
Python 101

Prof. Mateus Mendelson
mendelson.mateus@gmail.com

mmendelson.com



1. Introdução

- Python é uma linguagem interpretada (em contraste com linguagens compiladas) de propósito geral criada por Guido van Rossum em 1991.
- A criação do Python teve início como um projeto pessoal de Guido para se ocupar durante as férias de fim de ano (workaholic).
- Ele decidiu criar uma linguagem que substituísse a linguagem de programação ABC.
- O nome Python foi uma escolha irônica feita por Guido por ser fã do show *Monty Python*.



1. Introdução

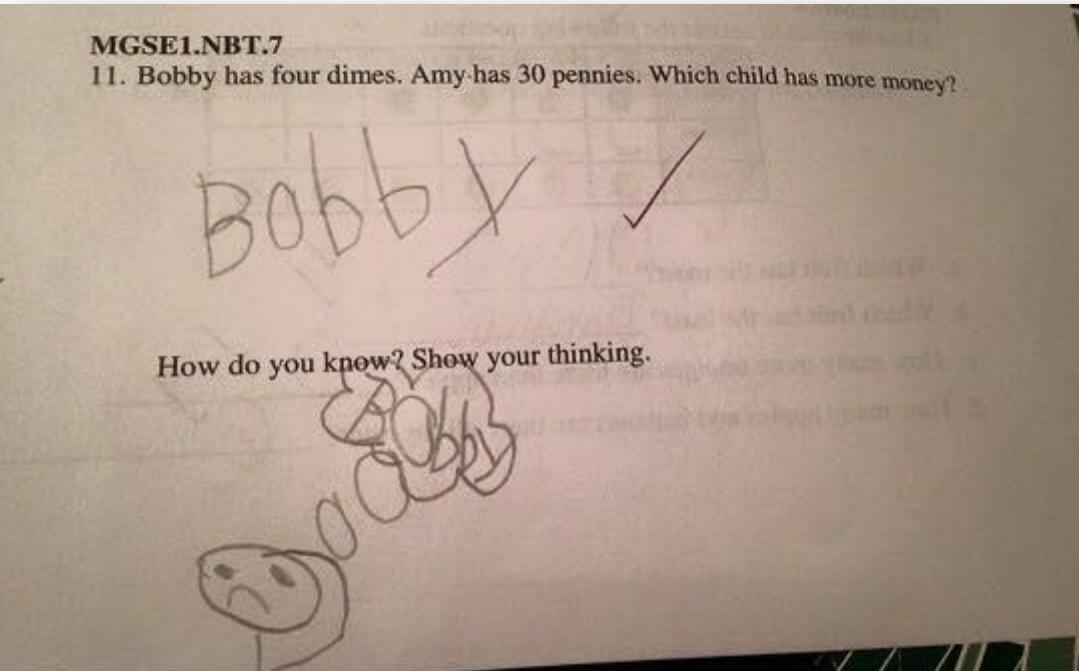
- É uma linguagem que valorize legibilidade de código por meio do uso de espaçamentos.
- Possui garbage-collector, ou seja, não precisamos nos preocupar com liberação de memória. Vocês conseguem me dizer a principal desvantagem do uso de garbage-collector?
- Suporta orientação a objetos e programação functional.
- Python 2.0 foi lançado em 2000 e foi descontinuada em 1 de janeiro de 2020.
- Python 3.0 foi lançado em 2008 e não é compatível com o Python 2.0. Por isso adotaremos aqui o Python 3.x.



O Conceito de Algoritmo



2. O Conceito de Algoritmo



2. O Conceito de Algoritmo

- Conhecimento declarativo:

- Corpo organizado de informações factuais (o que é).
- Ex.: **O bolo comum** é a massa de bolo básica para confeitar ou recheiar. É também um excelente acompanhamento para café ou chá.

