

Design de Software

Aula: Introdução ao Python, variáveis e funções

Objetivos de Aprendizagem do Curso

- Desenvolver programas de computador
- Identificar e desenhar estratégias algorítmicas computacionais
- Atuar em uma equipe autogerenciada de desenvolvimento

Informações importantes

- Office Hours (Atendimento)
 - Turma A: Terça 08h00~09h30
 - Turma B: Sexta 13h30~15h00
- Professora Assistente
 - Bárbara Tieko Agena

Atenção: verificar periodicamente o Blackboard e o e-mail para comunicados importantes

Mapeamento Objetivos-Avaliações

- Os objetivos serão mapeados nos seguintes instrumentos de avaliação:

	Provas	EP	Projeto
Obj. 1	X	X	X
Obj. 2	X	X	X
Obj. 3		X	X

Provas: Quiz, AI, e AF – Individual

EPs: Exercício programa – Duplas

Projeto: Projeto Final – Grupo de 3 pessoas

Critérios de Avaliação

- NI - Nota Individual:

Avaliação	Peso
AI	50%
AF	60%

- Atenção: A Prova Sub engloba todo o conteúdo

Critérios de Avaliação

- NG - Nota em Grupo:

Item	Peso
EP	20%
Projeto Final	80%

- Ambos serão avaliados por rubricas e será atribuído um conceito
- Para a conversão de conceito para nota, será utilizada a tabela oficial do Blackboard:

Conceito	Nota
A	9
B	7
C	5
D	4
I	0

Critérios de Avaliação

- Nota Final do curso:

Criterio final		NI	
		< 5	≥ 5
NG	< 5	$\min(\text{NI}, \text{NG})$	$\min(\text{NI}, \text{NG})$
	≥ 5	$\min(\text{NI}, \text{NG})$	$(\text{NI}+\text{NG})/2$

- Caso o aluno não entregue o EP, será reprovado por APS.
- Não existe recuperação (trabalho extra, prova extra, etc)

Datas Importantes

Provas

- P1 – Avaliações Intermediárias
- P2 – Avaliações Finais

Exercício Problema: entrega dia 06/10

Projeto: entrega dia 25/11

Linguagens de programação?



Fonte: <http://oldcomputers.net/altair-8800.html>

Como começamos a programar

Hello World!: programa de computador que imprime "Hello, World!", usualmente seguido de uma quebra de linha. Utilizado como um exemplo de código minimalista para introduzir as pessoas a uma linguagem de programação.



Linguagens e mais linguagens

“Hello, world” em várias linguagens

```
[ASSEMBLY]
MODEL SMALL
IDEAL
STACK 100H

DATASEG
    MSG DB 'Hello, world!', 13, '$'

CODESEG
Start:
    MOV AX, @data
    MOV DS, AX
    MOV DX, OFFSET MSG
    MOV AH, 09H
    INT 21H
    MOV AX, 4C00H
    INT 21H
END Start

[FORTRAN]
00      program hello
        write(*,*) 'Hello World!'
        stop
        end

[BASIC]
10 PRINT "Hello, world!"
20 END

[COBOL]
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-WORLD.
PROCEDURE DIVISION.
    DISPLAY "Hello, world!".
    STOP RUN.

[PROLOG]
:- write('Hello, world!'), nl.

[PYTHON]
print("Hello, world!")

[VBA]
Private Sub Form_Click()
    Form1.Hide
    Dim HelloWorld As New Form1
    HelloWorld.Width = 2500: HelloWorld.Height = 1000: HelloWorld.Caption = "Hello, world!":
    HelloWorld.CurrentX = 500: HelloWorld.CurrentY = 75: HelloWorld.Show: HelloWorld.Font = "Tahoma":
    HelloWorld.FontBold = True: HelloWorld.FontSize = 12: HelloWorld.Print "Hello, world!"
End Sub

[JAVA]
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}

[C]
#include <stdio.h>
int main(void) {
    printf("Hello, World!\n");
    return 0;
}

[PASCAL]
Program Hello;
begin
    writeln('Hello, world.');
end.

[BATCH]
@echo off
echo Hello World!
pause
exit


```

Python

O nome Python teve a sua origem no grupo humorístico britânico Monty Python



Fonte: <http://tinyurl.com/ls3w45p>

Onde Python é usado?

usam largamente Python:



reddit

yahoo!

