Ciência de Dados – Estudos

ESFORÇOS

- Aprender assuntos técnicos

- Aprender a se comunicar

- Se apaixonar pela solução e não pela ferramenta

- Preparando dados, na maior parte do tempo

- Lidar com empresas que não entendem Ciência de Dados

RECOMPENSAS

- Poder trabalhar em qualquer empresa, de qualquer ramo

- Aumento das chances de morar em outro país

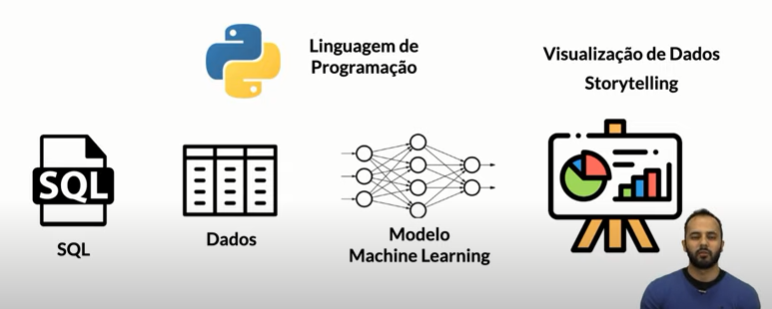
- Ver o impacto do seu trabalho, direto no resultado da empresa

- Reconhecimento

- Bons salários e condições de trabalho



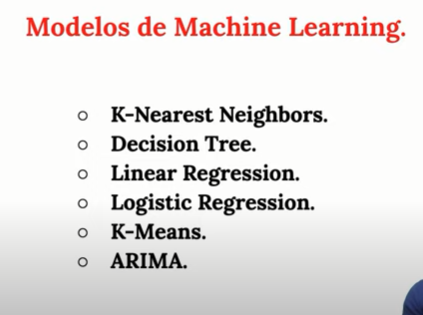
MAPA DE ESTUDO:



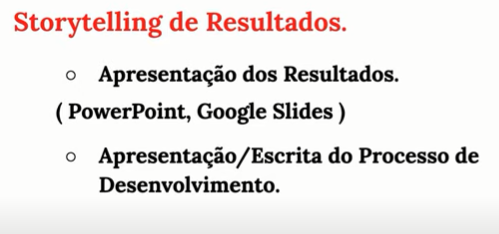
DADOS:



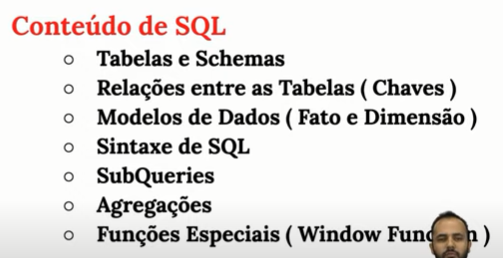
MACHINE LEARNING:



STORYTELLING:



SQL:



LINGUAGEM DE PROGRAMAÇÃO



O interpretador que iremos aprender: PYTHON

Ele já vem instalado no Linux, Mac e outros, mas não vem instalado no Windows. Tem que baixar.

Abrir o prompt de comando do Windows: Windows + r

Digita cmd + enter para entrar no prompt de comando

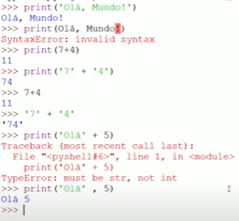
Iremos trabalhar com o idle que vem junto com o python pois é mais colorido (por ora só vemos essa diferença!)

COMANDOS

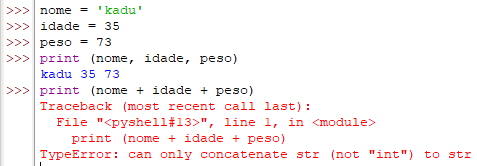
Todos os comandos são funções, e todas as funções têm parênteses.

- print 🡪 escreve na tela o que desejar, entre aspas. ex.: print (‘olá, mundo!’)



