



Oficina de vestibulandos 2019



PET
COMPUTAÇÃO

Algoritmos

Conjunto de instruções

Exemplos no cotidiano de algoritmos:

- Receita de bolo
- Resolução de uma equação de 2º grau

$$x^2 - 3x - 10 = 0$$

$$a = 1$$

$$b = -3$$

$$c = -10$$

$$\Delta = b^2 - 4ac$$

$$\Delta = (-3)^2 - 4 * 1 * (-10)$$

$$\Delta = 9 + 40$$

$$\Delta = 49$$

$$x = \frac{-b \pm \sqrt{\Delta}}{2a} \Rightarrow x = \frac{-(-3) \pm \sqrt{49}}{2 * 1}$$

$$x = \frac{3 \pm 7}{2} \begin{cases} x' = \frac{3+7}{2} = 5 \\ x'' = \frac{3-7}{2} = -2 \end{cases}$$

MODO DE PREPARO

Bata as claras em neve e reserve.

Misture as gemas, a margarina e o açúcar até obter uma massa homogênea.

Acrescente o leite e a farinha de trigo aos poucos, sem parar de bater.

Por último, adicione as claras em neve e o fermento.

Despeje a massa em uma forma grande de furo central untada e enfarinhada.

Asse em forno médio 180 °C, preaquecido, por aproximadamente 40 minutos ou ao furar o bolo com um garfo, este saia limpo.

EXERCÍCIO

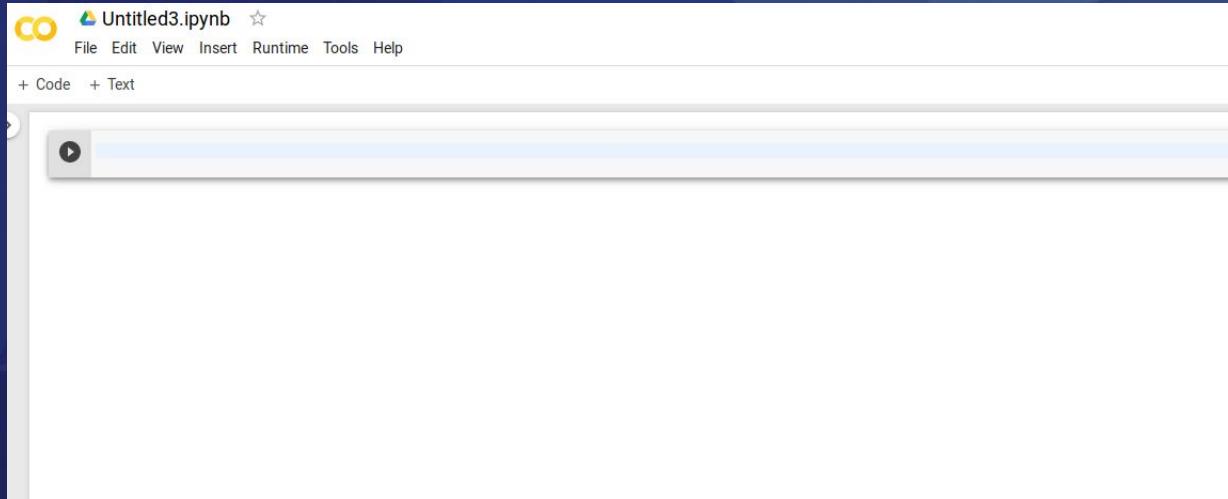
Escreva um algoritmo de alguma coisa que você faz no dia-a-dia!

Iremos usar um editor e compilador de código online

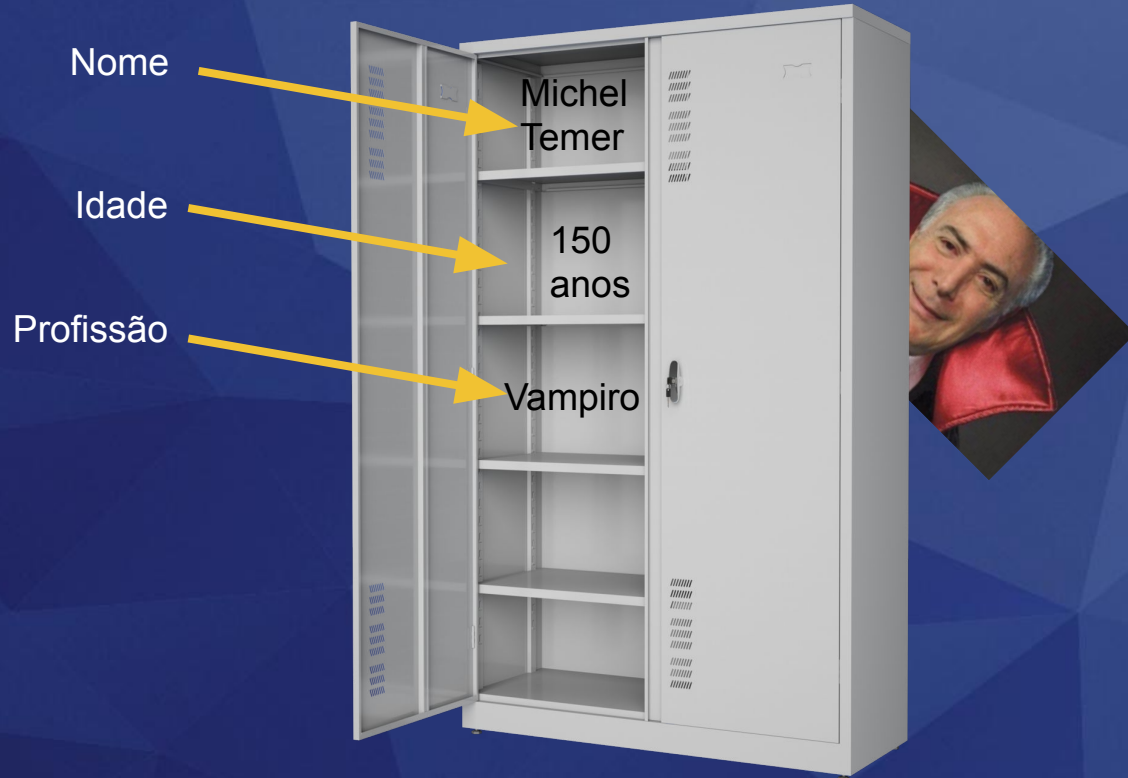
Por favor, clique no link e abra pelo navegador

