



Curso de Python

26/01/2021

<https://github.com/tsbressan/CursoPython>

Editor

- **Jupyter Notebook**

Funcionamento:

- Clicar no ícone: Jupyter Notebook no menu do Anaconda ou via linha de comando: `>> jupyter notebook`
- É Executado por um navegador (como um servidor web local).

<http://localhost:8888>

Files

Running

Clusters

Select items to perform actions on them.

Upload

New



0



/

Name

Last Modified

File size



codigo1.py

7 dias atrás

810 B



Dia1.pptx

4 minutos atrás

7.95 MB



{ECB14B49-2BC7-4AEA-B645-CABBEAFEB2AB}.tmp

21 horas atrás

7.94 MB

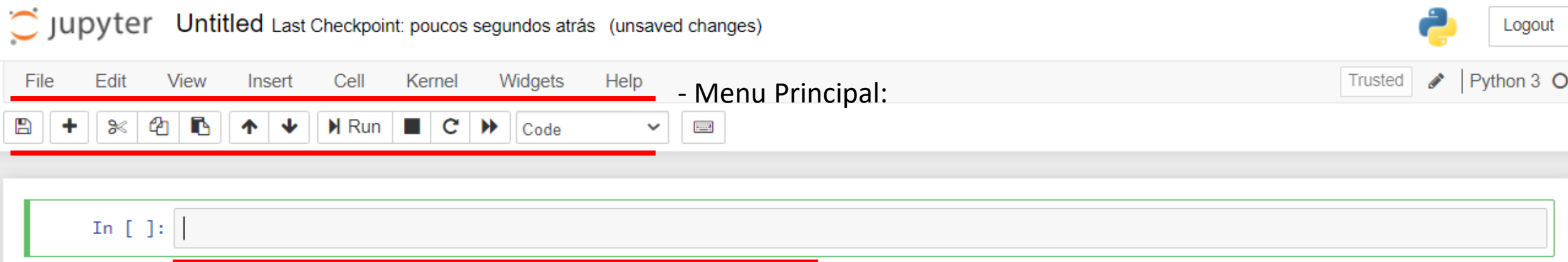
Files = Lista dos arquivos no diretório principal

New = novo arquivo Python3

Running = arquivo em aberto (em funcionamento)

Jupyter Notebook

- New / Python3:



The screenshot displays the Jupyter Notebook interface. At the top, the Jupyter logo is followed by the text "jupyter", the filename "Untitled", and a status message "Last Checkpoint: poucos segundos atrás (unsaved changes)". On the right side of the header, there is a Python logo and a "Logout" button. Below the header is a main menu bar with the following items: "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar, there is a "Trusted" status indicator, a pencil icon, and a "Python 3" dropdown menu. Below the menu bar is a toolbar containing icons for saving, adding a new cell, cutting, copying, pasting, undo, redo, and running the cell. A dropdown menu next to the run icon shows "Code" as the selected cell type. Below the toolbar is a code cell with the prompt "In []:" followed by a text input field.

