



# Algoritmos e Lógica de Programação

***Aula 01***

***Apresentação da Disciplina***

**Prof. Dr. Dilermando Piva Jr**

**1º Semestre - DSM**



# Conteúdo Programático

- Método para desenvolvimento de algoritmos; A lógica e os algoritmos; O raciocínio e as formas de resolução de problemas.
- Linearização de expressões matemáticas; Expressões lógico-matemáticas; Tipo de dados.
- Estrutura sequencial; Estrutura condicional simples e composta.
- Estrutura de repetição. Vetores, Matrizes e Strings.
- Modularização de algoritmos (procedimentos e funções)
- Testes de Software. Repositório de SW.

# Conteúdo Programático - Planejamento

Conteúdo Programático		
Semana	Data	Temas/Atividades
1	19/02	Apresentação da disciplina. O Raciocínio e as formas de resolução.
2	26/02	A Lógica e os Algoritmos. Construção do Ambiente de Desenvolvimento.
3	04/03	Expressões Matemáticas e Tipos de Dados
4	11/03	Comandos de Entrada e Saída; Estrutura Sequencial e Funções Básicas
5	18/03	Estrutura Condicional simples e composta
6	25/03	<i>Exercícios de Fixação</i>
7	01/04	Estruturas de Repetição
8	08/04	Estrutura de Dados ( <u>Strings</u> )
9	15/04	Estrutura de Dados (Vetores, Matrizes ou Listas)
10	22/04	<b>Primeira Avaliação (presencial)</b>
11	29/04	Correção da Avaliação. Estrutura de Dados (Coleções: Tuplas, Dicionários e Conjuntos)
12	06/05	Funções
13	13/05	Exercícios sobre funções e demais tópicos.
14	20/05	<i>Exercícios de Fixação</i>
15	27/05	Recursão e Módulos / Pacotes em Python
16	03/06	Trabalhando com arquivos.
17	10/06	Exercícios com módulos e arquivos.
18	17/06	<b>Segunda Avaliação (presencial)</b>
19	24/06	<i>Prova Substitutiva</i>
20	01/07	<i>Divulgação dos Resultados / Fechamento da Disciplina</i>

# Bibliografia...

## **BÁSICA:**

CORMEN, T. H. et al. Algoritmos. Rio de Janeiro: Campus, 2012.

MANZANO, J. A. N. G; OLIVEIRA, J. F. Algoritmos: Lógica para desenvolvimento de programação de computadores. São Paulo: Érica, 2009.

MEDINA, M., FERTIG, C. Algoritmos e Programação: Teoria e Prática. São Paulo: Novatec, 2006

## **COMPLEMENTAR:**

DEITEL, H; DEITEL, P. C: Como programar. 6 ed. São Paulo: Pearson, 2011.

BIANCHI, F. et al. Algoritmos e programação de computadores. Rio de Janeiro: Campus, 2012.

SOUZA, M. A. F. et al. Algoritmos e Lógica de Programação. São Paulo: Cengage Learning, 2019.

MENEZES, O. Introdução à Programação Com Python: Algoritmos e Lógica De Programação para iniciantes. 3 ed. São Paulo: Novatec. 2019.

SILVERMAN, R. E. Git: Guia prático. São Paulo: Novatec, 2019.

# Bibliografia...

## BÁSICA:

**CORMEN, T. H. et al. Algoritmos. Rio de Janeiro: Campus, 2012.**

MANZANO, J. A. N. G; OLIVEIRA, J. F. Algoritmos: Lógica para desenvolvimento

de programação de computadores. São Paulo: Érica, 2012.

MEDINA, M., FERTIG, C. Algoritmos e Programação: Teoria e Prática. São Paulo: Novatec, 2006

## COMPLEMENTAR:

DEITEL, H; DEITEL, P. C: Como programar 6 ed. São Paulo: Pearson, 2012.

BIANCHI, F. et al. Algoritmos: Teoria e Prática. Rio de Janeiro: Campus, 2012.

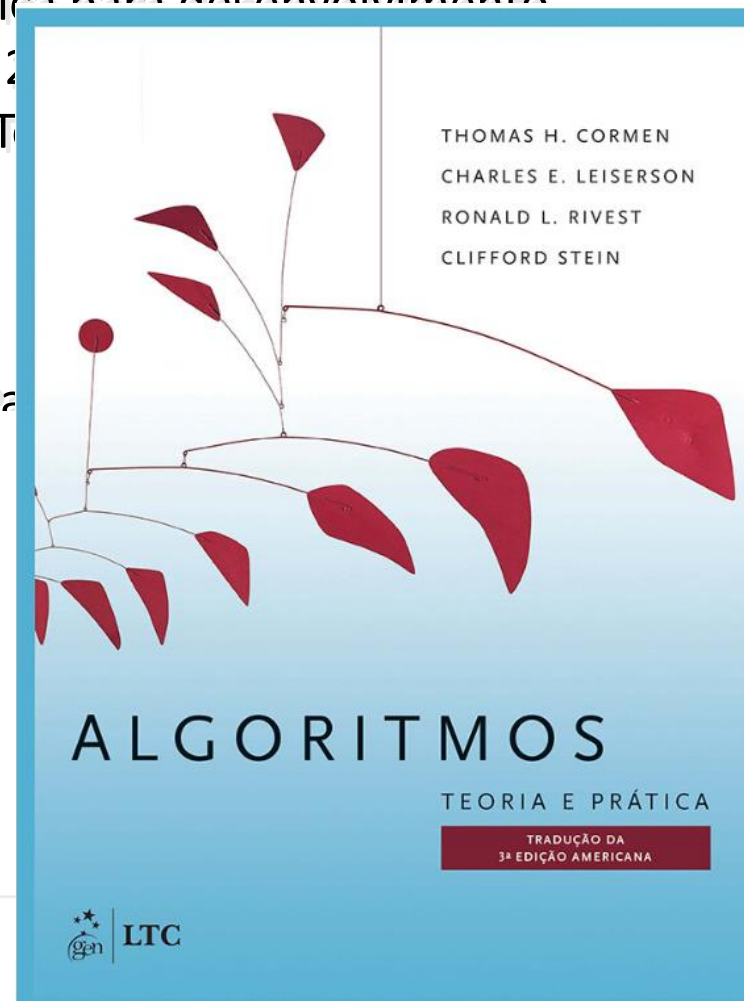
SOUZA, M. A. F. et al. Algoritmos: Teoria e Prática. São Paulo: Learning, 2019.