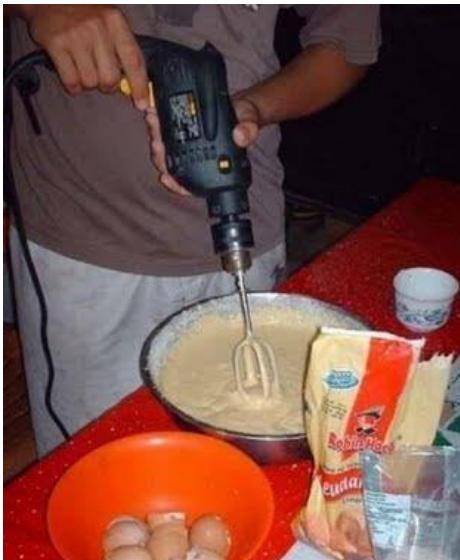


2. O Conceito de Algoritmo

- Conhecimento imperativo:

➤ É o que se manifesta da execução de uma tarefa (como fazer).

➤ Ex.: **Bolo comum:**



1. Bater o açúcar e a manteiga, com a essência de baunilha até branquear.
2. Acrescentar as gemas uma a uma, batendo sempre, até levantar bolhas.
3. Peneirar a farinha, a maizena e o fermento e ir acrescentando pouco a pouco, alternando com o leite, sem parar de bater.
4. Em separado, bater as claras em neve, com a pitada de sal.
5. Misturar as claras delicadamente à mistura
6. Assar em forma untada e polvilhada com farinha de trigo, em forno médio, por aproximadamente 40 minutos.



2. O Conceito de Algoritmo

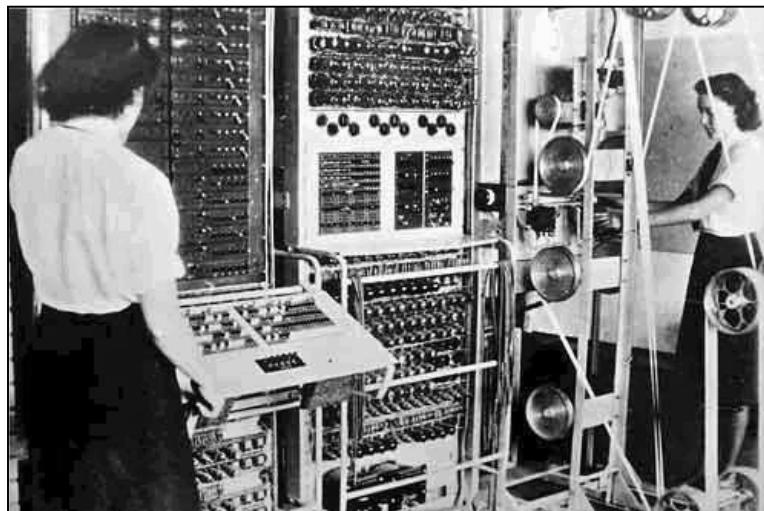


- **Alan Turing** (1912 - 1954), matemático britânico, descreve em 1936 uma máquina teórica, conhecida como **Máquina de Turing**, capaz de realizar qualquer tarefa computável, desde que execute adequadamente uma determinada seqüência de instruções.
- Esse seqüência de instruções é chamada de **algoritmo**.
- Em 1938, foi recrutado pelo departamento de análise criptográfica do governo.
- Consegiu decifrar o código da máquina de criptografia Enigma, que a Alemanha de Hitler usava para mandar mensagens militares cifradas durante a guerra.



2. O Conceito de Algoritmo

- Graças ao sistema de decodificação que ele criou, o Reino Unido passou a interceptar as mensagens e localizar os submarinos alemães, atacando-os e revertendo o avançar da guerra.
- Mas seu trabalho era secreto, e os feitos de Turing passaram sem aclamação na época.
- Colossus:



2. O Conceito de Algoritmo

- Como homossexual, no início dos anos 1950 foi publicamente humilhado.
- Perdeu o acesso de segurança aos laboratórios onde trabalhava porque, sob a mentalidade da Guerra Fria corrente, homossexuais eram uma brecha na segurança.
- Condenado a terapias à base de estrogênio, o que, de fato, equivalia a castração química e que teve o humilhante efeito secundário de lhe fazer crescer seios.
- Em 8 de junho de 1954, um criado de Turing encontrou-o morto em sua residência em Wilmslow, Cheshire.
- Um exame post-mortem estabeleceu que a causa da morte foi envenenamento por cianeto.