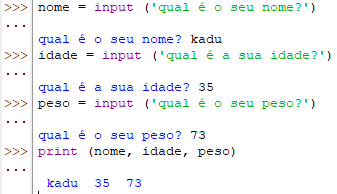


TODA VARIÁVEL É UM OBJETO PRO PYTHON. TODA VARIÁVEL PODE RECEBER VALORES (=)



Modo interativo:

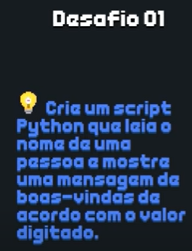
- input 🡪 serve para interagir com o usuário



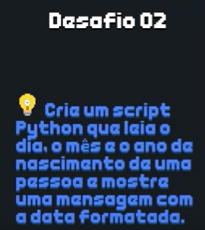
Para executar um projeto, você abre um script (file - new file), faz o projeto, salva e coloca (run – run modele) ou (F5)

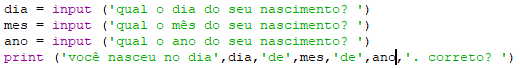
O modo interativo serve para você testar alguma coisa, e o modo de script serve para você realmente criar um programa

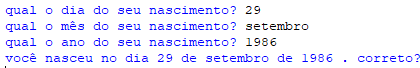
Desafios e soluções da aula 4:









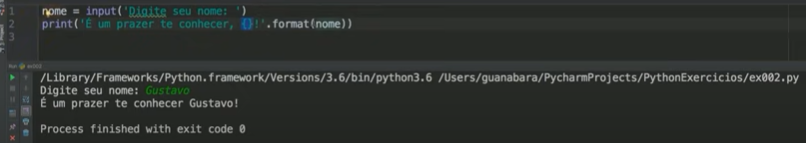


.format

É uma nova forma de usar print

Exemplo:

print(‘é um prazer te conhecer, {}!’.format(nome))



TIPOS PRIMITIVOS

int () : considerar um número inteiro (7, -4, 0, 9875...)

float () : números reais ou números com pontos flutuantes (4.5, 0.076, 7.0...)

bool () : valores lógicos ou booleanos (True or False)

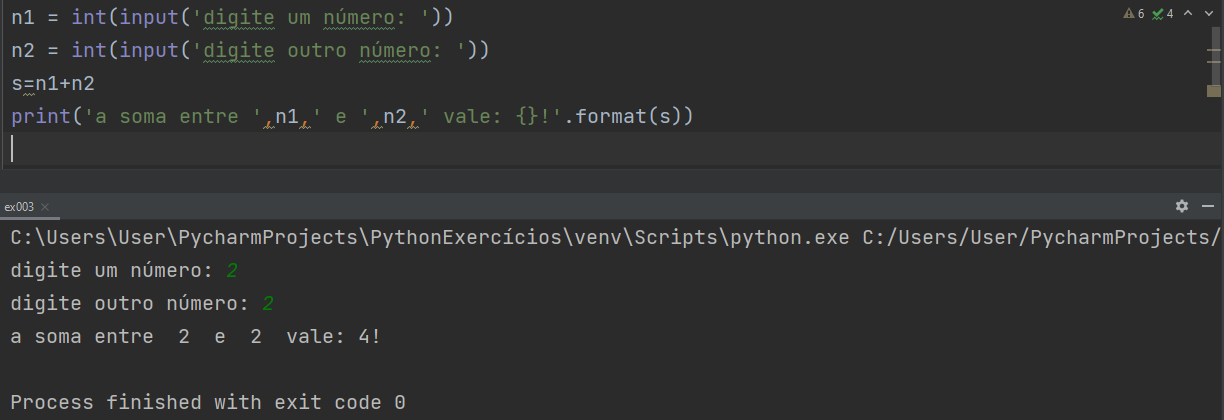
str () : valores caracteres ou stringes (‘olá’, ‘7.5’, ‘’)

print(type(n1))

pra saber o tipo que o programa está considerando

int ()