MENEZES, O. Introdução à Programação para iniciantes. São Paulo: LTC, 2012.

SILVERMAN, R. E. Git: Guia de Referência. São Paulo: Novatec, 2012.



Thomas Cormen  
**Algoritmos - Teoria e Prática**  
★★★★★  
190 classificações  
Capa comum: R\$384,90  
Adicionar ao carrinho  
Vendido e enviado por Amazon.com.br.  
Veja mais opções de compra





# Bibliografia...

## BÁSICA:

CORMEN, T. H. et al. Algoritmos. Rio de Janeiro: Campus, 2012.

MANZANO, J. A. N. G; OLIVEIRA, J. F. Algoritmos: Lógica para desenvolvimento de programação de computadores. São Paulo: Érica, 2009.

MEDINA, M., FERTIG, C. Algoritmos e Programação. São Paulo: Novatec, 2006

## COMPLEMENTAR:

DEITEL, H; DEITEL, P. C: Como programar. 6 ed

BIANCHI, F. et al. Alg  
Campus, 2012.

SOUZA, M. A. F. et a  
Learning, 2019.

MENEZES, O. Introdu  
Programação para ir

SILVERMAN, R. E. Gi



Algoritmos: Lógica  
Para Desenvolvimento  
de Programação de  
Computadores - Edição  
Revisada e Atualizada



507 classificações

Capa comum: R\$90,77



Adicionar ao carrinho

Vendido e enviado por Amazon.com.br.



# Bibliografia...

## **BÁSICA:**

CORMEN, T. H. et al. Algoritmos. Rio de Janeiro: Campus, 2012.

MANZANO, J. A. N. G; OLIVEIRA, J. F. Algoritmos: Lógica para desenvolvimento de programação de computadores. São Paulo: Érica, 2009.

MEDINA, M., FERTIG, C. Algoritmos e Programação: Teoria e Prática. São Paulo: Novatec, 2006

## **COMPLEMENTAR:**

DEITEL, H; DEITEL, P. C: Como programar. 6 ed. São Paulo: Campus, 2012.

BIANCHI, F. et al. Algoritmos e programação de computadores. São Paulo: Campus, 2012.

SOUZA, M. A. F. et al. Algoritmos e Lógica de Programação. São Paulo: Learning, 2019.

MENEZES, O. Introdução à Programação para iniciantes.

SILVERMAN, R. E. Git: Guia para iniciantes.

[Ver todos os formatos e edições](#)

Capa Comum  
a partir de R\$ 150,00

1 Usado a partir de R\$ 150,00



# Bibliografia...

## **BÁSICA:**

CORMEN, T. H. et al. Algoritmos. Rio de Janeiro: Campus, 2001.  
MANZANO, J. A. N. G; OLIVEIRA, J. F. Algoritmos: Lógica de programação de computadores. São Paulo: Érica, 2005.  
MEDINA, M., FERTIG, C. Algoritmos e Programação: Teoria e Prática. São Paulo: Novatec, 2006

## **COMPLEMENTAR:**

DEITEL, H; DEITEL, P. C: Como programar. 6 ed. São Paulo: Pearson, 2011.  
BIANCHI, F. et al. Algoritmos e Programação. São Paulo: Campus, 2012.  
SOUZA, M. A. F. et al. Algoritmos e Programação. São Paulo: Learning, 2019.

Comprar novo: **R\$ 57,00**  
De: ~~R\$ 89,00~~  
Você economiza: R\$ 32,00 (36%)  
prime

**MENEZES, O. Introdução à Programação Com Python: Algoritmos e Lógica De Programação para iniciantes. 3 ed. São Paulo: Novatec. 2019.**

SILVERMAN, R. E. Git: Guia prático. São Paulo: Novatec, 2019.





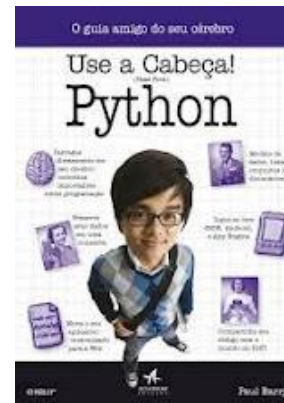
# Bibliografia em Python...