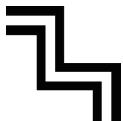
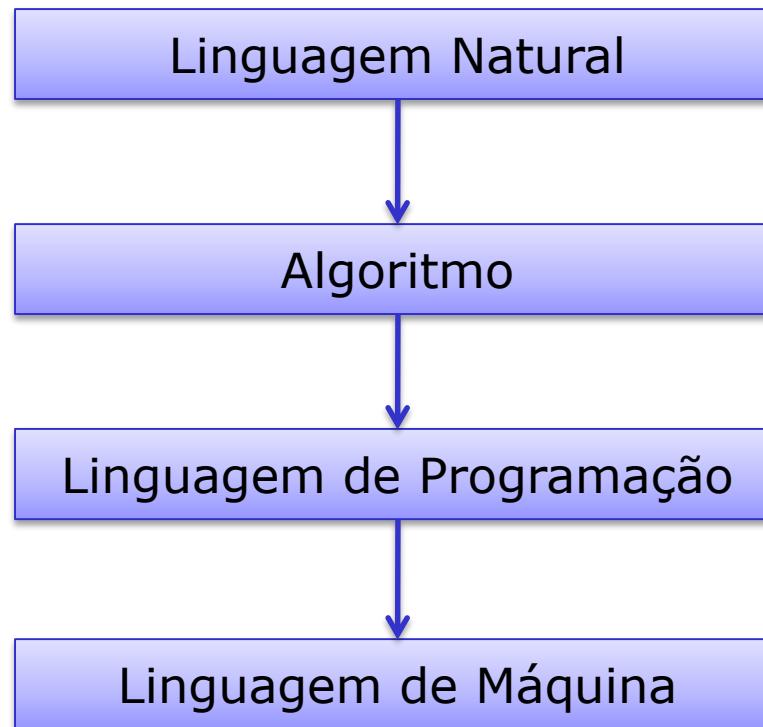


2. O Conceito de Algoritmo

- O conceito central da programação e da ciência da computação é o de **algoritmo**.
- Um algoritmo é a descrição de um padrão de comportamento, expresso em termos de um repertório bem definido e finito de **ações "primitivas"**, das quais damos por certo que elas podem ser executadas.
- Introduziremos de forma intuitiva a noção de algoritmo, motivando ao mesmo tempo as estruturas básicas de controle (**seqüência simples, alternativa e repetição**) como formas de raciocínio "naturais".



2. O Conceito de Algoritmo



3. Algoritmos Não-Computacionais

- Seqüência Simples

"traga a cesta com batatas do porão";
"traga a panela do armário";
"descasque as batatas";
"devolva a cesta ao porão";



3. Algoritmos Não-Computacionais

- Alternativas

"traga a cesta com batatas do porão";

"traga a panela do armário";

se "saia é clara" **então** "coloque avental";

"descasque as batatas";

"devolva a cesta ao porão";



3. Algoritmos Não-Computacionais

- Repetições

"traga a cesta com batatas do porão";

"traga a panela do armário";

se "saia é clara" então "coloque avental";

"descasque uma batata";

"descasque uma batata";

... (50 vezes)

"descaque uma batata";

"devolva a cesta ao porão"



3. Algoritmos Não-Computacionais

- Repetições

"traga a cesta com batatas do porão";

"traga a panela do armário";

se "saia é clara" então "coloque avental";

se "número de batatas é insuficiente" **então**

"descasque uma batata";

se "número de batatas é insuficiente" **então**

"descasque uma batata";

...(50 vezes)

se "número de batatas é insuficiente" **então**

"descasque uma batata";

"devolva a cesta ao porão";



3. Algoritmos Não-Computacionais

- Repetições

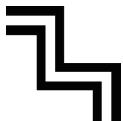
"traga a cesta com batatas do porão";

"traga a panela do armário";

se "saia é clara" então "coloque avental";

enquanto "número de batatas é insuficiente" **faça**
"descasque uma batata";

"devolva a cesta ao porão";



4. Algoritmos Computacionais

- Um algoritmo computacional é uma seqüência de instruções que manipula dados.
- **Instruções:** comandos que determinam a forma pela qual os dados devem ser tratados.
- **Dados:** informações recolhidas/fornecidas por diversos meios e que serão processadas pelo computador através das instruções.



