

Estrutura de Dados e Programação Orientada a Objetos

Prof^a. Dr^a. Alana Moraes
alanamm.prof@gmail.com

Sobre a disciplina



Ementa

- Linguagem Python
- Parte 1:
 - Introdução às Estrutura de Dados
 - Pilhas
 - Filas
 - Listas Sequenciais
 - Listas Duplamente Encadeadas
 - Listas Circulares
 - Árvores
 - Grafos

Avaliação

- Nota 1:
 - Prova
- Nota 2:
 - Prova Prática
- Nota 3:
 - Projeto
- Reposição
- Final

Bibliografia

- SZWARCFITER, Jayme Luiz; MARKENZON, Lilian. Estrutura de dados e seus algoritmos. 3. ed. Rio de Janeiro, RJ: LTC, 2013.
- GOODRICH, Michael T.; TAMASSIA, Roberto. Estruturas de dados e algoritmos em java. 2. ed. Porto Alegre, RS: Bookman, 2002.
- ZIVIANI, Nivio. Projeto de algoritmos: com implementação em pascal e C. 2. reimp. São Paulo, SP: Cengage Learning, 2013.

Bibliografia

- CELES, Waldemar; CERQUEIRA, Renato; RANGEL, José Lucas. Introdução a estruturas de dados: com técnicas de programação em C. Rio de Janeiro: Elsevier, 2004.
- PREISS, Bruno R.. Estrutura de dados e algoritmos: padrões de projetos orientados a objetos com java. Rio de Janeiro, RJ: Campus, 2001.
- PEREIRA, Silvio do Lago. Estruturas de dados fundamentais: conceitos e aplicações. 12. ed. rev. e atual. São Paulo, SP: Erica, 2008.

Bibliografia

- SOUZA, Marco A. Furlan ... et al. Algoritmos e lógica de programação: um texto introdutório para engenharia. São Paulo, SP: Cengage Learning, 2011.
- WIRTH, Niklaus. Algoritmos e estruturas de dados. Rio de Janeiro, RJ: LTC, 2012.

Como são seus conhecimentos em
algoritmos?

Dúvidas?



alanamm.prof@gmail.com

Revisão de Python

Prof^a. Dr^a. Alana Moraes
alanamm.prof@gmail.com

Recapitulando ...

Baixo Nível		Alto Nível
Linguagem de Máquina	Linguagem Assembly (<i>Mnemônica</i>)	Linguagem de Alto Nível
0010 0001 1110	LOAD R1, val1	val2 = val1+val2
0010 0010 1111	LOAD R2, val2	
0001 0001 0010	ADD R1, R2	
0011 0001 1111	STORE R1, val2	
(1)	(2)	(3)

Você já programou com a linguagem Python?



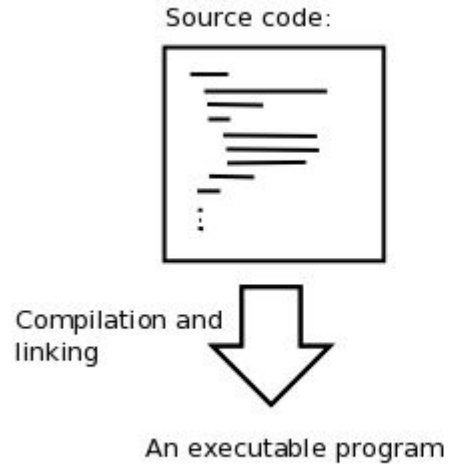
Python

- Linguagem de altíssimo nível;
- Linguagem interpretada;
- Linguagem de *script*;
- Linguagem imperativa;
- Linguagem orientada a objetos;
- Linguagem funcional;

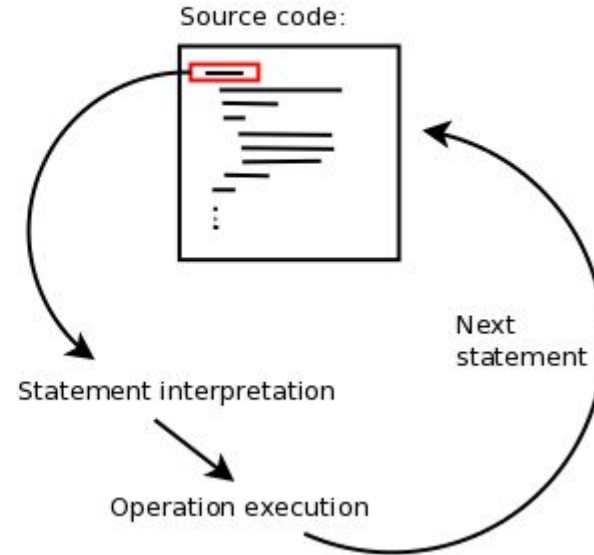


Recapitulando ...