## Linguagem:

LAMBERT, K. A. Fundamentos de Python: primeiros programas.  
Editora CENGAGE, 2022



BARRY, P. Python. Use a Cabeça! Editora Altabooks. 2018



## Algoritmos:

BIANCHI, F. et al. Algoritmos e programação de computadores.  
2ª.Ed. RJ: Campus/Elsevier, 2012.



# Forma de Avaliação

Instrumentos de Avaliação		Cronograma	Pesos
<b>Nota 1:</b>	Prova 1	<b>23/04</b>	<b>25%</b>
<b>Nota 2:</b>	Prova 2	<b>18/06</b>	<b>35%</b>
<b>Nota 3:</b>	Atividades (A.F.)	Ao longo das aulas (portfólio)	<b>15%</b>
<b>Nota 4:</b>	Projeto Interdisciplinar	Ao longo do semestre	<b>20%</b>
<b>Nota 5:</b>	Arredondamento	Aulas e Portfólio	-
<b>Nota 6:</b>	Avaliação Substitutiva	<b>25/06</b>	<b>P1 ou P2</b>

## Atenção:

A **avaliação substitutiva**, é aplicada aos estudantes que por ventura venham a perder alguma das duas avaliações presenciais (P1 ou P2). Existe a necessidade de comprovação formal do motivo da ausência.

# Algoritmos...

**Site / Blog da Disciplina:**

**http://www.piva.pro.br/**

## Algoritmos e Lógica de Programação

quarta-feira, 12 de julho de 2017

### Semana 00 - Orientações Gerais

#### Aulas

As aulas acontecerão nos dias e horários planejados no Sistema Acadêmico (SIGA). Por ser um curso presencial, exige a presença em pelo menos 75% dos 20 encontros (80 aulas) planejadas. Programação oficial da disciplina (igual o lançado no sistema SIGA) - [<aqui>](#)

#### Livros Textos

Qualquer livro de linguagem C pode ser utilizado/consultado. Essencialmente, utilizarmos como base, dois livros didáticos. São eles:

**Fundamentos da Programação de Computadores: algoritmos, Pascal, C/C++ e Java.**  
Editora Logman.  
Autoras: Ascencio, A.F.G. e Campos, E.A.V.

Quem sou eu



**Dilermando Piva Jr.**  
Seguir 337

Visualizar meu perfil completo

Arquivo do blog

▼ 2017 (21)

▼ Julho (21)

Semana 00 - Orientações Gerais

Semana 01

Semana 02

Semana 03



Prof. Dr. Dilermando Piva Jr.

[Principal](#) [Sobre](#) [Contato](#)

[Projeto pyPRO](#)

## Inscriva-se no Canal pyPRO

Inscriver...

Disciplinas...

### Sistemas M\_processados e M\_controlados

Disciplina no curso de Mecatrônica da Fatec Itu (4º Sem - Manhã)

### Algoritmos e Estrutura de Dados I

Disciplina no curso de Ciência de Dados para Negócios (1º Sem - Noturno)

### Algoritmos e Lógica de Programação

Disciplina no curso de DSM da Fatec Votorantim (1º Sem)

### Info. Aplic. à Gestão da Qualidade

Disciplina no curso de Gestão da Qualidade na Fatec Sorocaba (3º sem - Manhã)

Livros Publicados

# O que é um Algoritmo?



# O que é um Algoritmo?

Uma resposta de âmbito geral seria...

**“um conjunto de etapas para executar uma tarefa!”**

A vida é feita de algoritmos.

**Um algoritmo para escovar  
os dentes.**





# Algoritmo para escovar os dentes...

1. Abrir o tubo de pasta dental
2. Pegar a escova de dentes
3. Apertar o tubo de pasta dental sobre a escova e aplicar a qtd necessária de dentífrico
4. Fechar o tubo
5. Colocar a escova em um quadrante da boca
6. Movimentá-la para cima e para baixo durante alguns segundos etc.



# Outros exemplos de algoritmos em nossas vidas

- Se você pega ônibus ou metrô ou trem para ir trabalhar, terá um algoritmo.
- Se desejar fazer um sanduiche...
- Se tiver que trocar um pneu furado...
- Se for tomar banho...

E assim por diante!

# Algoritmos de Computadores

- **Você já utilizou um GPS para determinar uma rota de viagem?**
  - O aparelho executa um algoritmo denominado algoritmo de “caminho mínimo” para determinar essa rota!!
- **Você já fez ou faz compras na Internet?**
  - Você deve (ou deveria) ter utilizado um site seguro, que utiliza um algoritmo para criptografar e proteger os dados.
  - Tem também algoritmos que indicam possibilidades de produtos que você gostaria de comprar (algoritmos de recomendação)
- Esses algoritmos são executados em todos os lugares, em laptops, tablets, celulares, servidores etc.

# Qual a diferença entre esses tipos de Algoritmos?

Algoritmo que você executa

vs.