4. Algoritmos Computacionais

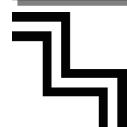
- **Identificadores:** nomes (rótulos) atribuídos às variáveis, funções e estruturas de dados que são utilizados em algoritmos.
- Regras para formação de identificadores:
 - o primeiro caractere deve ser, obrigatoriamente, uma letra ou um underscore;
 - do segundo caractere em diante, são permitidos números, letras e underscores;
 - não são permitidos espaços, caracteres acentuados e símbolos especiais na composição do nome de um identificador;
 - palavras reservadas não podem ser usadas;
 - há distinção entre maiúsculo e minúsculo.



4. Algoritmos Computacionais

- **Palavras reservadas no Python 3:**

and	except	lambda	with
as	finally	nonlocal	while
assert	false	None	yield
break	for	not	
class	from	or	
continue	global	pass	
def	if	raise	
del	import	return	
elif	in	True	
else	is	try	



4. Algoritmos Computacionais

- Identificadores

- Exemplos válidos:

DataNascimento

DATA_DE_NASCIMENTO

IDADE

Nota1

TRABALHO2

PESO

- Exemplos inválidos:

3CD

Meu Nome

Idade&

DA*TA

Sílaba



4. Algoritmos Computacionais

- **Variável:** é um espaço reservado na memória do computador para armazenar um determinado dado.
- Devem receber identificadores para poderem ser referenciadas e modificadas quando necessário.
- Embora uma variável possa assumir diferentes valores, ela só pode armazenar um valor a cada instante.
- Ela passa a existir no momento em que um valor é atribuído a ela (em contraste com C, não há declaração de variáveis em Python e a tipificação é fraca).



4. Algoritmos Computacionais

- Variável:

Memória	
27	Idade
71.5	Massa
BOBBY	Nome
.	
.	



5. Algoritmos Computacionais: Operadores

- Operador de atribuição:

Operador	Ação
=	Atribuição

Idade = 30

a = b = 1.5



5. Algoritmos Computacionais: Operadores

- Operadores aritméticos:

Operador	Ação
-	Subtração, também menos unário
+	Adição
*	Multiplicação
/	Divisão
%	Módulo da divisão (resto)
--	Decremento – PYTHON NÃO TEM
++	Incremento – PYTHON NÃO TEM



5. Algoritmos Computacionais: Operadores

- Operadores aritméticos:

Operador	Ação
-	Subtração
+	Adição
*	Multiplicação
/	Divisão

$$\text{delta} = b*b - 4*a*c$$



5. Algoritmos Computacionais: Operadores

- Menos unário:

Operador	Ação
-	Menos unário

`num1 = -num`



5. Algoritmos Computacionais: Operadores

- Operadores aritméticos:

Operador	Ação
%	Módulo da divisão (resto)

`num = 17%5`



5. Algoritmos Computacionais: Operadores

- Precedência dos operadores aritméticos:

Operador	Ação	Precedência
()	Parênteses	Mais alta
**	Potência	.
+X, -X, ~X	Positivo unário, negativo unário e not bitwise	.
*, /, //, %	Multiplicação, divisão, divisão inteira e módulo	.
+, -	Adição e subtração	.
==, !=, >, >=, <, <=, is, is not, in, not in	Comparações, identidade e operadores de membresia	.
not	Negação lógica	.
and	“E” lógico	.
or	“OU” lógico	Mais baixa



5. Algoritmos Computacionais: Operadores

- Precedência dos operadores aritméticos:

Operadores do mesmo nível de precedência são avaliados da esquerda para a direita.



5. Algoritmos Computacionais: Operadores

- Operadores aritméticos de atribuição:

Operador	Ação
$a += b$	$a = a + b$
$a -= b$	$a = a - b$
$a *= b$	$a = a * b$
$a /= b$	$a = a / b$
$a \% = b$	$a = a \% b$