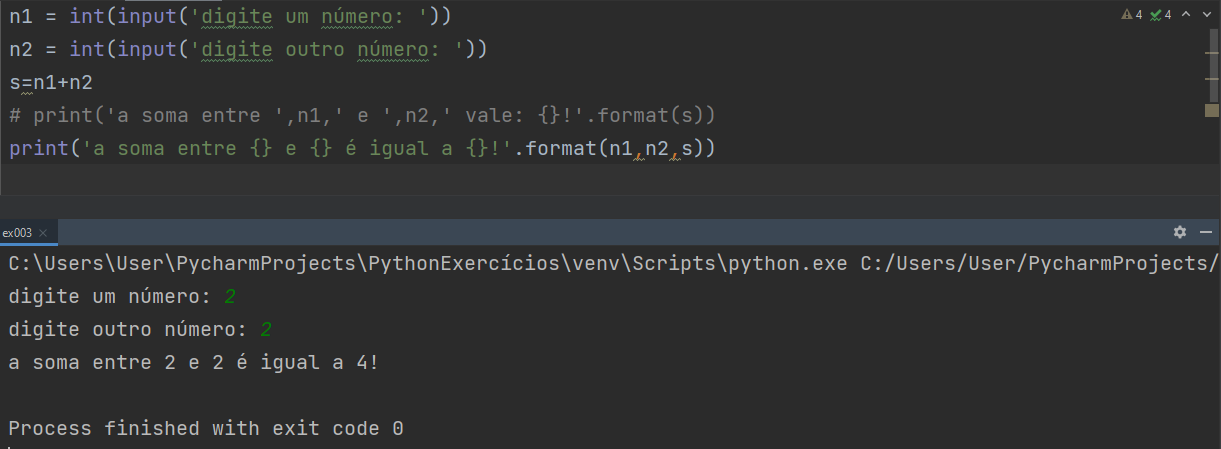
para mudar todo o número considerado para inteiro:



#

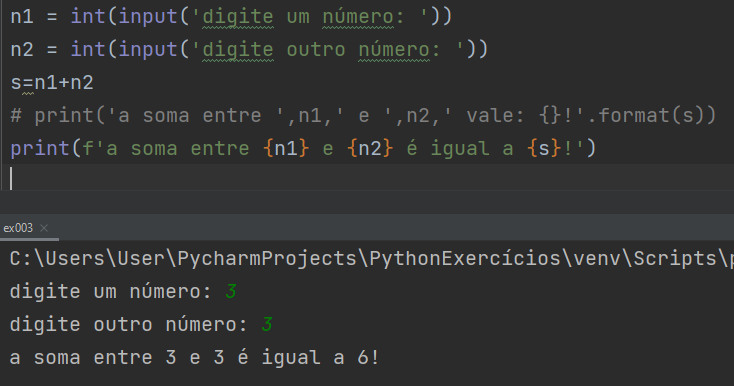
Ignora a linha completa

Exemplo usando o .format (formato novo do Python 3)



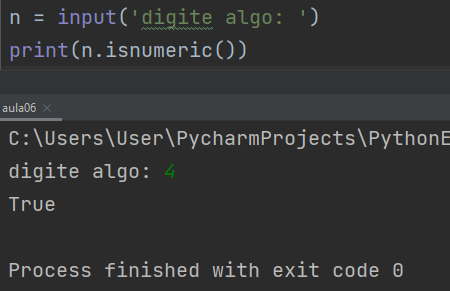
à partir do Python 3.7 também podemos usar o comando da forma f’’

(Na nova atualização do python, você nao precisa mais colocar .format(), da muito trabalho. Você agora só chama o f dentro print fora das aspas e coloca as chaves no lugares definitivos que você deseja. Por exemplo: print(f 'a soma entre {n1} e {n2} é igual a {s}'), o f no comecinho já puxa as informações que você deseja integrar na hora de imprimir o resultado)