Algoritmo que um computador executa

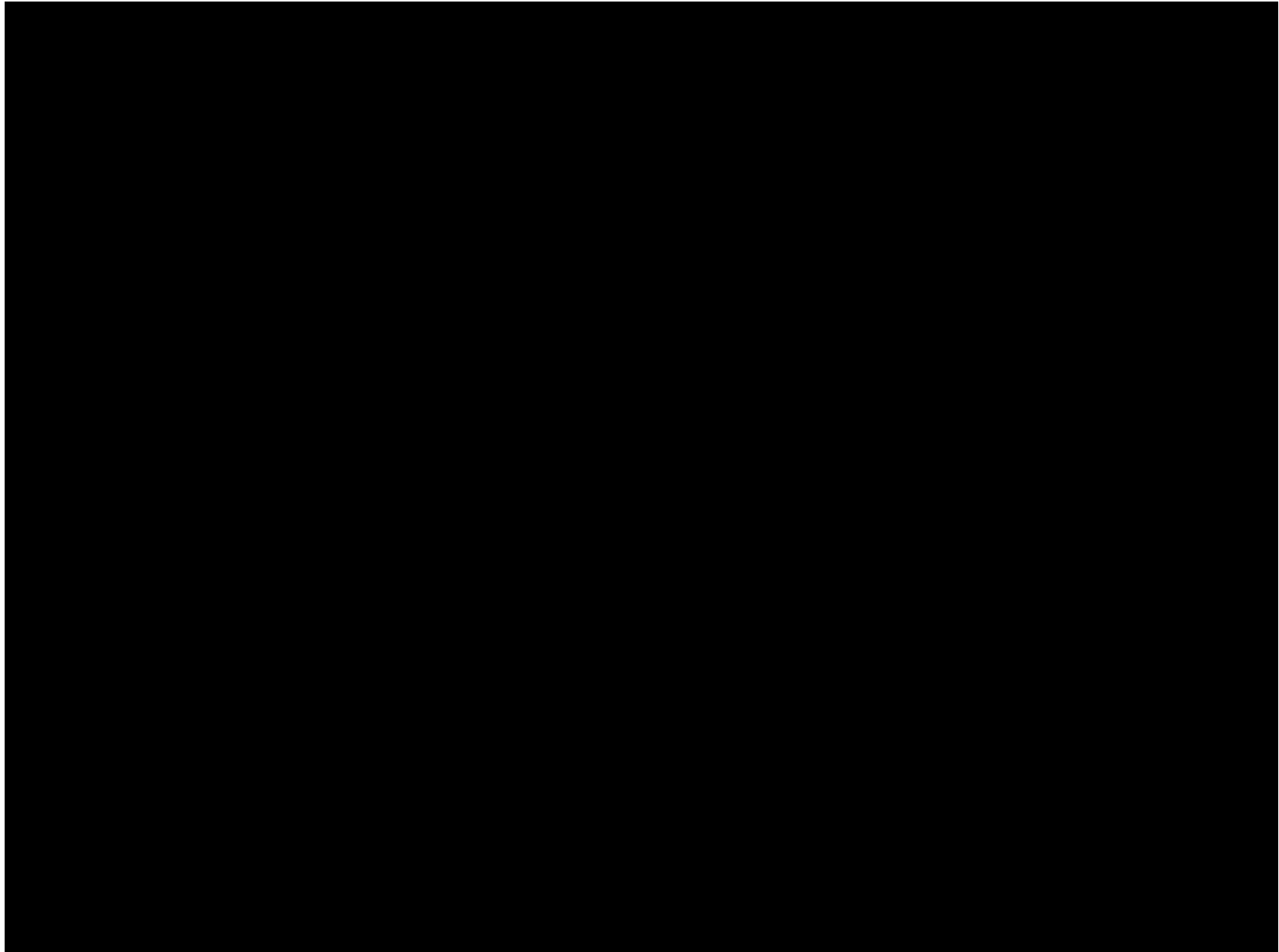


# Algoritmo Computacional

- Um algoritmo de computador é um **conjunto de etapas** para executar uma tarefa descrita com **precisão suficiente** para que o computador possa executá-la!!



# Exemplo de Linguagem...



# Linguagens de Programação

## PASCAL

```
program Hello;  
var mensagem : string;  
begin  
  mensagem := 'Hello World!';  
  write(mensagem);  
  
End.
```

## JAVA

```
public class Main {  
  public Main(){  
    System.out.println("Hello World");  
  }  
  public static void main(String [] args){  
    Main m =new Main();  
  }  
}
```

## C

```
#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
  printf("HELLO WORLD!!!");  
  return(0);  
}
```

## COBOL

```
function Hello (){  
  alert("Hello World!")  
}
```

# Linguagens de Programação

## PASCAL

```
program Hello;  
var mensagem : string;  
begin  
  mensagem := 'Hello World!';  
  write(mensagem);  
  
End.
```

## JAVA

```
public class Main {  
  public Main(){  
    System.out.println("Hello World");  
  }  
  public static void main(String [] args){  
    Main m =new Main();  
  }  
}
```

## C

```
#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
  printf("HELLO WORLD!!!");  
  return(0);  
}
```

## COBOL











```
function Hello (){  
  alert("Hello World!")  
}
```

## PYTHON

```
print("Hello World!")
```

# Qual linguagem utilizar?

<https://www.tiobe.com/tiobe-index/>

Feb 2024	Feb 2023	Change	Programming Language		Ratings	Change
1	1			Python	15.16%	-0.32%
2	2			C	10.97%	-4.41%
3	3			C++	10.53%	-3.40%
4	4			Java	8.88%	-4.33%
5	5			C#	7.53%	+1.15%
6	7	^		JavaScript	3.17%	+0.64%
7	8	^		SQL	1.82%	-0.30%
8	11	^		Go	1.73%	+0.61%
9	6	v		Visual Basic	1.52%	-2.62%
10	10			PHP	1.51%	+0.21%

# Qual linguagem utilizar?

Em Programação

Aquela que você conhece  
**PROFUNDAMENTE !!!**



# Algoritmo Computacional

- Um algoritmo de computador é um conjunto de etapas para executar uma tarefa descrita com **precisão suficiente** para que o computador possa executá-la!!