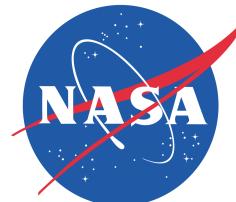
Google



Dropbox



Instagram



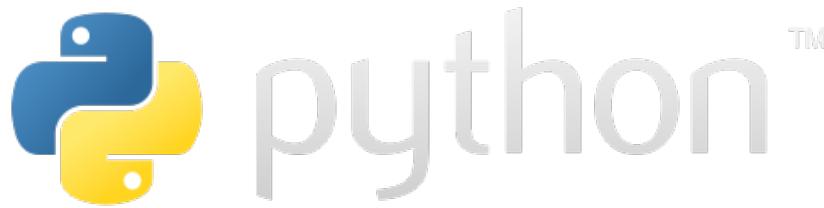
Uber

NETFLIX



Python

Instalar Python



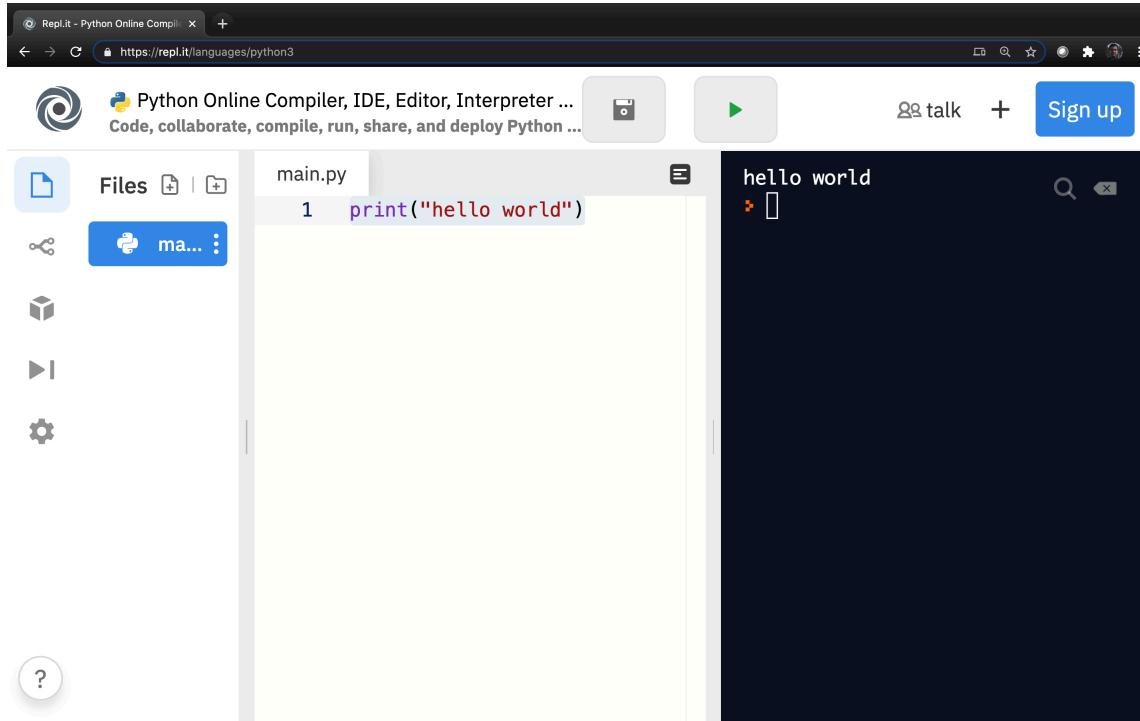
<https://www.python.org/>



Anaconda

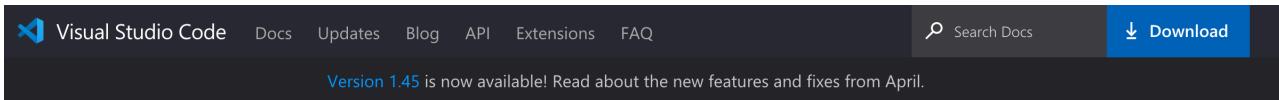
<https://www.anaconda.com/>

Para a aula hoje (e somente hoje)



<http://repl.it/languages/python3>

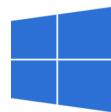
VS Code



The screenshot shows the official Visual Studio Code website. At the top is a dark header with the "Visual Studio Code" logo, a search icon, and a "Download" button. Below the header is a banner announcing "Version 1.45 is now available! Read about the new features and fixes from April." The main content area features the title "Download Visual Studio Code" and the subtitle "Free and built on open source. Integrated Git, debugging and extensions." Below this are three large download links: one for Windows (Windows logo), one for Linux (Tux logo), and one for Mac (Apple logo). Each link has a sub-section below it listing specific installer options.

