-Algoritmo



Dúvidas?