# O que se pretende de um algoritmo de computador?

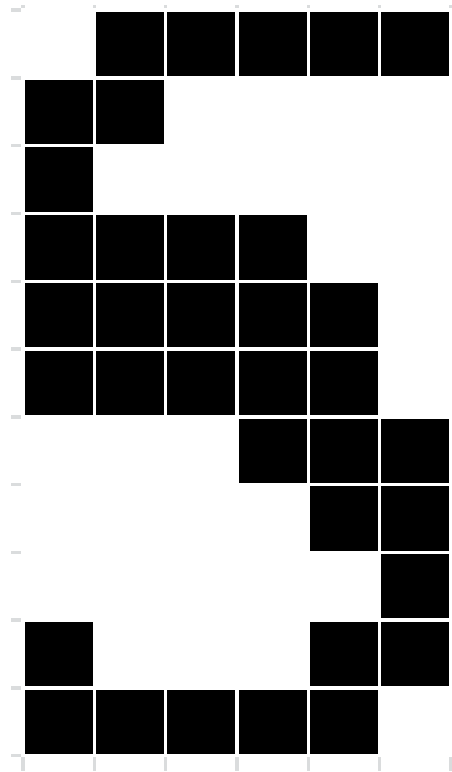
Algoritmos computacionais são feitos para resolver problemas...

- O algoritmo deve fornecer uma solução **correta** para o problema
- O algoritmo deve usar recursos computacionais **eficientemente** ao resolver o problema.

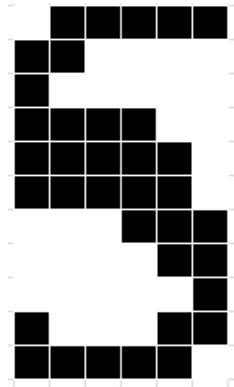
# Solução correta...

- O que significa produzir uma solução correta para um problema?
- **GPS**
  - Menor rota?
  - Mais rápido?
  - Mais barato (sem pagar pedágio)?

# Solução correta?



# Solução correta?



Um algoritmo de reconhecimento de caracteres ópticos, por exemplo.

**É um 5 ou um S?**

Podemos dizer que esse algoritmo produz um **resultado incorreto**?



# Sempre correta?

- Um algoritmo computacional sempre nos dará um resultado correto?
- Podemos aceitar um algoritmo que pode produzir uma resposta incorreta?



# Sempre correta?

- Um algoritmo computacional sempre nos dará um resultado correto?
- Podemos aceitar um algoritmo que pode produzir uma resposta incorreta?
- **DESDE QUE POSSAMOS CONTROLAR A FREQUÊNCIA COM QUE ISSO ACONTECE.... SIM!**

# Sempre correta?

## Exemplo

- Um algoritmo computacional sempre nos dará um resultado correto?
- Podemos aceitar um algoritmo que pode produzir uma resposta incorreta?
- **DESDE QUE POSSAMOS CONTROLAR A FREQUÊNCIA COM QUE ISSO ACONTECE.... SIM!**

## O Criptossistema RSA

- Dado um numero N (*grande*) determina se é PRIMO ou NÃO.
- 1 erro a cada  $2^{50}$  vezes
- Mais de um trilhão de vezes
- **1.125.899.906.842.620**

# O que se pretende de um algoritmo de computador?

Algoritmos computacionais são feitos para resolver problemas...

- O algoritmo deve fornecer uma solução **correta** para o problema
- O algoritmo deve usar recursos computacionais **eficientemente** ao resolver o problema.



# Eficientemente?

- O que significa um algoritmo usar recursos computacionais eficientemente?
- Pense novamente no GPS...
- Se ele demorar uma hora para determinar qual rota ele recomenda, você o usaria?

# Eficientemente?



## Tempo!!

É a medida principal de eficiência que usamos para avaliar um algoritmo.

# Eficientemente?

- Quantidade de memória (RAM limitada)
- Comunicação em rede (necessita de informações que estão em outro local)
- Operações em disco (HD, SSD...)

# Pergunta que não quer calar....

Por que eu tenho que me **preocupar** com ALGORITMOS de computador (ou computacionais)?





# Algoritmos de computador?



# Algoritmos de computador?

Muitas oportunidades!



CODE.ORG

Em 2030 haverá uma demanda de  
1.4 milhão de programadores



→ ritmo atual: 400 mil

1.000.000 de vagas



# Algoritmos de computador?

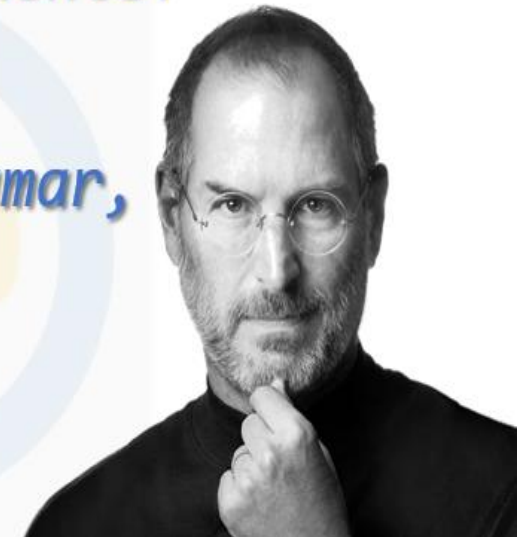
- Não só desktops ou notebooks
- Existe uma infinidade de dispositivos:
  - Celulares
  - Relógios (Smartwatch)
  - Tvs Inteligentes
  - Carros inteligentes
  - Equipamentos usáveis
  - IoT (Internet das Coisas)
  - Servidores
  - Programação em nuvem...