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows

Windows 7, 8, 10



↓ .deb

Debian, Ubuntu



↓ Mac

macOS 10.10+

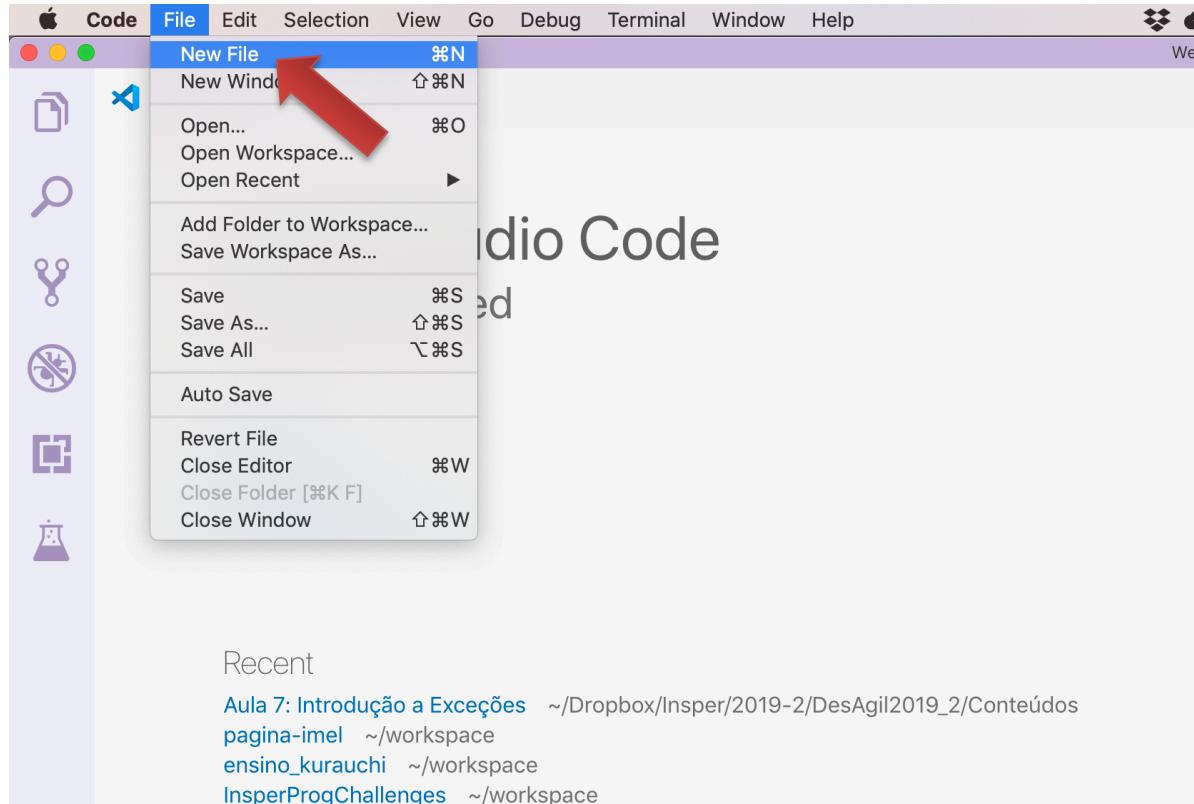
User Installer 64 bit 32 bit
System Installer 64 bit 32 bit
.zip 64 bit 32 bit

.deb 64 bit
.rpm 64 bit
.tar.gz 64 bit

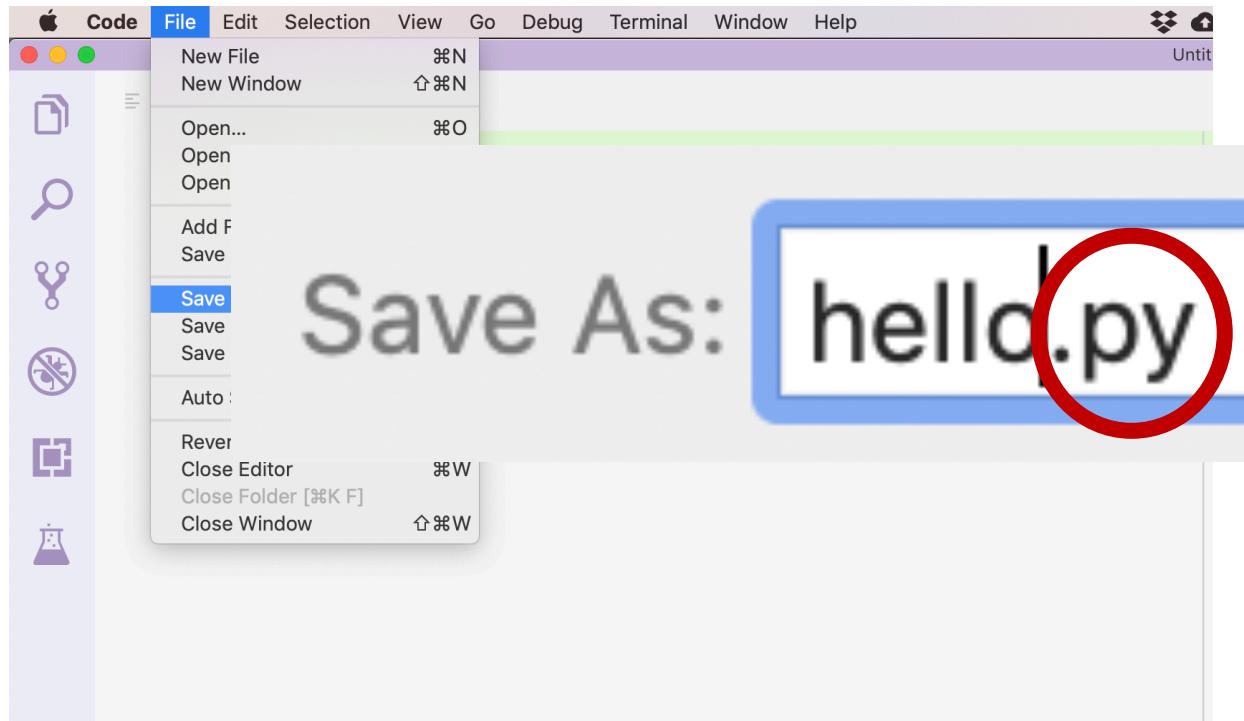
[Snap Store](#)