jupyter Untitled Last Checkpoint: poucos segundos atrás (unsaved changes) Logout

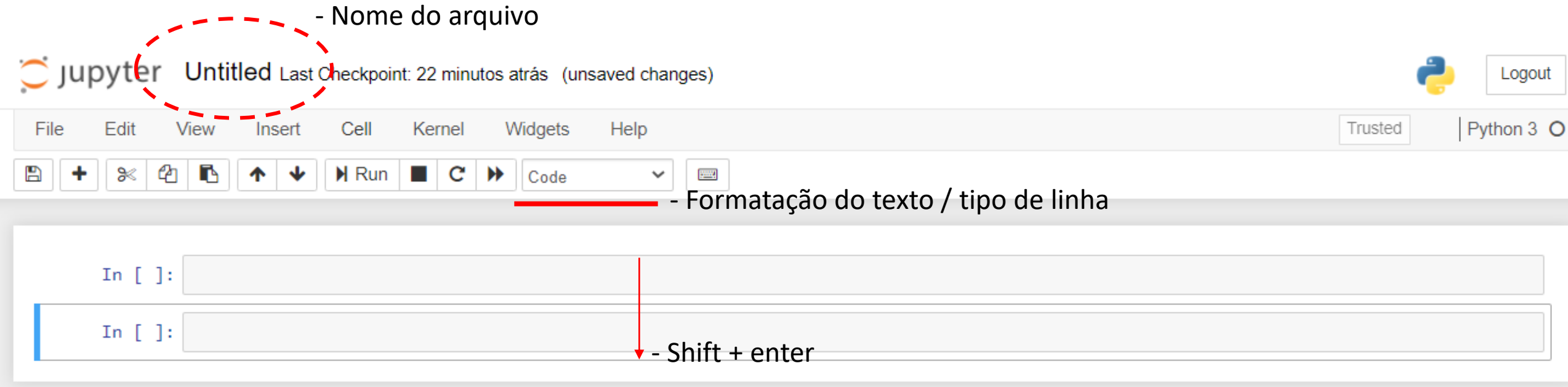
File Edit View Insert Cell Kernel Widgets Help - Menu Principal: Trusted Python 3

In []:

- Código:

Jupyter Notebook

- Nome do arquivo



jupyter Untitled Last Checkpoint: 22 minutos atrás (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

- Formatação do texto / tipo de linha


In []:

In []:

- Shift + enter

- Menu Insert: Insert line Above or Bellow
- Menu Edit: Delete Cells

Jupyter Notebook

 jupyter Untitled Last Checkpoint: 28 minutos atrás (unsaved changes)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3 



In [3]: `print ("Curso de Python")`

Curso de Python

↓ - Shift + enter

In [4]: `numero1 = 1`
`numero2 = 2`

In [5]: `print (numero1 + numero2)`

3

In []:

Estrutura da Linguagem Python – Sintaxe Inicial

- **VARIÁVEIS:**

- Utilizada para armazenar alguma “informação” conforme um tipo de dado
- Deve ser inicializada/criada antes de ser utilizada.
- Não existe criação automática de variáveis em Python

- Por exemplo:

```
>>> soma = numero1 + numero2
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'numero1' is not defined

Estrutura da Linguagem Python – Sintaxe Inicial

- **VARIÁVEIS:**

Tipo de dados em Python

- **inteiro:**

`a = 123` #decimal `a = 017` #Octal inicia em zero

`a = 0xAF` #Hexadecimal inicia em 0x

- **Float:**

`a = 0.024`

- **Long:** #números inteiros longos

`a = 145897896254`

Estrutura da Linguagem Python – Sintaxe Inicial

- **VARIÁVEIS:**

Tipo de dados em Python

- **bool:** #valores booleanos True ou False (Operadores lógicos, and-or)

a=True

b=False

- **None type:** #tipo None, ausência de valores, simular a null

a=None

- **String(str):** #entre aspas simples, duplas ou triplas.

a="Curso"

Estrutura da Linguagem Python – Sintaxe Inicial

- **VARIÁVEIS:**

EXEMPLOS PRÁTICOS(1):

a=2.25 #tipo float

b=55 #tipo inteiro

c=0740 #tipo inteiro octal

e=0xFFAB #tipo inteiro hexadecimal

f="Curso de Python" #tipo string

type(a) #mostra o tipo da variável

Estrutura da Linguagem Python – Sintaxe Inicial

- **VARIÁVEIS:**

EXEMPLOS PRÁTICOS(2):

```
meu_nome = "Carlos"
```

```
meu_sobrenome = 'Santini'
```

```
print ("Nome: %s, Sobrenome: %s" % (meu_nome.upper(), meu_sobrenome))
```

```
print (f'Nome: {meu_nome.upper()}, Sobrenome: {meu_sobrenome}') #Formatted string literals ou f-string
```