Compilation

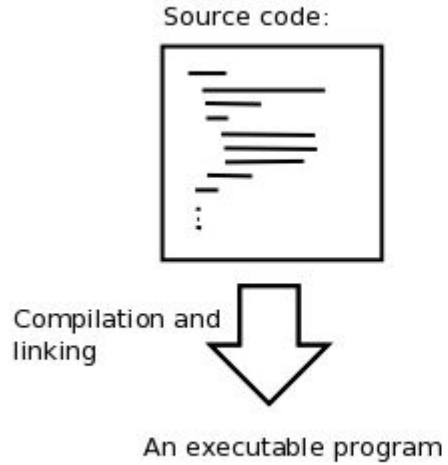


Interpretation

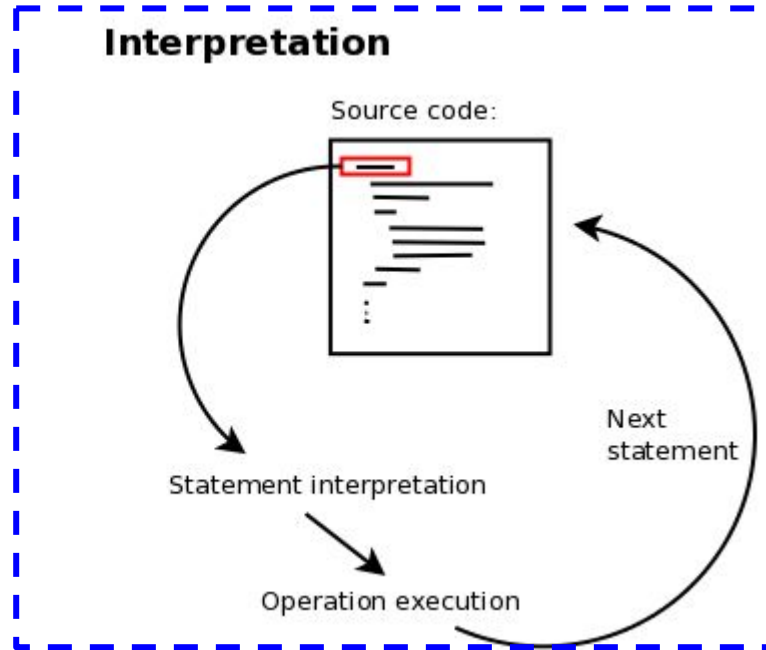


Recapitulando ...

Compilation



Interpretation



Python

- Possui um modelo de desenvolvimento aberto ;
- Linguagem dinamicamente tipada;
- A indentação do código é determinante em códigos em Python.

Dependências para usar Python nesta disciplina

- Interpretador Python = Python 3.7;
- IDE de desenvolvimento (opcional) = Pycharm Community;



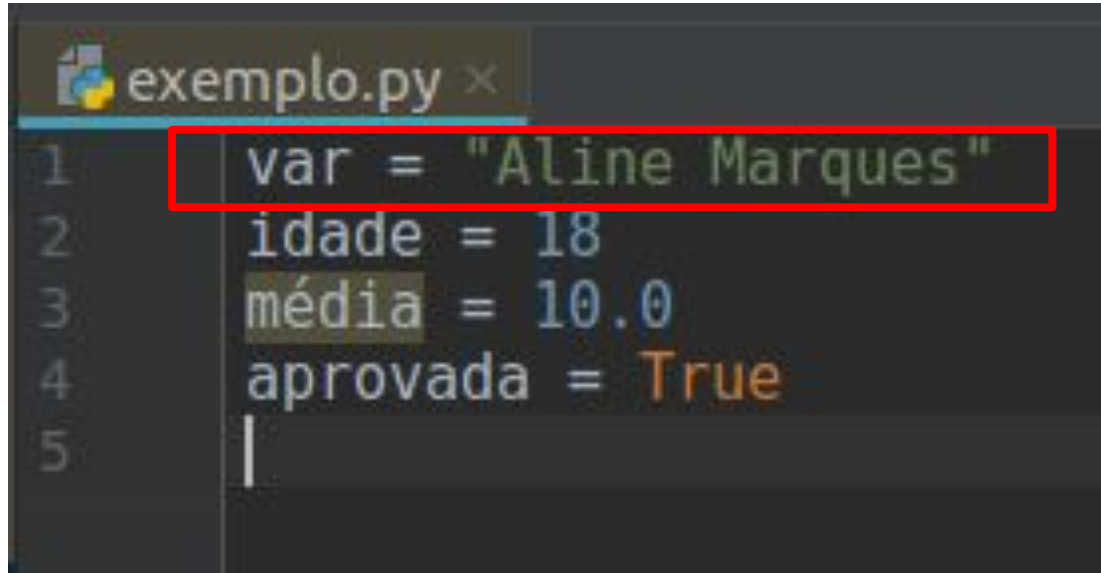
Sintaxe básica do Python

- Variável;
- Valor nulo;
- Saídas;
- Entradas;
- Funções
- Indentação;
- Blocos;
- Comentários;
- Bibliotecas.

Palavras reservadas

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

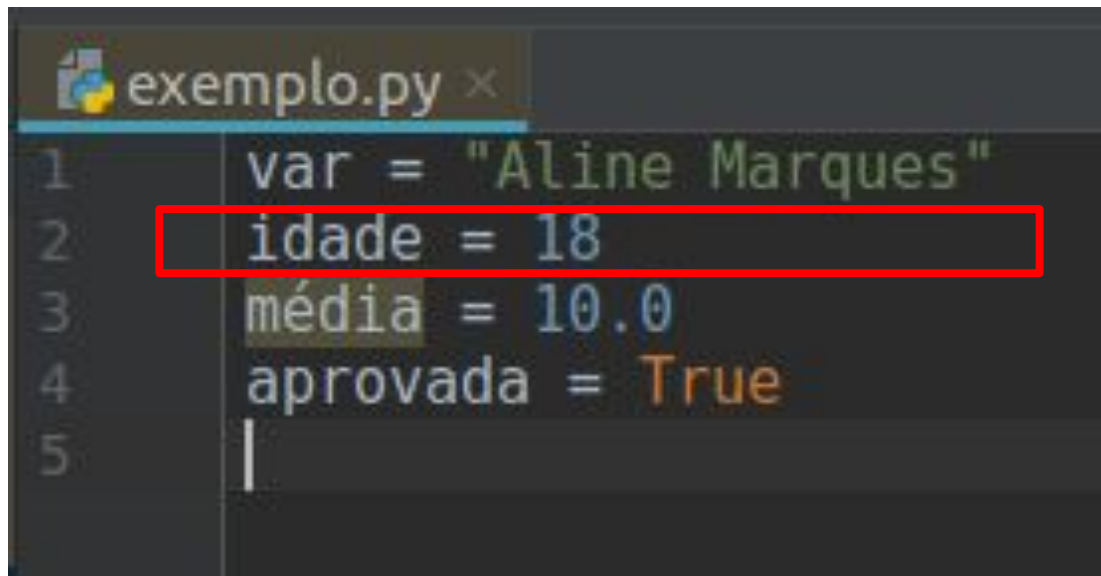
Variáveis - String



A screenshot of a code editor window titled 'exemplo.py'. The editor contains five lines of Python code. The first line, 'var = "Aline Marques"', is highlighted with a red rectangular box. The second line is 'idade = 18', the third is 'média = 10.0', and the fourth is 'aprovada = True'. The fifth line is empty. The code is written in a dark-themed editor with syntax highlighting: strings are green, integers are blue, floats are blue, and booleans are orange.

```
1 var = "Aline Marques"  
2 idade = 18  
3 média = 10.0  
4 aprovada = True  
5
```