<https://code.visualstudio.com/download>

Primeiro Programa



Primeiro Programa



Primeiro Programa: escrevendo

hello.py

hello.py

Users > toshi > workspace > insper > DesSoft > Aula01 > hello.py

```
1 print('Hello, world!')
```

File Edit Selection View Go Debug Terminal Help

hello.py

C: > Users > toshi > workspace > insper > DesSoft > Aula01 > hello.py

```
1 print('Hello, world!')
```

No Windows...

1

Python 3.7.1 64-bit 0 ! 0 i 1

UTF-8 LF Python 😊 1

Primeiro Programa: executando (Windows)

The screenshot shows a Windows desktop environment. In the foreground, there is a dark-themed code editor window titled "hello.py". The file content is:

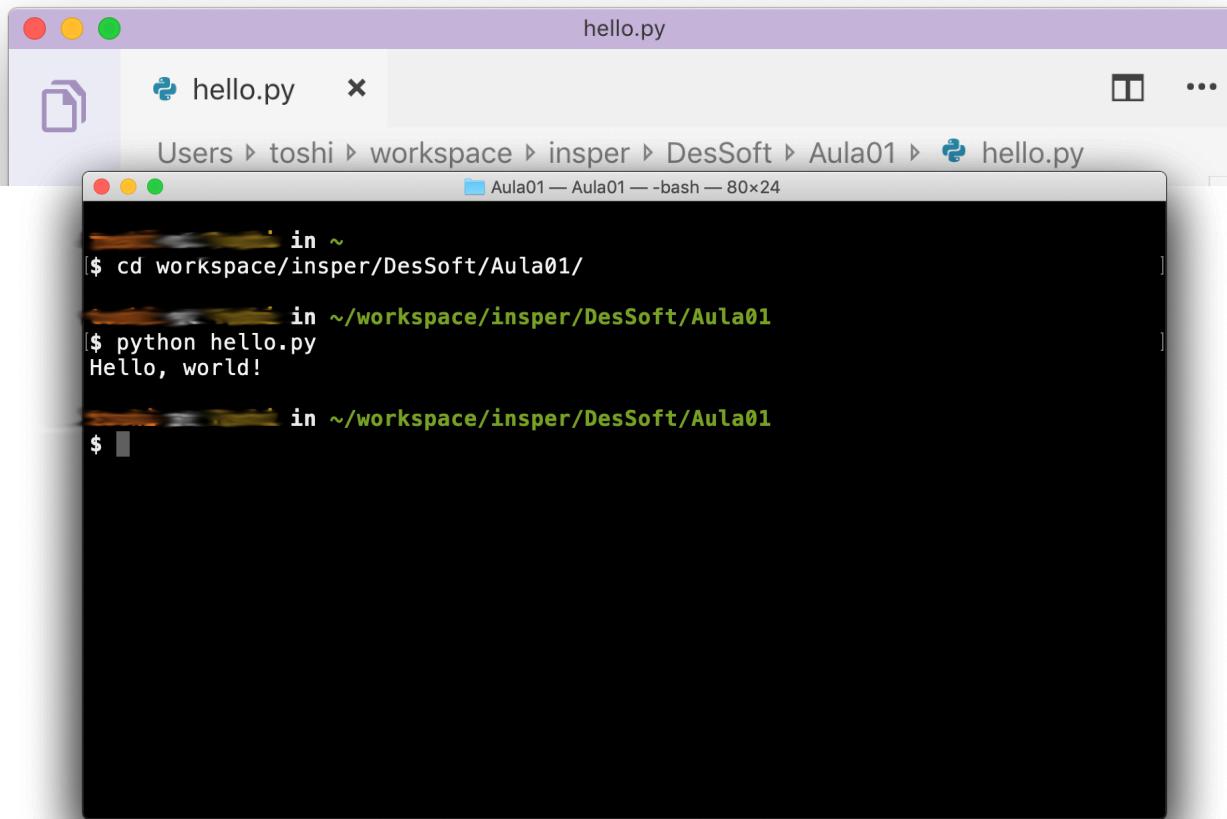
```
C: > Users > toshi > workspace > insper > DesSoft > Aula01 > hello.py
```

Below the code editor, a terminal window titled "Anaconda Prompt" is open. The terminal output is:

```
an internal or external command,  
operable program or batch file.  
  
C:\Users\toshi>cd workspace\insper\DesSoft\Aula01  
  
C:\Users\toshi\workspace\insper\DesSoft\Aula01>python hello.py  
Hello, world!  
  
C:\Users\toshi\workspace\insper\DesSoft\Aula01>
```

The desktop background features a red and white striped pattern.