<https://colab.research.google.com/notebook#create=true&language=python3>

Se tudo ocorreu bem, então você verá uma tela parecida com essa:



Variáveis e memória



Python

Python é uma linguagem de programação criada por Guido van Rossum em 1991, com objetivo de ser produtiva e legível, com código fácil de se entender e de se produzir.

Suas características são:

- baixo uso de caracteres especiais, o que torna a linguagem muito parecida com *pseudo-código executável*;
- o uso de indentação para marcar blocos;

Entrada e saída de dados

Entrada:

- É como o usuário se “comunica” com o computador
- O programador não precisa ficar alterando o programa sempre que vai usar, ou seja, depois do programa estar pronto, não precisa mais do programador
- Se o programa for visto como uma função, a entrada é a variável da função (o “x” de uma $f(x)$)

Saída:

- É a resposta do computador para o usuário
- Se um programa for visto como uma função, a saída é a imagem da função (o “y” de uma $f(x)$)

Exemplo entrada de dados

```
1 # exemplo media
2 qnt_nota = 3 # atribuir a variavel qnt_nota o valor 3
3 nota1 = input() # nota1 vai receber o valor que o usuario digito
4 nota2 = input() # nota2 vai receber o valor que o usuario digito
5 nota3 = input() # nota3 vai receber o valor que o usuario digito
6 media = (nota1 + nota2 + nota3) / qnt_nota
```

No python, para o programa interpretar o que o usuário digitou, a forma é

variável = input()

Exemplo saída de dados

```
1 # exemplo media
2 qnt_nota = 3 # atribuir a variavel qnt_nota o valor 3
3 nota1 = input() # nota1 vai receber o valor que o usuario digitou
4 nota2 = input() # nota2 vai receber o valor que o usuario digitou
5 nota3 = input() # nota3 vai receber o valor que o usuario digitou
6 media = (nota1 + nota2 + nota3) / qnt_nota
7 print(media)
```

No python, para o programa mostrar informações ao usuário, a forma é

```
print(variável)
```

Atribuição de variáveis

Números:

- `x = 10`
- `y = 8`
- `metade = 0.5`

Caracteres (strings):

- `morango = 'vermelho'`
- `programar = 'bom'`
- `hoje = 'sexta'`

Operações

- $x + y$
- $z = x + y$
- $w = x - y$

← não salva na memória
← salva na memória

- $media = z * metade$
- $um_sexto = media / 3$

- $um_sexto \% 2$

soma: +
subtração: -

multiplicação: *
divisão: /

resto: %

Decisões

- `if ()`:
- `elif ()`:
- `else`:

Maior:	<code>></code>
Menor:	<code><</code>
Maior ou igual:	<code>>=</code>
Menor ou igual:	<code><=</code>
Igual:	<code>==</code>
Diferente:	<code>!=</code>

Operadores lógicos (AND e OR)

Usaremos false como 0 e true como 1

AND (multiplicação)

```
0 AND 0 = 0
1 AND 0 = 0
0 AND 1 = 0
1 AND 1 = 1
```

OR (soma)

```
0 OR 0 = 0
1 OR 0 = 1
0 OR 1 = 1
1 OR 1 = 1
```

AND

```
False and False = False
False and True = False
True and False = False
True and True = True
```

OR

```
False or False = False
False or True = True
True or False = True
True or True = True
```

Comparando dois números



```
banana = 6
manga = 5
if (banana > manga):
    print('banana tem mais letras')
if (manga > banana):
    print('manga tem mais letras')
if (manga == banana):
    print('mesma quantidade de letras')
```



```
banana tem mais letras
```