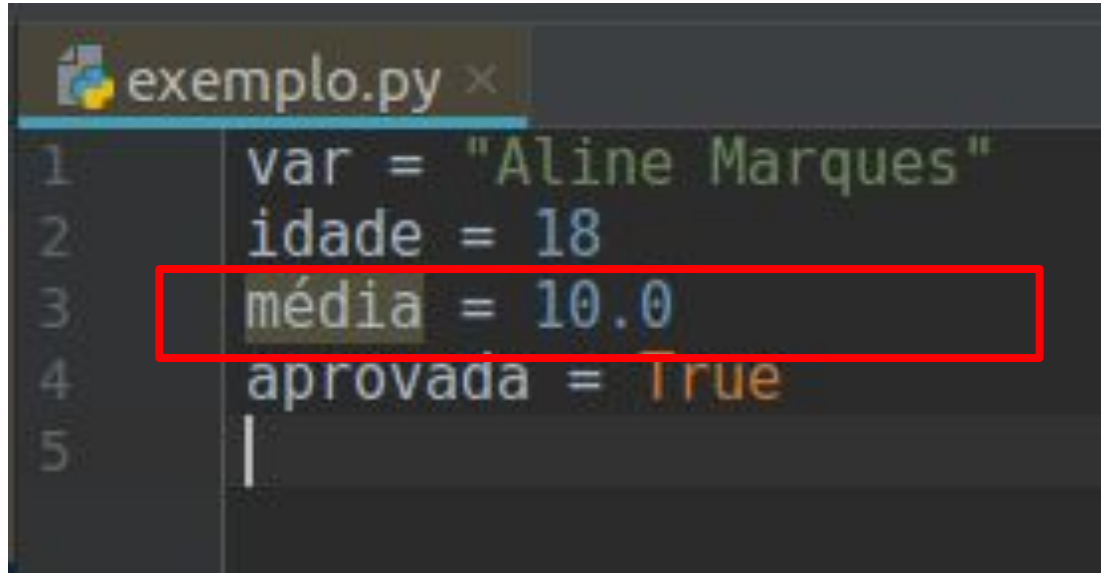
Variáveis - int



A screenshot of a code editor window titled 'exemplo.py'. The editor contains five lines of Python code. The second line, 'idade = 18', is highlighted with a red rectangular box. The code is as follows:

```
1 var = "Aline Marques"
2 idade = 18
3 média = 10.0
4 aprovada = True
5 |
```

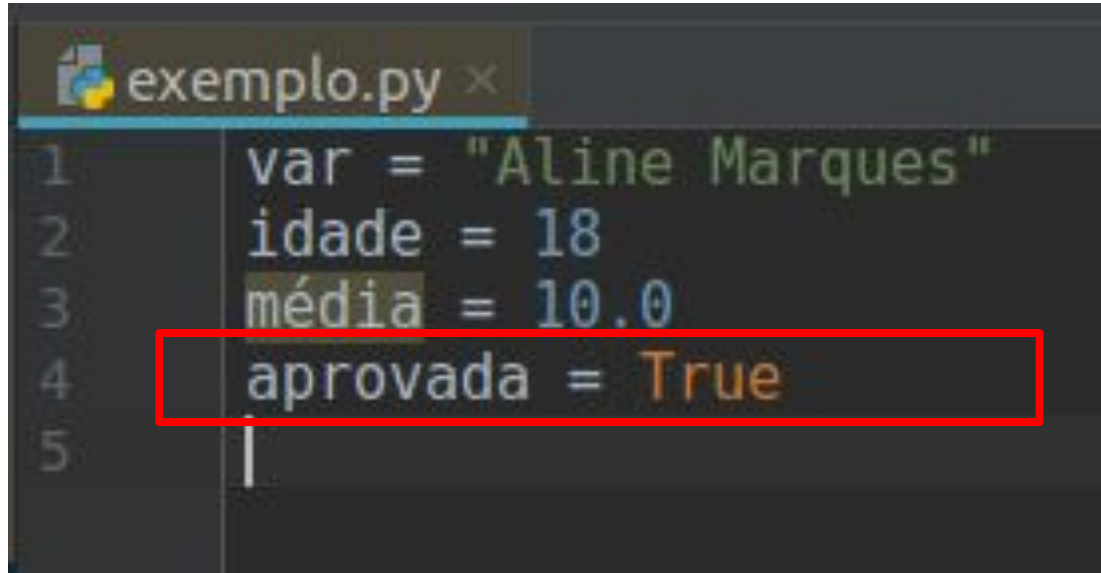
Variáveis - float



```
exemplo.py x
1 var = "Aline Marques"
2 idade = 18
3 média = 10.0
4 aprovada = True
5 |
```

The image shows a code editor window titled 'exemplo.py'. It contains five lines of Python code. The third line, 'média = 10.0', is highlighted with a red rectangular box. The variable 'média' is a float, and the value '10.0' is also a float. The other lines show string, integer, and boolean assignments.

Variáveis - bool



```
exemplo.py x
1 var = "Aline Marques"
2 idade = 18
3 média = 10.0
4 aprovada = True
5 |
```

Valor nulo

```
5  
6  
7 nulo = None  
8
```


Como descobrir o tipo da variável ?