Primeiro Programa: executando (OS X)

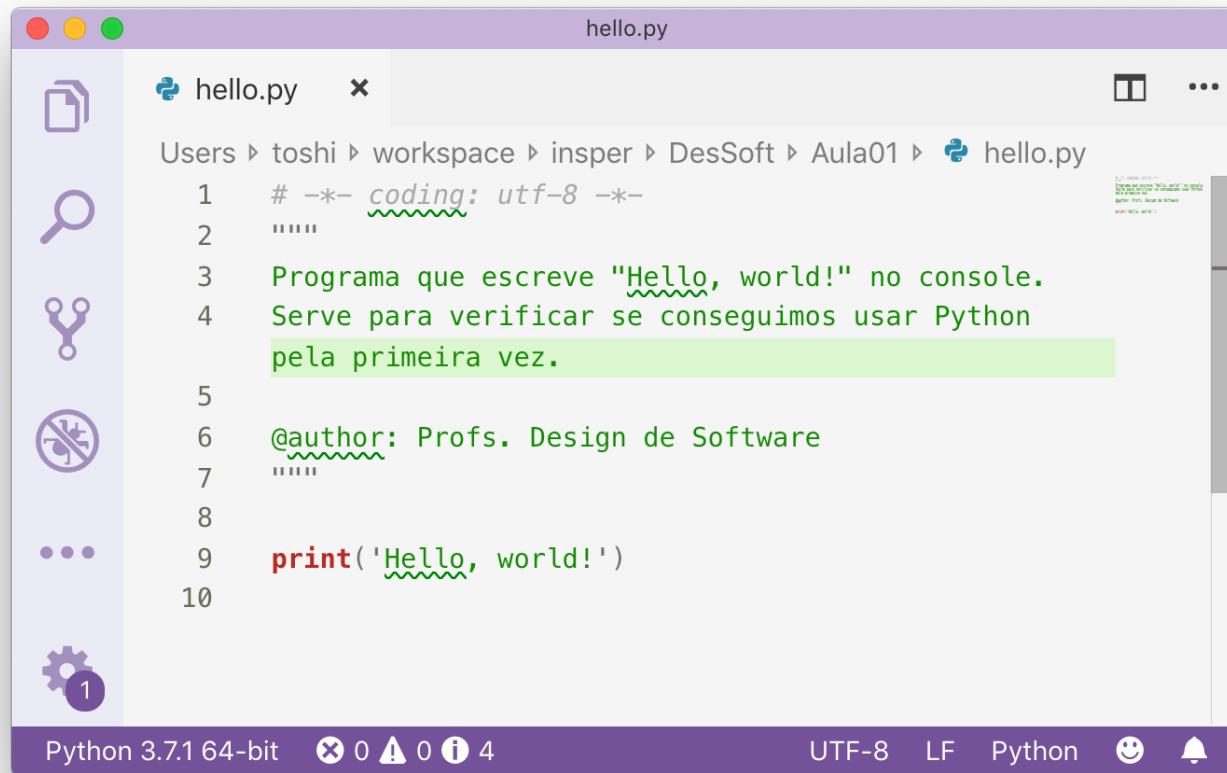


The image shows a Mac OS X desktop environment. In the foreground, a terminal window is open with the title bar "Aula01 — Aula01 — bash — 80x24". The terminal shows the following command-line session:

```
in ~
$ cd workspace/insper/DesSoft/Aula01/
in ~/workspace/insper/DesSoft/Aula01
$ python hello.py
Hello, world!
in ~/workspace/insper/DesSoft/Aula01
$
```

Behind the terminal, a file browser window titled "hello.py" is visible. The file icon is a blue Python logo. The path listed in the file browser is "Users > toshi > workspace > insper > DesSoft > Aula01 > hello.py".

Dica Pro: documente o código!



The screenshot shows a Python code editor window titled "hello.py". The sidebar on the left contains icons for file operations, search, and other tools. The main pane displays the following code:

```
hello.py
Users > toshi > workspace > insper > DesSoft > Aula01 > hello.py
1 # -*- coding: utf-8 -*-
2 """
3 Programa que escreve "Hello, world!" no console.
4 Serve para verificar se conseguimos usar Python
5 pela primeira vez.
6
7 @author: Profs. Design de Software
8
9 print('Hello, world!')
10
```

The code includes a multi-line docstring starting at line 3, which is highlighted with a green background. The code editor interface includes a status bar at the bottom showing "Python 3.7.1 64-bit", build statistics (0 errors, 0 warnings, 4 info), encoding ("UTF-8"), line ending ("LF"), language ("Python"), and icons for smiley face and notifications.

