

Fundamentos de Algoritmos

Aula 01

Apresentação da Disciplina e Introdução a Lógica Sequencial

Profa. Dra. Gabriela Biondi

Ciência da Computação
Ciência de Dados e Inteligência Artificial

Centro Universitário FEI

1º Semestre de 2025

Apresentação

Docentes

Gabriela Biondi - gbiondi@fei.edu.br

- Mini-curriculum:
 - Bacharel em Ciência da Computação
 - Centro Universitário da FEI - 2011
 - Mestrado em Ciência da Computação
 - Universidade Federal do ABC - 2014
 - Doutorado em Engenharia da Informação
 - Universidade Federal do ABC - 2020

Apresentação

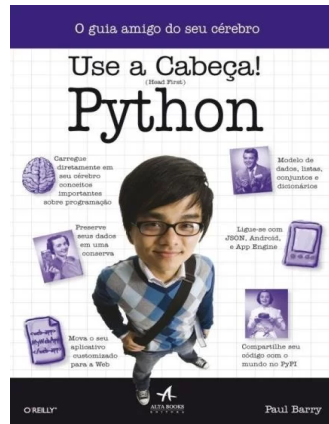
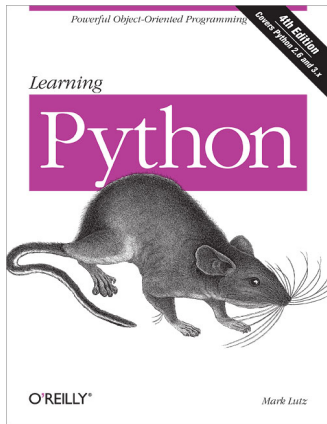
Docentes

Rafael Gomes Alves - `ralves@fei.edu.br`

- Mini-curriculo:
 - Bacharel em Engenharia Mecânica
 - Centro Universitário da FEI - 2017
 - Mestrado em Engenharia Mecânica
 - Universidade Federal do ABC - 2020
 - Doutorado em Engenharia Elétrica
 - Universidade Federal do ABC - 2021 (em andamento)

Apresentação

Bibliografia Básica



Apresentação

Bibliografia Complementar



Apresentação

Moodle

- Sobre a Disciplina
- Cronograma
- Teoria e Laboratório
- Projeto
- Materiais complementares
- Monitoria

Introdução

Conceitos importantes

- Ciclo de Processamento de Dados
- Dado x Informação
- Algoritmo

Variáveis

Objetivo

- Serve para escrever e ler dados da memória do computador
- São usadas para criar uma região de memória para armazenar dados:

```
1 valor = 25
```

- valor é uma variável que vai armazenar o número 25

Variáveis

Objetivo

- A variável modela as unidades de memória
- Possui um identificador
- Associa um endereço físico de memória
- Tem um tamanho em bytes

Variáveis

Regras de nomes de variáveis em Python

- Devem começar, obrigatoriamente, com uma letra ou _
- Porém, o restante dos caracteres podem conter números
- Em programação, não se utiliza acentos nos nomes das variáveis porém o Python 3, permite esta utilização
- O Python diferencia letras maiúsculas de letras minúsculas
- Não deve haver espaços

Variáveis

Regras de nomes de variáveis em Python

nome	válido	explicação
a1	sim ✓	começa com letra
velocidade	sim ✓	letras
velocidade90	sim ✓	letras e números
salario_medio	sim ✓	letras e _
salario medio	não ✗	não pode conter espaços
_b	sim ✓	_ é aceito no início do nome
1a	não ✗	não pode iniciar com números

Variáveis

Regras de nomes de variáveis em Python

- Escolher nomes de variáveis com base no que a variável vai armazenar
- Exemplo:

```
1 nome = "Fulano da Silva"
2 universidade = "FEI"
3 idade = 18
4 ano = 2025
```

Variáveis

Regras de nomes de variáveis em Python

- Algumas palavras são reservadas em Python e não podem ser nomes de variáveis. São elas:

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

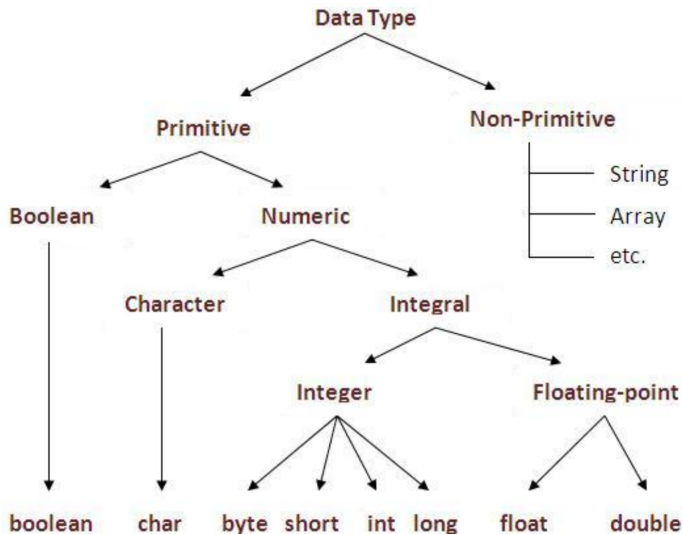
Variáveis

Tipos de Dados

- Variáveis armazenam dados!
- Dados são as entidades fundamentais que um programa manipula!
- Os dados podem ser de diferentes tipos
- Números:
 - ▶ Inteiro (`int`): 1, 2, 3, 0, 10
 - ▶ Real (`float`): 1.3, -3.63, 7.2, 16.42
 - ▶ Complexo (`complex`): $6 + 3j$, $-2 + 4j$
- Texto (`string`): "Olá", "Isto é uma string"
- Tipo Lógico (`bool`): True, False

Variáveis

Tipo de Dados



Variáveis

Tipos de Dados

Função `type()`

- Utilizada para descobrir qual é o tipo de um determinado dado

Código em python

```
print(type(3.6))  
print(type(2))  
  
print(type("ola"))  
print(type("2.8"))  
  
print(type(2+6j))
```

Saída esperada

```
<class 'float'>  
<class 'int'>  
<class 'str'>  
<class 'str'>  
<class 'complex'>
```


Variáveis

Tipos de Dados

- `int()`, `float()` e `str()`

```
a = "3"  
b = int(a)  
print(type(b))
```

<class 'int'>

```
a = 3  
b = str(a)  
print(type(b))
```

<class 'str'>

Variáveis

Atribuição de Valores

- Para armazenar valores nas variáveis, utilizamos o igual (=)
- Esta operação é chamada de atribuição, pois um valor é atribuído a uma variável

Exemplo de atribuição

```
a = 5  
b = 6  
print(a + b)
```

Saída esperada

```
11
```

Saída de Dados

Função `print()`

- A função `print()` é utilizada para imprimir dados no terminal
- Podemos combinar texto com o valor de alguma variável
- Para isso, utilizamos a vírgula `(,)` entre um dado e outro

Código

```
a = 3  
print("a vale", a)
```

Saída esperada

```
a vale 3
```

Saída de Dados

Função `print()` - composição

- Combinar várias strings com o valor de variáveis nem sempre é prático, especialmente se estivermos utilizando a vírgula para combiná-los

Código

```
a = "ola"  
dia = 16  
mes = 8  
print(a, "hoje eh dia ", dia, " do mes ", mes)
```

Saída esperada

```
ola hoje eh dia 16 do mes 8
```

não é muito prático

Saída de Dados

Função `print()` - composição

- Outra forma de combinar string com variável é através da composição

Código

```
anos = 30  
print("Joao tem %d anos" % anos)
```

Saída esperada

Joao tem 30 anos

- O símbolo `%d` é chamado de marcador
- O marcador indica que naquela posição estaremos colocando o valor da variável `anos`, que neste exemplo, deve ser um número inteiro

Saída de Dados

Função `print()` - composição marcadores

marcador	tipo
<code>%d</code>	Número inteiro
<code>%f</code>	Números decimais (float)
<code>%s</code>	strings

Saída de Dados

Função `print()` - composição marcadores

- Exemplos:

Código

```
a = "ola"  
dia = 16  
mes = 8
```

```
print("%s hoje eh dia %d do mes %d" % (a,dia,mes))
```