Como descobrir o tipo da variável ?

```
valor = 10.21  
print(type(valor))
```

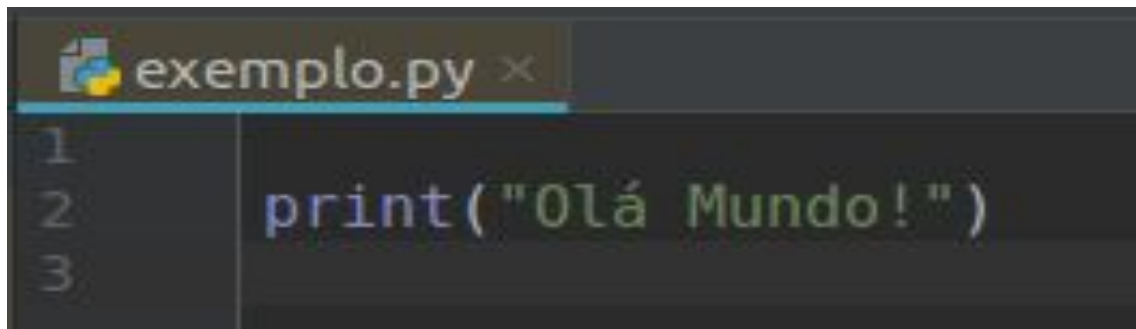
Variáveis

Class	Description
bool	Boolean value
int	integer (arbitrary magnitude)
float	floating-point number
list	mutable sequence of objects
tuple	immutable sequence of objects
str	character string
set	unordered set of distinct objects
frozenset	immutable form of set class
dict	associative mapping (aka dictionary)

Uma variável pode mudar de tipo?



Saída

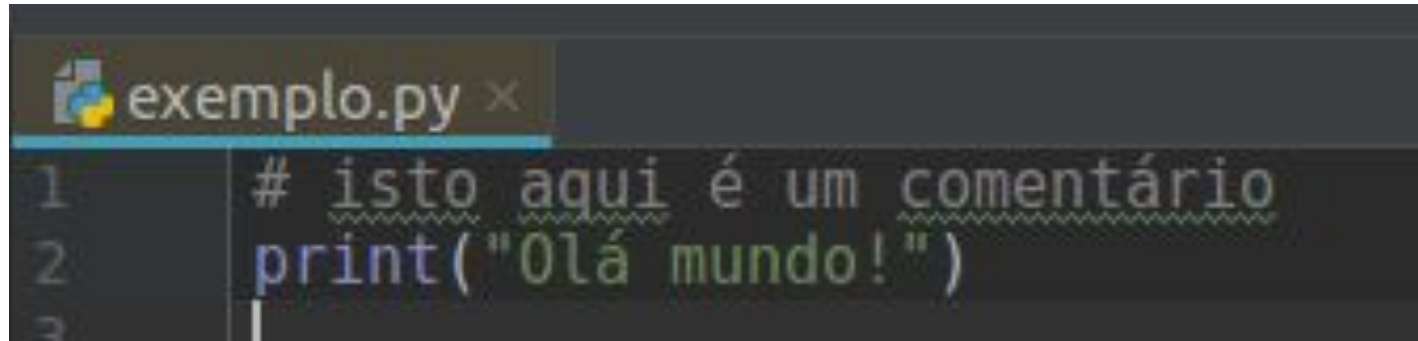


```
exemplo.py ×  
1  
2 print("Olá Mundo!")  
3
```

Saída de uma variável

```
10  
11     print(var)  
12     print("Seu nome é "+var)  
13
```

Comentário



A screenshot of a code editor window titled "exemplo.py". The editor shows two lines of Python code. Line 1 is a comment: `# isto aqui é um comentário`. Line 2 is a print statement: `print("Olá mundo!")`. The code is syntax-highlighted: the hash symbol is grey, the comment text is light blue, the `print` function is blue, and the string is green. The editor has a dark background and a light blue line number margin on the left.

```
1 # isto aqui é um comentário
2 print("Olá mundo!")
3
```

Vamos exercitar ...



Exercício A

Crie um código que armazene seu nome completo numa variável chamada nome e imprima ela na tela

Como atribuir uma entrada a uma variável?



Entrada

```
variável_digitada = input("digite seu nome")
```

Exercício B

Crie um código que receba **seu nome** e **sua cidade**.
Esse código gera como saída a seguinte mensagem:

olá “Fulano”, você mora em “João Pessoa”.

Exercício C

Calcule a média aritmética de um aluno IFPB com as notas das duas avaliações

Entrada:

nota 1= 9

nota 2= 7

Saída:

média = 8.0

Exercício D

Faça um programa que converta metros para centímetros.

Exercício E

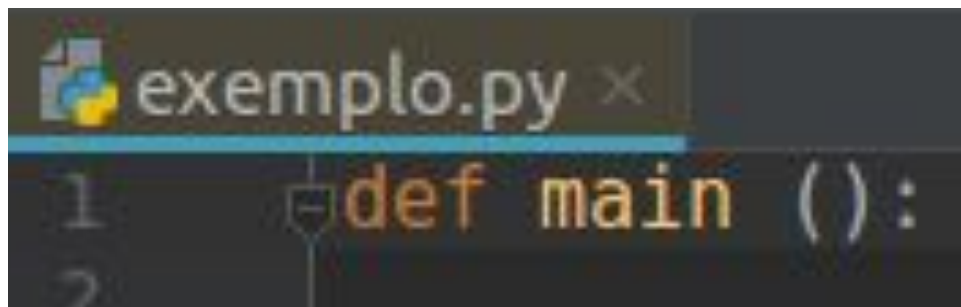
Faça um programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês.

Calcule e mostre o total do seu salário no referido mês.