# Algoritmos de computador?

- Aprender a programar faz bem para nossa vida!
- Nos ajuda a pensar melhor... Mais logicamente... Raciocínio mais rápido.
- Diferencial (social e profissional)
- Nos ajuda a resolver problemas mais rapidamente!

*“Todos nesse país deveriam aprender a programar, pois isso nos ensina a pensar.”*

Steve Jobs





# Algoritmos de computador?

- VELHO ou JOVEM
- HOMEM ou MULHER

Não existe idade, raça, gênero ...

Existe sim...

- FORÇA DE VONTADE
- OBJETIVO
- DETERMINAÇÃO!

# Maiores Obstáculos?

- Falta de compromisso / objetivo
- Ter pressa
  - Aprender um novo idioma  
(não adianta ler o dicionário todo!!)
- Não praticar (fazer os exercícios/desafios)
- Dedicção do início ao fim
- Importância dos fundamentos (básico)
- Desafios
  - Eles te farão crescer

# O que é uma L.P.?



# O que é uma L.P.?





# O que é uma L.P.?



# O que é uma L.P.?

## *Compilada vs Interpretada*

### Linguagens Compiladas

- Do latim *compilare*
- Significa REUNIR, AJUNTAR
- Fonte traduzido diretamente para linguagem de máquina

# O que é uma L.P.?

## *Compilada vs Interpretada*

### Processo de Compilação

Programa (Alto nível)

Análise léxica

Análise sintática

Geração de código (Baixo nível)

Programa (Objeto executável)

# O que é uma L.P.?

## *Compilada vs Interpretada*

### Processo de Compilação

- Análise léxica  
Reconhece as sequências de símbolos que representam uma unidade: o nome de uma variável, constante e palavras de instrução (while, for...)

```
pos = ini + val * 60
```

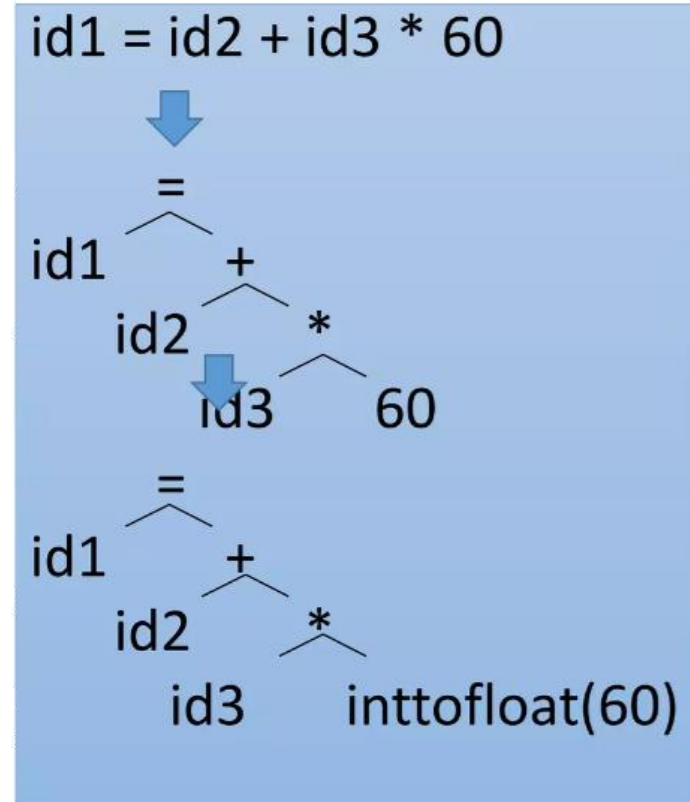
```
id1 = id2 + id3 * 60
```

# O que é uma L.P.?

## *Compilada vs Interpretada*

### Processo de Compilação

- Análise sintática  
Identifica a estrutura gramatical do programa e o papel de cada componente. É construída uma *árvore sintática* (binária, estruturalmente) e uma tabela de símbolos que representam as variáveis.

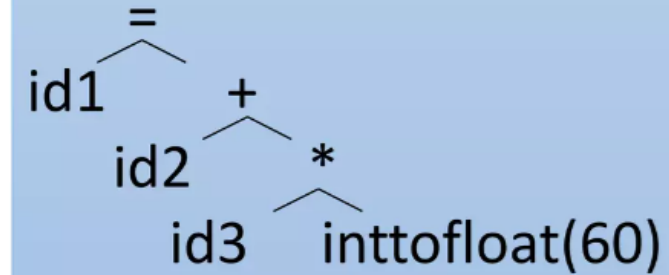


# O que é uma L.P.?

## *Compilada vs Interpretada*

### Processo de Compilação

- Geração de código
- Processo de construir instruções da linguagem de máquina (em assembly) que simulam as instruções reconhecidas na análise sintática.



temp1 = inttofloat(60)  
temp2 = id3 \* temp1  
temp3 = id2 + temp2  
id1 = temp3