Outras fontes de ajuda

- Documentação oficial da linguagem Python:
<https://docs.python.org/3/>
- StackOverflow: <https://stackoverflow.com/>
- Páginas de busca em geral, obviamente!

Como estudar?

Servidor de Desafios Olá Fulano Silva [trocar senha] Sandbox

Logout

[Exercício 1: Juros Compostos](#)

[Exercício 2: Velocidade média](#)

[Exercício 3: Calcula gaussiana](#)

[Exercício 4: Classificador de idade](#)

[Exercício 5: Maior primo menor que N](#)

[Exercício 6: Máximo de uma matriz](#)

[Exercício 7: Norma de um vetor](#)

[Exercício 8: Progressão Aritmética e Progressão Geométrica](#)

[Exercício 9: Listando todos os sufixos de uma string](#)

[Exercício 10: Convertendo libras](#)

[Exercício 11: Celsius para Fahrenheit](#)

[Exercício 12: Área do triângulo](#)

[Exercício 13: Movimento retílineo uniforme](#)

[Exercício 14: Volume de uma esfera](#)

[Exercício 15: Volume da PIZZA!](#)

[Exercício 16: Distância entre dois pontos](#)

[Exercício 17: Equação de primeiro grau](#)

[Exercício 18: Cateto de um triângulo retângulo](#)

[Exercício 19: Lançamento de projétil](#)

[Exercício 20: Todo mundo odeia o](#)

Tutoriais

- [Tutorial 1: Iniciando em Python](#)
- [Tutorial 2: Desvio Condicional](#)

Exercícios

Exercício	Tentativas	Resolvido	Tags
Exercício 1: Juros Compostos	0	-	função
Exercício 2: Velocidade média	0	-	função
Exercício 3: Calcula gaussiana	0	-	função
Exercício 4: Classificador de idade	0	-	função if
Exercício 5: Maior primo menor que N	0	-	if while loop
Exercício 6: Máximo de uma matriz	0	-	while lista loop
Exercício 7: Norma de um vetor	0	-	lista for loop
Exercício 8: Progressão Aritmética e Progressão Geométrica	0	-	if while for loop
Exercício 9: Listando todos os sufixos de uma string	0	-	while lista string loop
Exercício 10: Convertendo libras	0	-	função
Exercício 11: Celsius para Fahrenheit	0	-	função
Exercício 12: Área do triângulo	0	-	função

Operações matemáticas

```
print('Usando alguns operadores numéricos: ')
print(2 + 3) # Soma.
print(2 - 3) # Subtração.
print(2 * 3) # Multiplicação.
print(2 ** 3) # Potenciação ou exponenciação.

print(7 / 3) # Divisão.
print(7 // 3) # Divisão inteira: quociente.
print(7 % 3) # Resto da divisão inteira.

print((1 + 2) * (-3 - 2)) # Parênteses.
```

Operadores e Operandos

Operador	Ação
+	Adição
-	Subtração
*	Multiplicação
**	Potência
/	Divisão
//	Divisão de inteiros
%	Resto da divisão

Ordem dos Operadores

Python segue a convenção matemática do PEMDAS:

Parênteses têm a maior prioridade e pode ser usado para forçar uma expressão a ser avaliada na ordem que você quiser.

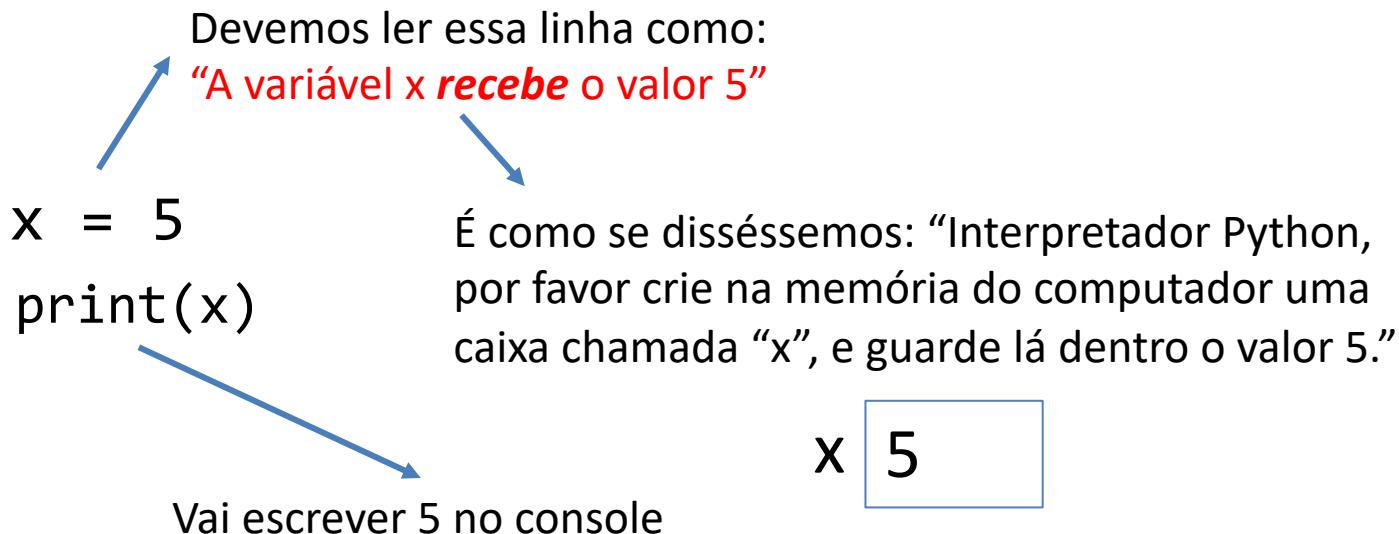
A Exponenciação tem a próxima precedência mais alta.