Exercício F

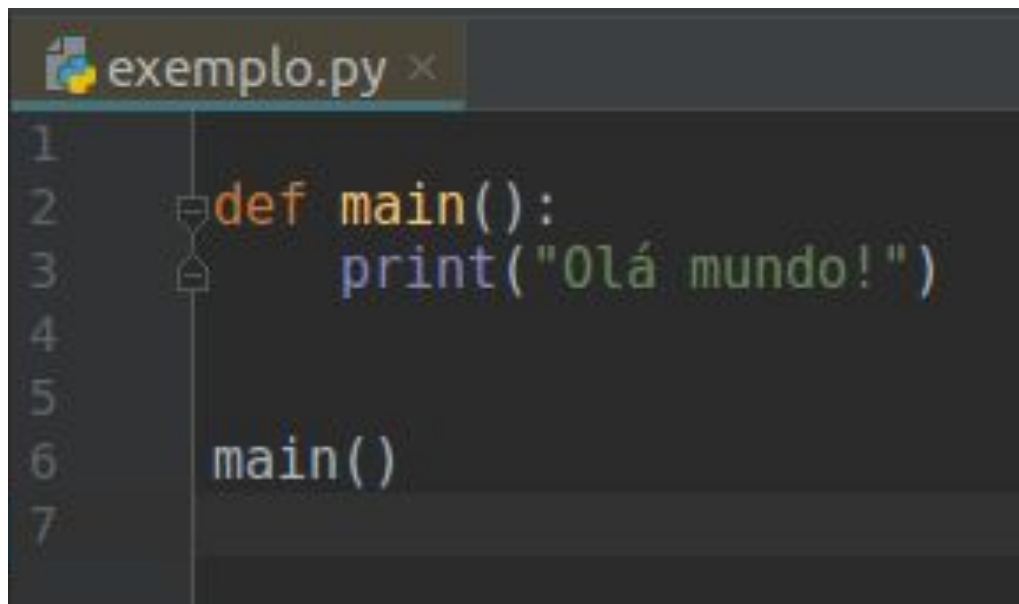
Faça um programa para uma loja de tintas. O programa deverá pedir o tamanho em metros quadrados da área a ser pintada. Considere que a cobertura da tinta é de 1 litro para cada 3 metros quadrados e que a tinta é vendida em latas de 18 litros, que custam R\$ 80,00. Informe ao usuário a quantidades de latas de tinta a serem compradas e o preço total.

Funções



A screenshot of a code editor window titled "exemplo.py". The editor shows a Python function definition on line 1: `def main ():`. The word "def" is highlighted in orange, and "main" is in white. The opening parenthesis "(" is also in white, and the closing parenthesis ")" is in orange. The colon ":" is in white. Line 2 is partially visible and contains the number "2".

Funções



A screenshot of a code editor window titled "exemplo.py". The editor shows a Python function definition and its invocation. The code is as follows:

```
1
2 def main():
3     print("Olá mundo!")
4
5
6 main()
7
```

The code is displayed with syntax highlighting: `def` is orange, `main()` is orange, `:` is white, `print` is blue, and the string `"Olá mundo!"` is green. Line numbers 1 through 7 are visible on the left side of the editor.

Indentação



```
1  
2 def main():  
3     print("Olá mundo!")  
4  
5
```

Bloco

```
valor = input("Digite um número: ")
num = int(valor) #convertendo de str p/ int

print("O número digitado foi: ", str(num))

if (num > 10):
    print("O número digitado é maior do que 10 ")
    print()
    print()
    print()
    print()
```

Bloco

```
#coding: utf-8

#bloco 1
#bloco 1
#bloco 1
    #bloco 2
    #bloco 2
    #bloco 2
        #bloco 3
        #bloco 3
        #bloco 3
        #bloco 3
    #bloco 4
    #bloco 4
    #bloco 4
    #bloco 4
#bloco 1
#bloco 1
#bloco 1
```

Exercício G

Faça um Programa que peça 2 números inteiros e um número real. Calcule e mostre:

1. o produto do dobro do primeiro com metade do segundo .
2. a soma do triplo do primeiro com o terceiro.
3. o terceiro elevado ao cubo.

Exercício H

Faça um programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês, sabendo-se que são descontados 11% para o Imposto de Renda, 8% para o INSS e 5% para o sindicato, faça um programa que nos dê:

1. salário bruto.
2. quanto pagou ao IPRF
3. quanto pagou ao INSS.
4. quanto pagou ao sindicato.
5. o salário líquido.
6. o valor descontado

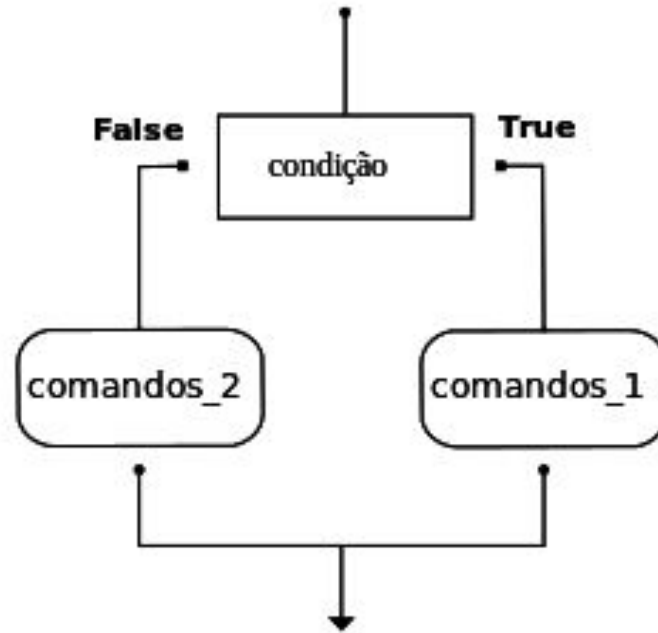
Estruturas condicionais

```
12  if (x == True):  
13      print("ação 1")  
14  else:  
15      print("ação 2")  
16
```