//otimização  
temp1 = id3 \* 60.0  
id1 = id2 + temp1



# O que é uma L.P.?

## *Compilada vs Interpretada*

### Processo de Compilação

- Geração de código
- Processo de construir instruções da linguagem de máquina (em assembly) que simulam as instruções reconhecidas na análise sintática.

```
temp1 = id3 * 60.0  
id1 = id2 + temp1
```



```
load    id3    r2  
mul     60.0   r2  
load    id2    r1  
add     r2     r1  
store   r1     id1
```

# O que é uma L.P.?

## *Compilada vs Interpretada*

### Linguagens Compiladas

- Vantagens:
  - Execução rápida, o código já está traduzido!
  - Executáveis são pequenos
  - VOCÊ CONTROLA o hardware
- Desvantagens:
  - Só roda na arquitetura em que for compilado
  - VOCÊ CONTROLA o hardware



# O que é uma L.P.?

## *Compilada vs Interpretada*

### Linguagens Interpretadas

- Do latim *interpretare*
- Significa EXPLICAR, TRADUZIR
- Fonte traduzido para uma linguagem intermediária (normalmente Bytecode) que mais tarde será novamente traduzido para linguagem de máquina

# O que é uma L.P.?

## *Compilada vs Interpretada*

### Processo de Interpretação

Programa (Alto nível)

Programa (Intermediário/Executável)

Máquina Virtual (Interpretadora)

# O que é uma L.P.?

## *Compilada vs Interpretada*

### Processo de Interpretação

Programa (Alto nível)

Programa (Intermediário/Executável)

Máquina Virtual (Interpretadora)

# O que é uma L.P.?

## *Compilada vs Interpretada*

### Processo de Interpretação

- Máquina virtual (VM)

Traduz os comandos da linguagem intermediária para linguagem de máquina em tempo de execução. A VM reconhece toda a arquitetura de hardware e traduz de acordo com o equipamento (culpa do Java).

# O que é uma L.P.?

## *Compilada vs Interpretada*

### Linguagens Interpretadas

- Vantagens
  - Indepe<sup>nde</sup> de arquitetura (desde que suporte a máquina virtual)
  - A MÁQUINA VIRTUAL controla o hardware
- Desvantagens
  - Desempenho inferior à linguagem compilada por causa da tradução Just-in-time, mas nada alarmante
  - Consome bastante hardware (estou olhando pra você, Java)
  - A MÁQUINA VIRTUAL controla o hardware

# O que é uma L.P.?

## *Compilada vs Interpretada*





# VAMOS PARA A PRÁTICA ?!!!

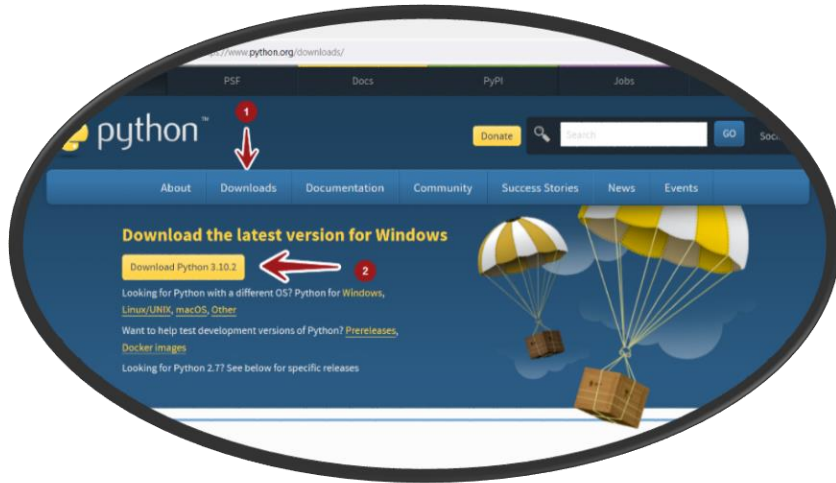


# Por que Python?

- Simples o suficiente para um curso introdutório
- Muitos recursos
  - Orientação a Objetos
  - Escalável (módulos, classes, controle de exceções)
  - Biblioteca embutida extensa e grande número de módulos fornecidos por terceiros
- Grande variedade de aplicações
- Linguagem interpretada (script)
- Multi-plataforma
- Grátis!
- Comunidade bastante grande



# Muitas formas...



**<https://python.org/downloads/>**

1



```
Prompt de Comando - python
C:\Users\Piva>python
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
>>> 2+3
5
>>> print("Olá!")
Olá!
>>>
```

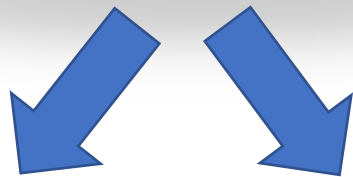
# Muitas formas...



<https://python.org/downloads/>

1

2

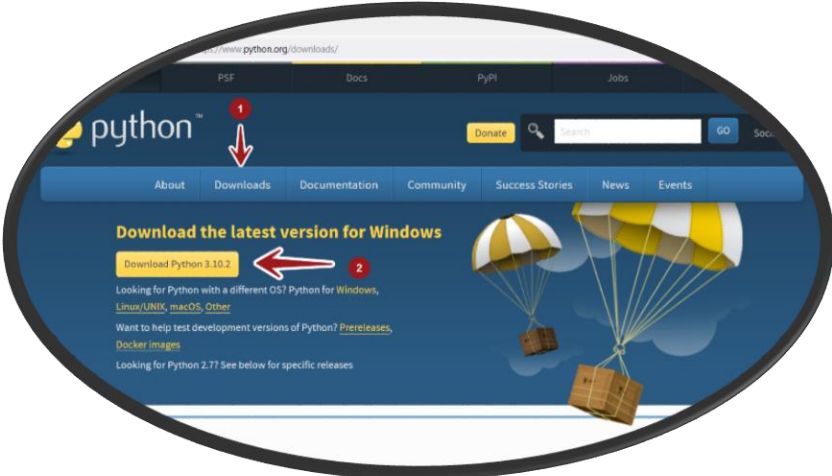


**IDE**



```
Prompt de Comando - python
C:\Users\Piva>python
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2+3
5
>>> print("Olá!")
Olá!
>>>
```