Multiplicação e Divisão têm a mesma precedência, que é maior do que Adição e Subtração.

Além disso as operações de mesma prioridade são avaliados da esquerda para a direita.

Variáveis

Armazenam valores para o programador



Variáveis

Podemos usar as variáveis dentro de outras expressões:

```
x = 3  
y = 4  
z = x * y
```

```
print('O retângulo de lados {0} e {1} tem área {2}'.format(x, y, z))
```



Observação: usamos `format()` para escrever valores no meio do texto

Variáveis

Podemos usar variáveis para armazenar qualquer tipo de dados em Python:

```
# Variáveis podem armazenar texto (string).  
disciplina = 'Design de Software'  
  
# Variáveis podem armazenar inteiros (int).  
num_alunos = 120  
  
# Variáveis podem armazenar números reais. Em computação,  
# são chamados de números em ponto flutuante (float).  
base = 0.5  
altura = 4.2e1 # 4.2 x 10**1 => 42.0  
area = base * altura / 2.0
```

Variáveis

Podemos usar variáveis para armazenar qualquer tipo de dados em Python:

```
# Variáveis podem armazenar os valores especiais True e False
# São usados em expressões lógicas, ou também chamadas
# expressões booleanas.
a = True
b = False
c = a or b # Operação booleana 'OU'.

# Variáveis armazenam muitos outros tipos que veremos depois.
ponto = (3.2, 5.7) # Tupla.
teste = [5.2, 'laranja', False, 42] # Lista.
notas = {'dessoft': 10.0, 'gde': 10.0,
         'modsim': 10.0} # Dicionário.
```

Atividade

O que será impresso pelo programa a seguir?

```
a = 5  
b = 7  
a = a + 5
```

```
print("a vale {0}, b vale {1}".format(a, b))
```

```
b -= 3 # Equivale a b = b - 3  
c = a * b  
c += 2 # Equivale a c = c + 2
```

```
print("c vale {0}".format(c))
```

a vale 10, b vale 7
c vale 42

Funções

Seja uma função matemática que converte uma distância em milhas para km:

$$f(x) = 1.60934 x$$

Em Python: Argumento da função

```
def f(x):  
    y = 1.60934*x  
return y
```

Retorno da função

Funções

Seja uma função matemática que converte uma distância em milhas para km:

$$f(x) = 1.60934 x$$

Definição da
função

Em Python:

```
def f(x):  
    y = 1.60934*x  
    return y
```

Indentação

Corpo da função

Funções

Agora podemos usar nossa nova função matemática em nossos cálculos:

```
def f(x):  
    y = 1.60934*x  
    return y
```

```
# Quanto é 10 mi em km?  
a = 10  
b = f(a)  
print(b)
```

Aqui estamos somente **definindo** a função $f(x)$

- O def é o “manual de instruções da função” (chamado de *definição da função*)
- A definição serve para que o Python saiba o que fazer quando você fizer a *chamada da função* (ou seja, quando você pedir ao Python que use a função para fazer algo para você)
- É só a definição, nada executa ainda!

Funções

Agora podemos usar nossa nova função matemática em nossos cálculos:

```
def f(x):  
    y = 1.60934*x  
    return y  
  
# Quanto é 10 mi em km?  
a = 10  
b = f(a)  
print(b)
```

Aqui estamos chamando a função,
agora sim o código da função vai rodar!