Comparando dois números

```
▶ banana = 6
  manga = 5
  if (banana > manga):
    print('banana tem mais letras')
  elif (manga > banana):
    print('manga tem mais letras')
  elif (manga == banana):
    print('mesma quantidade de letras')
```

↳ banana tem mais letras

Comparando dois números

```
▶ banana = 6
  manga = 5
  if (banana > manga):
      print('banana tem mais letras')
  elif (manga > banana):
      print('manga tem mais letras')
  else:
      print('mesma quantidade de letras')
```

```
↳ banana tem mais letras
```


Comparando strings

```
▶ usuario = input()
if (usuario == 'marcos'):
    print('faaala marquinhos!')
elif (usuario == 'pedro'):
    print('daeee pedrao!')
else:
    print('Bem vindo ao Departamento de Informatica!')
```

```
↳ marcos
faaala marquinhos!
```

```
▶ usuario = input()
if (usuario == 'marcos'):
    print('faaala marquinhos!')
elif (usuario == 'pedro'):
    print('daeee pedrao!')
else:
    print('Bem vindo ao Departamento de Informatica!')
```

```
↳ pedro
daeee pedrao!
```

```
▶ usuario = input()
if (usuario == 'marcos'):
    print('faaala marquinhos!')
elif (usuario == 'pedro'):
    print('daeee pedrao!')
else:
    print('Bem vindo ao Departamento de Informatica!')
```

```
↳ Vinicius
Bem vindo ao Departamento de Informatica!
```

Trabalhando com comparadores lógicos

OR

```
▶ dia = input('Dia: ')
  if (dia == 'sabado') or (dia == ('domingo')):
    print('O RU do politécnico está fechado!')
  else:
    print('O Ru do politécnico está aberto!')
```

```
↳ Dia: sabado
  O RU do politécnico está fechado!
```

```
▶ dia = input('Dia: ')
  if (dia == 'sabado') or (dia == ('domingo')):
    print('O RU do politécnico está fechado!')
  else:
    print('O Ru do politécnico está aberto!')
```

```
↳ Dia: segunda
  O Ru do politécnico está aberto!
```

AND

```
▶ dia = input('Dia: ')
  lugar = input('Lugar :')

  if (dia == "sexta") and (lugar == "laboratorio"):
    print('Você está no lugar certo')
  else:
    print('Você está perdido')
```

```
↳ Dia: sexta
  Lugar :laboratorio
  Você está no lugar certo
```

```
▶ dia = input('Dia: ')
  lugar = input('Lugar :')

  if (dia == "sexta") and (lugar == "laboratorio"):
    print('Você está no lugar certo')
  else:
    print('Você está perdido')
```

```
↳ Dia: sexta
  Lugar :disney
  Você está perdido
```

Exercício

Tente inventar seu próprio IF ou tome como base algum dos exemplos abaixo:

- Declare uma variável com um número e verifique se é menor, se menor, imprima “É menor” SENAO “Não é menor”
- Declare uma variável suplemento com o valor “whey”, SE suplemento igual a “whey” então imprima “VEM MONSTRO”



While

Repetições são importantes para automatizar tarefas muito extensas, sem que o programador tenha que fazer um código gigante e repetitivo.

Exemplos:

- Descobrir se um número é primo
- Calcular uma série finita
- Calcular quanto irei receber daqui a X meses aplicando os juros do meu banco

```
▶ contador = 1
while (contador <= 7):
    print ("Gol da Alemanha")
    contador = contador + 1
```

↳ Gol da Alemanha
Gol da Alemanha
Gol da Alemanha
Gol da Alemanha
Gol da Alemanha
Gol da Alemanha
Gol da Alemanha



Exemplo while na prática

Estático

```
#Imprima todos os números de 1 até 10
n = 10
contador = 1
while (contador <= n):
    print(contador)
    contador = contador + 1
```

```
↳ 1
   2
   3
   4
   5
   6
   7
   8
   9
  10
```

Dinâmico

Necessário usar int() e input()

```
#Imprima todos os números de 1 até N (número fornecido pelo usuário)
n = int(input())
contador = 1
while (contador <= n):
    print(contador)
    contador = contador + 1
```

```
↳ 15
   1
   2
   3
   4
   5
   6
   7
   8
   9
  10
  11
  12
  13
  14
  15
```

Agora que sabemos lidar com IF e WHILE, vamos realizar alguns exercícios

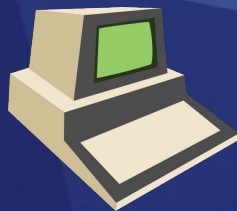
Problema 1: Imprima os números inteiros de 1 a 1000.

Problema 2: Imprima só os pares de 1 a 1000.

Problema 3: Imprima só os múltiplos de 5 entre 1 e 1000.

Problema 4: Imprima só os múltiplos de 100 de 1 a 1000.

Esperamos que tenha gostado!
Obrigado, futuro calourx
Não se esqueça do nosso feedback



PET
COMPUTAÇÃO