Estruturas condicionais

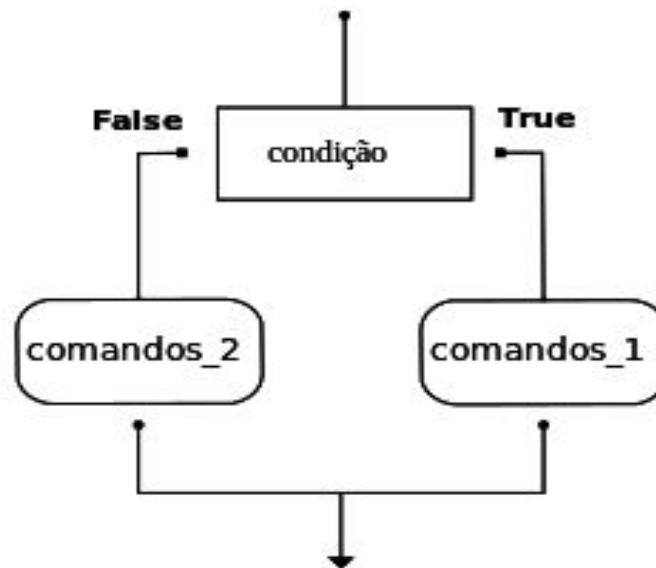
Possibilita a escolha de um grupo de ações e estruturas a serem executadas quando determinadas condições são ou não satisfeitas

Estruturas condicionais



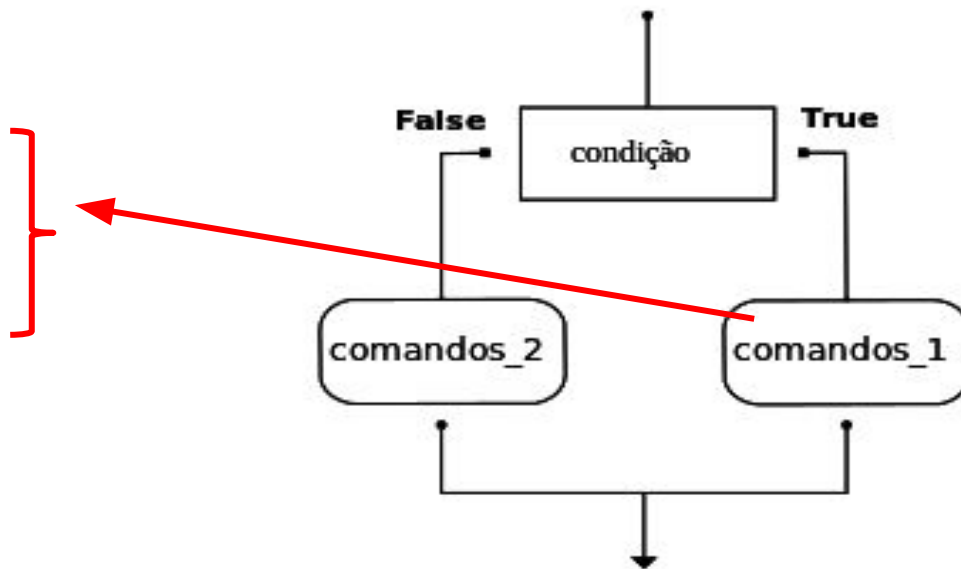
Estruturas condicionais - Python

```
if (condição):  
    comandos_1  
else:  
    comandos_2
```



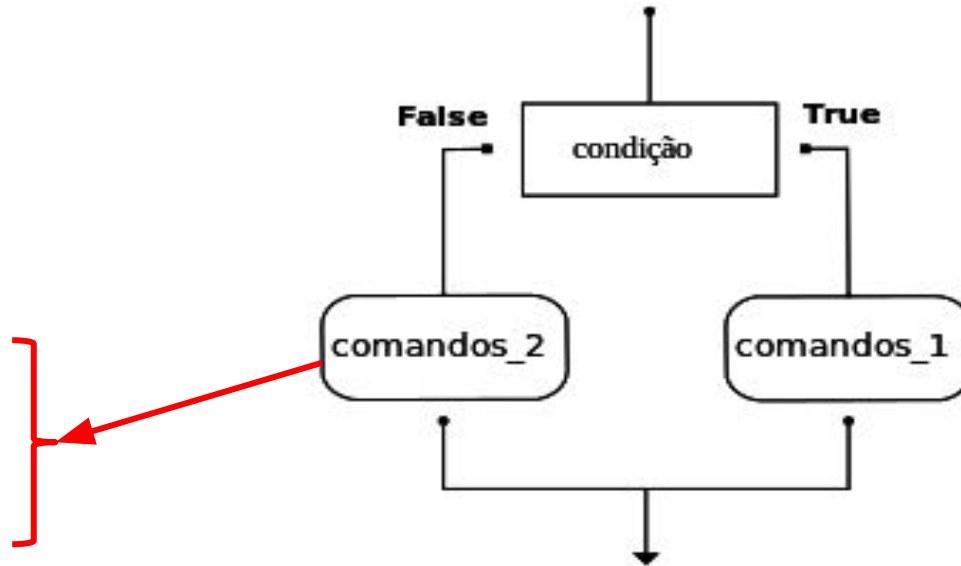
Estruturas condicionais - Python

```
if (condição):  
    comandos_1  
else:  
    comandos_2
```



Estruturas condicionais - Python

```
if (condição):  
    comandos_1  
else:  
    comandos_2
```