```
print("Nome: {0}, Sobrenome: {1}".format(meu_nome.upper(), meu_sobrenome)) #new-style string formatting
```

```
print ("Meu nome começa com a letra ", meu_nome[0])
```

```
print ("Meu nome começa com a letra ", meu_nome[0].lower())
```

```
print ("Meu primeiro nome é ", meu_nome[0:6])
```

```
print ("Meu sobrenome é ", meu_sobrenome[0:7])
```

Estrutura da Linguagem Python – Sintaxe Inicial

- **VARIÁVEIS:**

Conversão de tipos em python

```
a = float(21/4)
```

```
b = int(4.8)
```

```
c = int(4.9)
```

```
d = int(0xff500)
```

```
e = float(int(3.9))
```

```
f = int(float(3.9))
```

```
g = int(float(3))
```

```
h = round(3.9)
```

```
i = round(3)
```

```
j = int(round(3.9))
```

```
print (a,b,c,d,e,f,g,h,i,j)
```


Estrutura da Linguagem Python – Sintaxe Inicial

- **VARIÁVEIS:**

Palavras reservadas – não utilizar como uma variável:

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

Código estruturado com função

```
def main():      #função com o nome main()
    a = 3
     b = 4
    soma = a + b
    print("A soma de a + b e igual a: ", soma)
```

```
main()      # executa a função
```

- Cuidar a indentação;
- Revisar aspas, parênteses, pontos, vírgulas,..., programação é revisar o texto várias e várias vezes

Entrada de dados – input()

- **Responsável por receber os dados que o usuário fornece via teclado.**

Formato:

```
variável = input ("prompt")
```

Exemplo:

```
nome = input ("Qual é o seu nome? ")
```

```
print ("O seu nome é:", nome)
```

Entrada de dados – input()

```
def main():
```

```
    a = input("Digite o primeiro numero: ")
```

```
    b = input("Digite o segundo numero: ")
```

```
    soma = int(a) + int(b)
```

```
    print("A soma de", a, "+", b, "e igual a", soma)
```

```
main()
```


Exemplos práticos:

- **1) Operações matemáticas com 2 variáveis**
- **2) Operações matemáticas com 3 variáveis**
- **3) Operações com String**
- **4) Conversão de tipos**
- **5) Operações com booleanos**
- **6) Operação com String - avançado**

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

- Utilizada para decidir qual bloco de código deve ser executado ou não, através de uma condição lógica em linguagem de programação
- -----

Se estiver chovendo: #expressão verdadeira

Levarei guarda-chuva

Senão: #expressão falsa

Não levarei

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

- Utiliza o bloco de comando se..senão ou se..senãose
- Traduzindo para a linguagem Python: if ...else ou if..elif

```
a = 10
if a > 3:  #verdadeiro
    print("Número maior que 3")
else:     #falso
    print("Número menor que 3")
```

Saída:::

```
>> Número maior que 3
```

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

```
valor_entrada = 10
if valor_entrada == 1:
    print("a entrada era 1")
elif valor_entrada == 2:
    print("a entrada era 2")
elif valor_entrada == 3:
    print("a entrada era 3")
elif valor_entrada == 4:
    print("a entrada era 4")
else:
    print("o valor de entrada não era esperado em nenhum if")
```

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Operadores utilizados na estrutura de controle:

== #igual

> #maior

< #menor

>= #maior igual

<= #menor igual

!= #diferente

and-or # operadores lógicos

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Exemplo utilizando função:

```
def main():
```

```
    ← idade= int(input ("Quantos anos voce tem?"))
```

```
        if idade >= 16:
```

```
            ← print ("Você já tem idade para votar.")
```

```
        elif idade > 10 and idade < 16:
```

```
            print ("Você é adolescente")
```

```
        else:
```

```
            print("Você ainda é um garoto(a). Aproveite para brincar...")
```

```
main()
```

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Exemplo:

def main():