Funções

Vamos fazer um “teste de mesa” para ver como tudo funciona:

```
def f(x):
    y = 1.60934*x
    return y

# Quanto é 10 mi em km?
a = 10
b = f(a)
print(b)
```

Definição: nada acontece. É só o Python conhecendo o manual de instruções da função

Memória do computador



Console



Funções

Vamos fazer um “teste de mesa” para ver como tudo funciona:

```
def f(x):
    y = 1.60934*x
    return y

# Quanto é 10 mi em km?
a = 10
b = f(a)
print(b)
```

Comentário: ignorado

Memória do computador

Console

Funções

Vamos fazer um “teste de mesa” para ver como tudo funciona:

```
def f(x):
    y = 1.60934*x
    return y

# Quanto é 10 mi em km?
a = 10
b = f(a)
print(b)
```

Memória do computador

a: 10

Aqui temos a primeira ação: a variável “a” recebe o valor 10

Console

Funções

Vamos fazer um “teste de mesa” para ver como tudo funciona:

```
def f(x):
    y = 1.60934*x
    return y

# Quanto é 10 mi em km?
a = 10
b = f(a)
print(b)
```

Aqui temos a chamada da função.
Primeiro temos que calcular $f(a)$,
ou seja:

- Chamar a função $f(x)$ com o argumento x valendo o valor de a

Memória do computador

a: 10

Console

Funções

Vamos fazer um “teste de mesa” para ver como tudo funciona:

```
def f(x):  
    y = 1.60934*x  
    return y  
  
# Quanto é 10 mi em km?  
a = 10  
b = f(a)  
print(b)
```

Memória do computador

a: 10

f:

Primeiramente:

- Vamos criar uma “memória privada” para a execução da função f.

Console

Funções

Vamos fazer um “teste de mesa” para ver como tudo funciona:

```
def f(x):  
    y = 1.60934*x  
    return y  
  
# Quanto é 10 mi em km?  
a = 10  
b = f(a)  
print(b)
```

Assim como numa função matemática, o argumento da função valerá aquele valor que foi passado.

Memória do computador

a: 10 f: x: 10

Console

Funções

Vamos fazer um “teste de mesa” para ver como tudo funciona:

```
def f(x):  
    y = 1.60934*x  
    return y  
  
# Quanto é 10 mi em km?  
a = 10  
b = f(a)  
print(b)
```

Memória do computador

a: 10

f:

x: 10

y: 16.0934

Console

Dentro da função executamos o cálculo de y.
IMPORTANTE: tanto x como y existem apenas DENTRO de f

Funções

Vamos fazer um “teste de mesa” para ver como tudo funciona:

```
def f(x):  
    y = 1.60934*x  
    return y  
  
# Quanto é 10 mi em km?  
a = 10  
b = f(a)  
print(b)
```

Vamos agora *retornar* da chamada da função. Onde antes tinha $f(a)$, agora teremos 16.0934

Memória do computador

a: 10

f:

x: 10

y: 16.0934

Console

Funções

Vamos fazer um “teste de mesa” para ver como tudo funciona:

```
def f(x):
    y = 1.60934*x
    return y

# Quanto é 10 mi em km?
a = 10
b = f(a) 16.0934
print(b)
```

Memória do computador

a: 10

Após retornar da chamada da função, a “memória local” de f é destruída

Console

Funções

Vamos fazer um “teste de mesa” para ver como tudo funciona:

```
def f(x):
    y = 1.60934*x
    return y
```

```
# Quanto é 10 mi em
# km?
a = 10
b = f(a) 16.0934
print(b)
```

A variável b é criada, e armazena o valor retornado da chamada de f

Memória do computador

a: 10
b: 16.0934

Console

Funções

Vamos fazer um “teste de mesa” para ver como tudo funciona:

```
def f(x):  
    y = 1.60934*x  
    return y  
  
# Quanto é 10 mi em km?  
a = 10  
b = f(a) 16.0934  
print(b)
```

Finalmente, escrevemos o valor da variável b no console

Memória do computador

a: 10
b: 16.0934

Console

16.0934

Dica Pro: use bons nomes!

```
def f(x):  
    y = 1.60934*x  
    return y  
  
a = 10  
b = f(a)  
print(b)
```

No dia seguinte você abre esse programa: o que faz mesmo este código?

- Use nomes que façam sentido!
 - Variáveis, funções, etc.
- Use comentários!

Dica Pro: use bons nomes!

```
"""
Programa que converte de milhas para kilometros. Versão
melhor documentada.
```

```
@author: Profs. de Design de Software
"""
```

```
# Função que converte um valor de distância
# dado em milhas para quilômetros.
```

```
def converte_milhas_para_km(distancia_mi):
    distancia_km = 1.60934 * distancia_mi
    return distancia_km
```

```
# Quanto é 10 mi em km?
```

```
milhas = 10
```

```
km = converte_milhas_para_km(milhas)
```

```
print(km)
```

Este código está
muito melhor
documentado!

Recursos online

Enthought Academic

<https://www.enthought.com/>

Codecademy

<http://www.codecademy.com/en/tracks/python>

Learn Python

<https://learnpython.trinket.io/>

Insper

www.insper.edu.br