Estruturas condicionais - Python

```
if (condição):
```

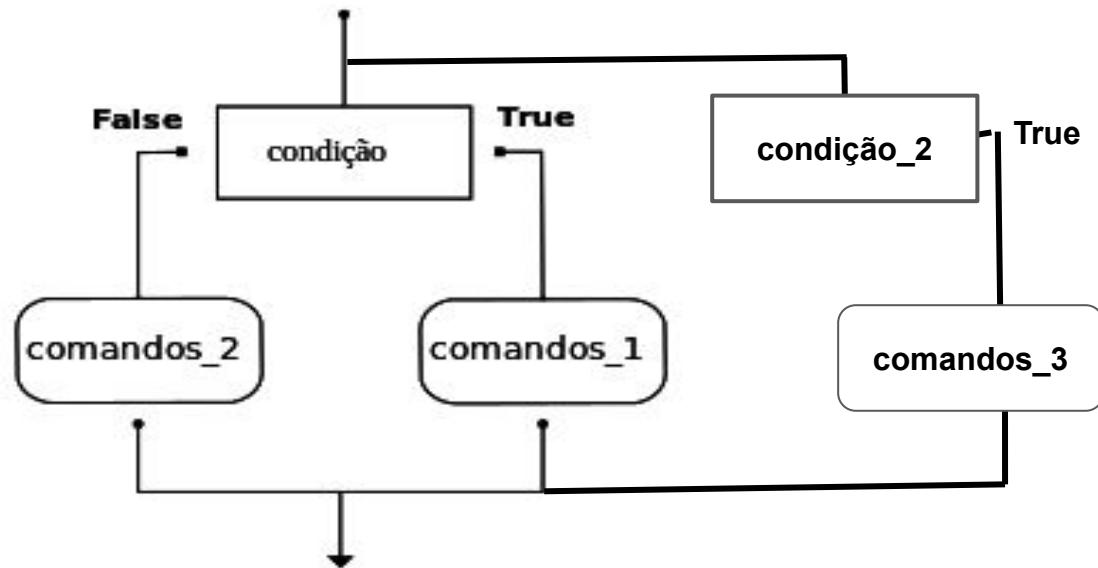
```
    comandos_1
```

```
elif(condição_2):
```

```
    comandos_3
```

```
else:
```

```
    comandos_2
```



Condição - Operadores

Operadores Relacionais

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

Condição - Operadores

in	verifica se o valor A está contido em um conjunto.
!	negação
not	Representa a negação (inverso) da variável atual. Se ela for verdade, torna-se falsa, e vice-versa.
and	operador lógico onde a resposta da operação é verdade se ambas as variáveis de entrada forem verdade.
or	operador lógico onde a resposta da operação é verdade se e somente se pelo menos uma das variáveis de entrada for verdade.

Condição composta

```
if (condição_1 AND condição_2):
```

Verdadeira

Verdadeira

Condição composta

Verdadeira Falsa
if (condição_1 **OR** condição_2):

Condição composta

Falsa Verdadeira
if (condição_1 **OR** condição_2):

Condição composta

Verdadeira Verdadeira
if (condição_1 **OR** condição_2):

Condição composta

if ^{Falsa} (!condição_1):

Condição composta

Verdadeira

```
if(condição_1):
```

Condição composta

if (condição_1 AND condição_2):

if (condição_1 OR condição_2):

if (!condição_1):

if(condição_1):

Exercício I

Faça um programa que peça um valor e mostre na tela se o valor é positivo ou negativo.

Exercício J

Faça um Programa que verifique se uma letra digitada é "F" ou "M". Conforme a letra escrever: F - Feminino, M - Masculino, Gênero Inválido.

Exercício K

Faça um programa para a leitura de duas notas parciais de um aluno. O programa deve calcular a média alcançada por aluno e apresentar:

- A mensagem "Aprovado", se a média alcançada for maior ou igual a sete;
- A mensagem "Reprovado", se a média for menor do que sete;
- A mensagem "Aprovado com Distinção", se a média for igual a dez.

Exercício L

Faça um programa que pergunte o preço de três produtos e informe qual produto você deve comprar, sabendo que a decisão é sempre pelo mais barato.

Estruturas de repetição

- Algumas vezes, uma parte do código precisa ser executada mais de uma vez.

Estruturas de repetição

- Algumas vezes, uma parte do código precisa ser executada mais de uma vez.

Exemplo: Imprima na tela os valores entre 1 a 5:

```
print(1)
print(2)
print(3)
print(4)
print(5)
```

Estruturas de repetição

- Algumas vezes, uma parte do código precisa ser executada mais de uma vez.

Exemplo: Imprima na tela os valores entre 1 a 1000000:

Estruturas de repetição

- Algumas vezes, uma parte do código precisa ser executada mais de uma vez.

Exemplo: Imprima na tela os valores entre 1 a 1000000:



Estruturas de repetição



- Os comandos serão repetidos zero ou mais vezes, enquanto o valor da condição for verdadeiro.

Estruturas de repetição - Python

- for
- while

Estruturas de repetição - Python

- **for**
- **while**

Estruturas de repetição - for

```
for variável_temporária in objeto:  
    bloco de instrução
```


Estruturas de repetição - for

```
lista = ["p", "y", "t", "h", "o", "n"]
```

```
for item in lista:  
    print item
```

Estruturas de repetição - for

```
lista = ["p", "y", "t", "h", "o", "n"]
```




```
for item in lista:  
    print item
```