$a += 2$
equivale a
 $a = a + 2$



6. Operadores: Exemplos

- Exemplos:

$$2x^2 - \frac{3x^{(x+1)}}{2} + \frac{\sqrt{x+1}}{x}$$



6. Operadores: Exemplos

- Exemplos:

$$2x^2 - \frac{3x^{(x+1)}}{2} + \frac{\sqrt{x+1}}{x}$$

`2*(x**2) - 3*(x**(x+1))/2 + ((x+1)**0.5)/x`



6. Operadores: Exemplos

- Exemplos:

$$2h - \left(\frac{45}{3x} - 4h(3-h) \right)^{22k}$$



6. Operadores: Exemplos

- Exemplos:

$$2h - \left(\frac{45}{3x} - 4h(3-h) \right)^{22k}$$

`2*h - (45/(3*x) - 4*h*(3-h))**(22*k)`



6. Operadores: Exemplos

- Exemplos:

$$a+b+(34 + (9^{**}e))/(u - (89^{**}0.5))$$


6. Operadores: Exemplos

- Exemplos:

a+b+(34 + (9**e))/(u - (89**0.5))

$$a + b + \frac{34 + 9^e}{u - \sqrt{89}}$$



6. Operadores: Exemplos

- Exemplos:

$$((a+x^{**}2+w) - 3*a)/2$$



6. Operadores: Exemplos

- Exemplos:

$$((a+x^{**2+w}) - 3*a)/2$$

$$\frac{(a + x)^{2+w} - 3a}{2}$$



7. Entrada e Saída de Dados

- Saída de dados (via monitor):

```
print(f'Sua idade é: {idade} anos.')
print(f'Sua altura é: {altura} metros.')
print(f'A letra é: {letra}.', letra)
print(f'I: {idade}, A: {altura}, L: {letra}.')
```

Não há diferença entre aspas simples e duplas.



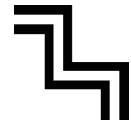
7. Entrada e Saída de Dados

- Entrada de dados (via teclado):

`idade = input()`



Algoritmos Sequenciais



8. Algoritmos Sequenciais

- **Construindo os primeiros algoritmos**

Problema: Escreva um algoritmo que leia dois números, calcule a média aritmética entre esses dois valores e mostre o resultado na tela do computador.

- Basicamente a construção de um algoritmo se resume às seguintes etapas:
 1. Entendimento do problema;
 2. Definição dos dados, e respectivos tipos, que serão necessários para resolvê-lo (variáveis de entradas, auxiliares e de saída);
 3. Obtenção destes dados (entradas);
 4. Elaboração do processamento em si;
 5. Exibição dos resultados (saídas).



8. Algoritmos Sequenciais

- **Construindo os primeiros algoritmos**

Problema: Escreva um algoritmo que leia dois números, calcule a média aritmética entre esses dois valores e mostre o resultado na tela do computador.

- **Etapa 1** (entendimento do problema):

A média aritmética de dois valores é calculada por $(a+b)/2$.



8. Algoritmos Sequenciais

- **Construindo os primeiros algoritmos**

Problema: Escreva um algoritmo que leia dois números, calcule a média aritmética entre esses dois valores e mostre o resultado na tela do computador.

- **Etapa 2** (definição dos dados e seus tipos):

Os dados necessários serão os dois valores, que colocaremos em duas variáveis A e B, e uma terceira variável, que chamaremos MEDIA, onde o resultado será armazenado.



8. Algoritmos Sequenciais

- **Construindo os primeiros algoritmos**

Problema: Escreva um algoritmo que leia dois números, calcule a média aritmética entre esses dois valores e mostre o resultado na tela do computador.

- **Etapa 3** (obtenção dos dados):

A obtenção dos dados neste algoritmo é simples. Basta pedir ao usuário que digite os valores. Os valores serão capturados por meio da função input().



8. Algoritmos Sequenciais

- **Construindo os primeiros algoritmos**

Problema: Escreva um algoritmo que leia dois números, calcule a média aritmética entre esses dois valores e mostre o resultado na tela do computador.

- **Etapa 4** (processamento):

O processamento aqui é o cálculo da média a partir dos valores de A e B. O resultado do cálculo será armazenado na variável MEDIA.



8. Algoritmos Sequenciais

- **Construindo os primeiros algoritmos**

Problema: Escreva um algoritmo que leia dois números, calcule a média aritmética entre esses dois valores e mostre o resultado na tela do computador.

- **Etapa 5** (exibição do resultado):

Basta exibir o conteúdo da variável MEDIA por meio da função print().

Vamos ver esse código no nosso notebook!