Saída esperada

```
ola hoje eh dia 16 do mes 8
```

um pouco mais prático

Código

```
pi = 3.141592  
print("0 pi vale %f" % pi)
```

Saída esperada

```
0 pi vale 3.141592
```

Saída de Dados

Função `print()` - composição marcadores

- Limitando a quantidade de casas decimais na impressão:
 - ▶ Basta incluir o `.x` entre o `%` e o `f`
 - ▶ em que `x` é o número desejado de casas decimais
- Exemplo:

Código

```
pi = 3.141592  
print("0 pi vale %.2f" % pi)
```

Saída esperada

0 pi vale 3.14

Saída de Dados

Função `print()` - `format()`

- O método `format()`:
 - ▶ Formata a string dada de acordo com o desejado para a saída de dados
- Exemplo:

Código

```
nome = "Fulano"  
peso = 78.51  
print("O {0} pesa {1:.1f} Kg".format(nome,peso))
```

Saída esperada

O Fulano pesa 78.5 Kg

Entrada de Dados

Função `input()`

- Objetivo
 - ▶ Ler um dado digitado pelo usuário
 - ▶ Fazer algum processamento com esse dado
- Para isso, utilizaremos o comando `input()`
- Observações importantes
 - ▶ A entrada de dados é feita pela função `input()`
 - ▶ `input()` aceita como parâmetro uma mensagem a ser exibida
 - ▶ O valor recebido pelo `input()` deve ser atribuído a uma variável
- Exemplo:

```
nome = input("Digite o seu nome: ")  
print("nome digitado: " % nome)
```

Entrada de Dados

Função `input()`

- Observações importantes

- ▶ Todo valor recebido pela função `input()` tem sempre o tipo `string`
- ▶ Se a ideia é utilizar o valor recebido em alguma conta ou cálculo então o valor deve ser convertido para algum tipo numérico.

- Exemplo:

```
n1 = int(input("Digite um numero: "))  
n2 = int(input("Digite outro numero: "))  
soma = n1 + n2  
print("Soma dos numeros: %d" % soma)
```

Entrada de Dados

Função `input()`

- Exemplo:

- ▶ O valor digitado para `pi` será recebido como string
- ▶ Porém, antes de ser atribuído a variável `pi`, ele será convertido em float:

Código

```
pi = float(input("Digite o valor de pi: "))  
print("O valor digitado eh %.1f " % pi)
```

Saída esperada

```
Digite o valor de pi: 3.14159  
O valor digitado eh 3.1
```

Operadores Aritméticos

Operador	Significado
+	Soma
-	Sub
*	Multiplicação
/	Divisão
//	Divisão com resultado inteiro
%	Módulo: retorna o resto da divisão
**	Potência: $x^{**}y$ (x elevado a y)

Operadores Aritméticos - Exemplos

Código

```
a = 4
b = 5
c = a + b
print("Soma: ")
print(c)
c = a - b
print("Subtracao: ")
print(c)
c = a * b
print("Multiplicacao: ")
print(c)
c = b/a
print("Divisao: ")
print(c)
```

Saída esperada

```
Soma:
9
Subtracao:
-1
Multiplicacao:
20
Divisao:
1.25
```

Operadores Aritméticos - Exemplos

Código

```
a = 4
b = 5
c = b // a
print("Divisao (resultado parte inteira): ")
print(c)
c = a ** b
print("Potencia: ")
print(c)
c = b % a
print("Modulo: ")
print(c)
```

Saída esperada

```
Divisao (resultado parte inteira):
1
Potencia:
1024
Modulo:
1
```

Funções Matemáticas

- É necessário importar o módulo de matemática

```
from math import *
```

função	descrição
abs(x)	valor absoluto de x
sqrt(x)	raiz quadrada de x
log(x)	retorna o logaritmo natural de x
log10(x)	retorna o logaritmo base-10 de x
sin(x)	retorna o seno de x radianos
cos(x)	retorna o cosseno de x radianos
exp(x)	retorna e^x
round(x,n)	número x arredondado para n dígitos

Funções Matemáticas

- Exemplo:

Código

```
from math import *  
  
print(sqrt(9))  
print(log(3))
```

Saída esperada

```
3.0  
1.0986122886681098
```

Fundamentos de Algoritmos

Aula 01

Apresentação da Disciplina e Introdução a Lógica Sequencial

Profa. Dra. Gabriela Biondi

Ciência da Computação
Ciência de Dados e Inteligência Artificial

Centro Universitário FEI

1º Semestre de 2025