Estruturas de repetição - for

```
lista = ["p", "y", "t", "h", "o", "n"]
```

```
for item in lista:  
    print item
```



variável temporária

Estruturas de repetição - for

```
lista = ["p", "y", "t", "h", "o", "n"]
```

```
for item in lista:
```

```
    print item
```

p

y

t

h

o

n

Estruturas de repetição - Python

- for
- **while**

while

```
while condição:  
    bloco de instrução
```

while

```
count = 0
while count <= 5:
    print(count)
    count = count + 1
```

while

```
count = 0
```

```
while count <= 5:  condição verdadeira
```

```
    print(count)
```

```
    count = count + 1
```



while

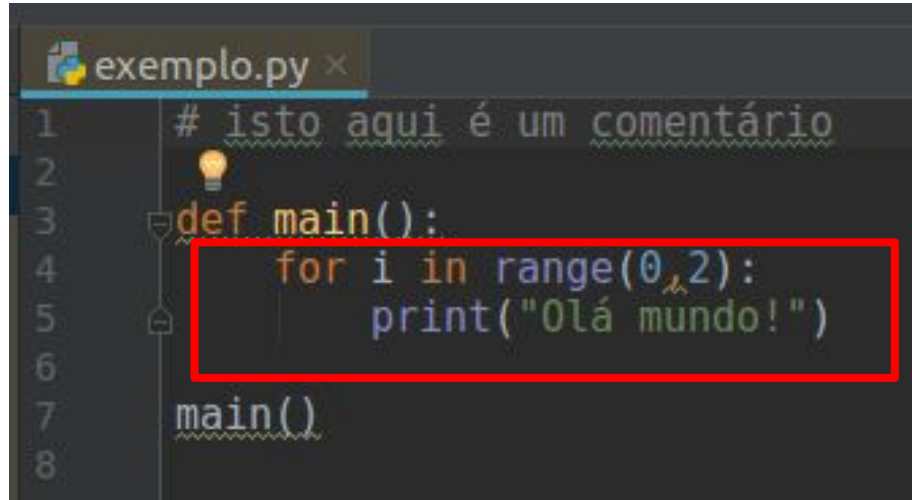
```
count = 0
while count <= 5:
    print(count)
    count = count + 1
```

0
1
2
3
4
5

Estruturas de repetição

```
exemplo.py x
1  # isto aqui é um comentário
2
3  def main():
4      x = 1
5      while True:
6          print("To infinity and beyond! We're getting close, on %d now!" % (x))
7          x += 1
8
9  main()
10
```

Estruturas de repetição



A screenshot of a Python code editor window titled "exemplo.py". The code is as follows:

```
1 # isto aqui é um comentário
2
3 def main():
4     for i in range(0, 2):
5         print("Olá mundo!")
6
7 main()
8
```

The code block from line 4 to line 5 is highlighted with a red rectangle. The code includes a comment on line 1, a function definition on line 3, a loop on lines 4-5, and a function call on line 7.

Exercício M

Dados três números, imprimi-los em ordem crescente.

Entrada:

num1=5

num2=1

num3=2

Saída:

sequência = 1, 2 ,5

Exercício N

Faça um programa que calcule e mostre a média aritmética de N notas (pergunte o número de notas).

Exercício O

Dada uma sequência de números inteiros, imprimir seus quadrados.

Entradas:

nº de início da sequência = 0

nº de fim da sequência = 4

Saída:

sequência : 0, 1, 4, 9, 16

Exercício P

Dado um inteiro não-negativo n , determinar $n!$

Entrada:

$$n = 4$$

Saída:

$$1 \times 2 \times 3 \times 4 = 24$$

Exercício Q

Desenvolva um gerador de tabuada, capaz de gerar a tabuada de qualquer número inteiro entre 1 a 10. O usuário deve informar de qual número ele deseja ver a tabuada.

Biblioteca

```
exemplo.py x
1  from datetime import date
2
3  now = date.today()
4  print(now)
5
```

```
exemplo.py x
1  import datetime
2
3  now = datetime.date.today()
4  print(now)
5
```

Biblioteca

- os (Sistema Operacional);
- glob
- sys
- re
- math
- random
- urllib2
- zlib
- timeit ...

Exercício R

Digite um número e descubra se ele é um palíndromo.

Palíndromo:

o 1º algarismo de n é igual ao seu último algarismo,

o 2º algarismo de n é igual ao penúltimo algarismo, e assim sucessivamente.

Exercício S

Escreva um programa que leia três números naturais e verifica se eles formam os lados de um triângulo retângulo.

$$\text{hipotenusa}^2 = \text{cateto}^2 + \text{cateto}^2$$

Exercício T

Faça um programa que leia e valide as seguintes informações:

1. Nome: maior que 3 caracteres;
2. Idade: entre 0 e 150;
3. Salário: maior que zero;
4. Sexo: 'f' ou 'm';
5. Estado Civil: 's', 'c', 'v', 'd';
6. Sair

1) Atividade para entregar (parte 1) - classroom

Escreva uma função que receba por parâmetro o peso metabólico (PM) de um animal (peso total do animal expresso em quilogramas) e o grupo ao qual esse animal pertence e calcule a taxa metabólica basal (TMB).

A taxa metabólica basal é o peso metabólico elevado a 0.75 e multiplicado por uma constante K.

O valor de K depende do grupo do animal, conforme a tabela a seguir.

1) Atividade para entregar (parte 2) - classroom

Fórmula de cálculo da taxa metabólica basal:

$$\text{TMB} = (\text{PM}^{0.75}) * K$$

Código do grupo	Nome do grupo	Constante (K)
1	Passeriformes	129
2	Não passeriformes	78
3	Mamíferos Placentários	70
4	Marsupiais e Edentatas	49
5	Répteis	10

2) Atividade para entregar (parte 1) - classroom

Escreva uma função que receba por parâmetro o peso metabólico (PM) de um animal (peso total do animal expresso em quilogramas), o grupo ao qual esse animal pertence e uma letra (flag).

Se a letra for B, o procedimento deve chamar uma função para calcular a taxa metabólica basal (TMB).

Se a letra for E, o procedimento deve chamar uma função para calcular a taxa metabólica específica (TME). Para o cálculo da TME, usar a fórmula a seguir.

2) Atividade para entregar (parte 2) - classroom

O valor da constante K corresponde ao grupo do animal, sendo o mesmo tanto para o cálculo da TMB como da TME.

Fórmula de cálculo da taxa metabólica específica:

$$TME = (PM^{0.25}) * K$$

Dúvidas?

alanamm.prof@gmail.com