```
←→ Numero1= int(input ("Digite o número 1:"))
    Numero2= int(input ("Digite o Número 2:"))
    if Numero1 > Numero2:
        ←→ print ("Primeiro número é maior que o segundo número")
    elif Numero2 > Numero1:
        print ("Segundo número é maior que o primeiro número")
    else:
        print("Os números são iguais")
```

main()

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Vetor simples unidirecional:

Lista de dados de um mesmo tipo. No Python é possível armazenar em vetor dados de vários 'tipos', a qual chamamos de Lista:

```
Notas = [5.0, 8.5, 7.8, 9.3]  
lista = ['A', 1, 2, 'Casa', 2.3]
```

```
Notas = [5.0, 8.5, 7.8, 9.3]  
índice: 0    1    2    3
```

Para acessar, basta indicar a posição(índice) do vetor (iniciando sempre na posição zero)

Notas[0] -> 5.0

Lista[2] -> 2

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Exemplo utilizando vetor simples:

```
Numeros = [] #inicio do vetor
```

```
def main():
```

```
    Numeros.append(input ("Digite o número 1:"))
```

```
    Numeros.append(input ("Digite o Número 2:"))
```

```
    if Numeros[0] > Numeros[1]:
```

```
        print ("Primeiro número é maior que o segundo número")
```

```
    elif Numeros[1] > Numeros[0]:
```

```
        print ("Segundo número é maior que o primeiro número")
```

```
    else:
```

```
        print("Os números são iguais")
```

```
main()
```

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Erro comum – acessar posição do vetor que não existe

```
alunos = ['Andre', 'Lucas', 'Antonio', 'Maria']  
print(alunos[4])
```

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Mais algumas funções aplicados sobre um vetor de dados:

```
alunos = ['Andre', 'Lucas', 'Antonio', 'Maria']
```

```
len (alunos) #tamanho do vetor alunos-4
```

```
append() #método responsável por adicionar um novo elemento na próxima posição do vetor
```

Concatenar valores em vetor

```
lista = [1,2,3]
```

```
lista = lista + [4]
```

```
print (lista)
```

```
lista = lista + [4,5,6]
```

```
print (lista)
```

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Mais algumas funções aplicados sobre um vetor de dados:

Multiplicação de vetores (duplicar conteúdo do vetor):

```
lista = [1,2,3]
```

```
lista * 3          #repete 3x o conteúdo da lista
```

Utilizado para inicializar um vetor com zero. Ex.:

```
tamanho = 10
```

```
lista = [0]
```

```
lista * tamanho
```

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Exemplos práticos. Armazenar os valores em vetor:

- 1) Escreva um código Python que, dados 2 números diferentes (a e b), encontre o menor e o maior.
- 2) Leia um número e imprima a raiz quadrada do número caso o número seja positivo ou igual a zero.
- 3) Leia dois números e efetue a adição. Caso o valor somado seja maior que 20, este deverá ser apresentado somando-se a ele mais 8; caso o valor somado seja menor ou igual a 20, este deverá ser apresentado subtraindo-se 5.

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Exemplos práticos. Armazenar os valores em vetor:

4) Faça um algoritmo que leia dois vetores de 3 posições e realize a soma dos valores nas mesmas posições.

5) Faça um algoritmo que leia 3 números float, armazene em vetor e calcule a soma dos valores, a média dos valores

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Matriz = linha x coluna

O Python não possui em sua estrutura (de forma nativa da linguagem) um tipo de dado matriz.

Como resolver?

- Lista de listas
- Biblioteca Numpy (será visto com detalhes)

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Matriz = linha x coluna

Lista de listas

```
>>> m = [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]]
```

Operações: acessar posição, remover item, apagar lista, operações matemáticas.

Estrutura de controle e Estrutura de Repetição

- **Estrutura de controle:**

Exemplos práticos – matriz com lista

1) Faça um código Python para criar uma matriz com 3 linhas e 5 colunas. Inicializar com valores aleatórios (tipo float). Realizar a soma dos valores por linha e por coluna. Mostrar os valores.