# Muitas formas...



<https://python.org/downloads/>

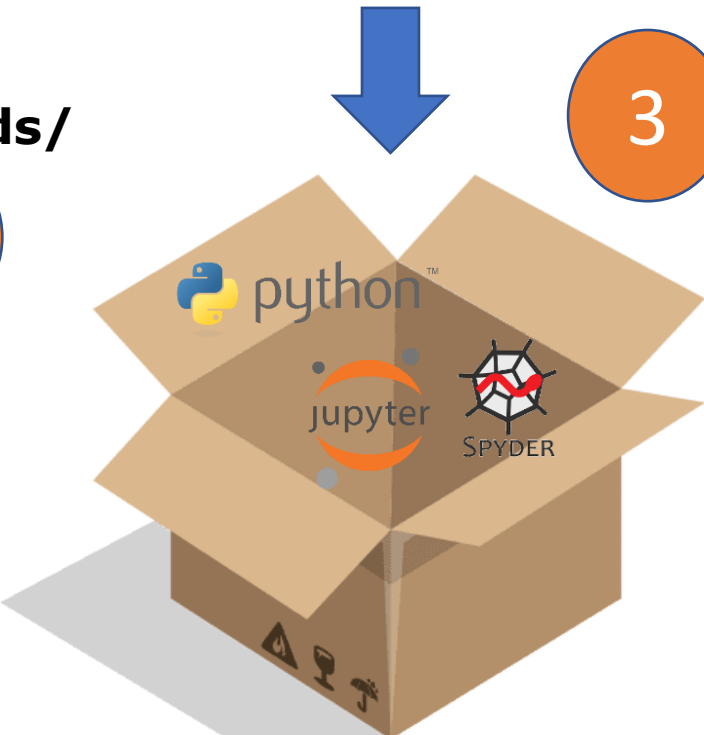
1

```
Prompt de Comando - python
C:\Users\Piva>python
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
>>> 2+3
5
>>> print("Olá!")
Olá!
>>>
```

2



3

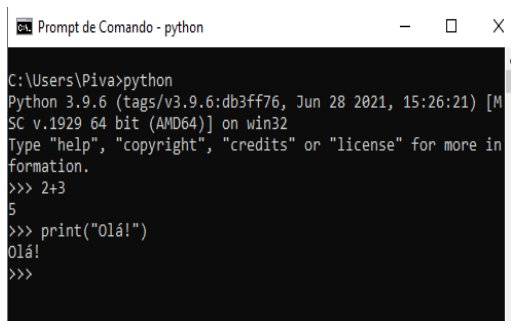


# Muitas formas...



<https://python.org/downloads/>

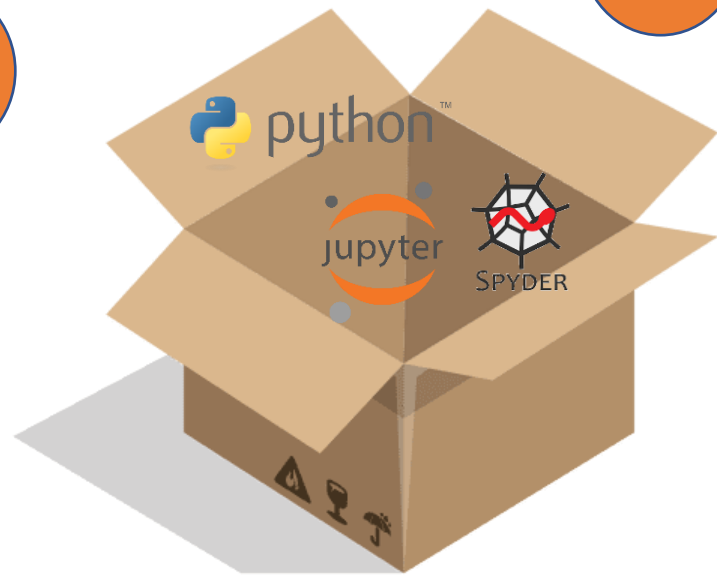
1



2

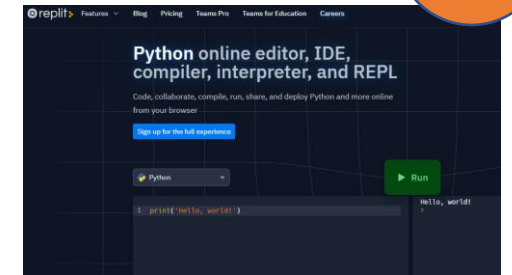


3



## SERVIÇOS ONLINE

4



Navegador  
<https://replit.com/>

App p/ Celular  
QPython3



# Muitas formas...



```
Prompt de Comando - python
C:\Users\Piva>python
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
>>> 2+3
5
>>> print("Olá!")
Olá!
>>>
```