Algoritmos com Alternativas



9. Algoritmos com Alternativas

- Operadores lógicos:

Operador	Ação
not	NAO (Negação)
and	E (Conjunção)
or	OU (Disjunção inclusiva)

not 0 and 0 or 0
not (0 and 0) or 0

ATENÇÃO: O operador “not” tem mais alta precedência que qualquer operador aritmético. Já os operadores “and” e “or” têm mais baixa precedência.



9. Algoritmos com Alternativas

- Operadores lógicos:

NOT

A	X
0	1
1	0

not A

AND

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

A and B

OR

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

A or B



9. Algoritmos com Alternativas

- Operadores relacionais:

Operador	Ação
>	Maior que
>=	Maior ou igual a
<	Menor que
<=	Menor ou igual
==	Igual
!=	Diferente

$A > B+8$

ATENÇÃO: Esse operadores são menores em precedência que qualquer operador aritmético.



9. Algoritmos com Alternativas

- **Precedência entre operadores lógicos e relacionais:**

Operador	Precedência
not	Mais alta
> >= < <=	.
== !=	.
and	.
or	Mais baixa

10 > 5 and not (10 < 9) or 3 <= 4



9. Algoritmos com Alternativas

- Na vida real tomamos decisões a partir da análise de algumas condições.

✓ Exemplo: Se eu tiver pelo menos R\$ 50,00 (na conta corrente ou poupança), então irei ao cinema hoje à noite.

Trata-se de uma expressão lógica, uma vez que a pergunta: “Tenho R\$ 50,00 sobrando?” Pode ser respondida com um “Sim” ou com um “Não”.

- Analogamente, em linguagens de programação um determinado bloco básico de comandos **será executado ou não**, dependendo da avaliação de expressões lógico-aritmético-relacionais. A isso chamados de **algoritmo com alternativa**.



9. Algoritmos com Alternativas

- **Alternativa simples (exemplo):**

Exemplo 1: Escreva um algoritmo para calcular a área de um círculo, fornecido o valor do raio, **que deve ser positivo**. Em seguida o algoritmo deve imprimir o valor da área na tela do computador.



9. Algoritmos com Alternativas

- **Alternativas simples:**

- Forma geral:

```
if <expressão>:
```

```
<comandos>
```



9. Algoritmos com Alternativas

- Alternativa simples (exemplo):**

Exemplo 2: Escreva um algoritmo que lê o valor da mercadoria, calcula e mostra o valor final a ser pago incluindo, se for o caso, a taxa de embrulho para presente que é de R\$ 1,50.



9. Algoritmos com Alternativas

- **Alternativa composta:**

- Forma geral:

```
if <expressão>:  
    |  
    |     <comandos>  
else:  
    |  
    |     <comandos>
```

- Como podemos refazer o Exemplo 1?



9. Algoritmos com Alternativas

- **Alternativas aninhadas:**

```
if <expressão>:  
    <comandos>  
    if <expressão>:  
        <comandos>  
    else:  
        <comandos>  
  
<comandos>  
  
else:  
    <comandos>  
    if <expressão>:  
        <comandos>  
    else:  
        <comandos>  
  
<comandos>
```



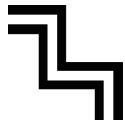
9. Algoritmos com Alternativas

- **Alternativas de múltiplas escolhas (exemplo):**

Exemplo 3: Escreva um algoritmo que solicita ao usuário dois operandos e um código de operação (1 - Soma, 2 - Subtração, 3 - Divisão ou 4 - Multiplicação) e realiza a operação correspondente sobre os operandos fornecidos pelo usuário.



Algoritmos com Repetições



10. Algoritmos com Repetições

- **Repetições com teste no início:** enquanto...faca.

➤ Forma geral:

```
enquanto <expressão> faça  
    <lista de comandos>  
fim enquanto
```



10. Algoritmos com Repetições

- **Repetições com teste no início (exemplo):**

Exemplo 1: Escreva um algoritmo que solicita ao usuário um valor inteiro positivo N e imprime na tela do computador todos os números inteiros de 0 a N.



10. Algoritmos com Repetições

- **Repetições com teste no início:** while.

➤ Forma geral:

while <expressão>:

| <comandos>



10. Algoritmos com Repetições

- Vamos ficar um pouco mais no nosso Notebook?**