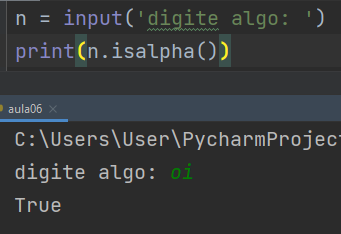


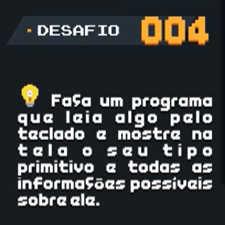
Exemplos de comandos de métodos de testes de tipo

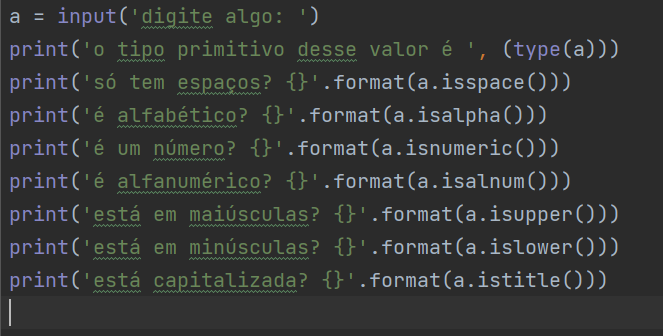
n.isnumeric

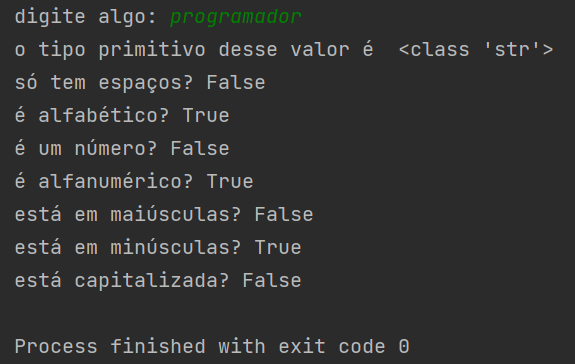


n.isalpha









OPERAÇÕES ARITMÉTICAS

(podem ser números ou variáveis que contém números)

(são operadores binários, precisam de 2 operandos)

+ ADIÇÃO

- SUBTRAÇÃO

\* MULTIPLICAÇÃO

/ DIVISÃO

\*\* POTÊNCIA (ou pow(4,2) por exemplo)

// DIVISÃO INTEIRA

% RESTO DA DIVISÃO

== É IGUAL (PORQUE APENAS 1 = SIGNIFICA ‘RECEBE’)

Ordem de precedência:

1. ()
2. \*\*
3. \*, /, //, %
4. +, -

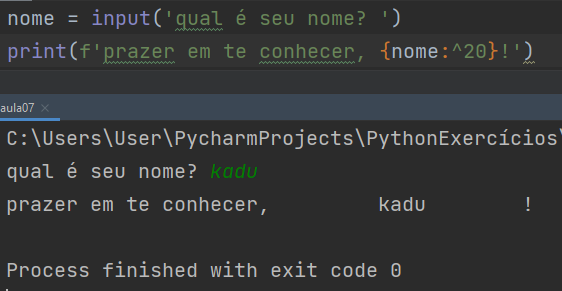
Dicas:

RAIZ QUADRADA n\*\*(1/2)

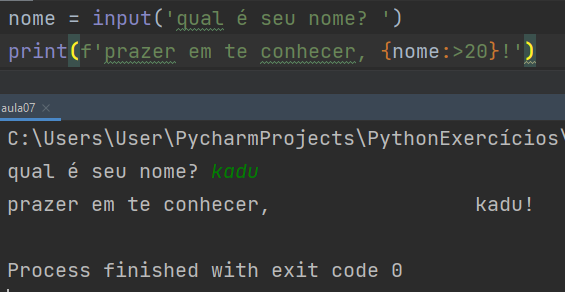
RAIZ CÚBICA n\*\*(1/3)

Uso dos símbolos <>^ para posicionamento:

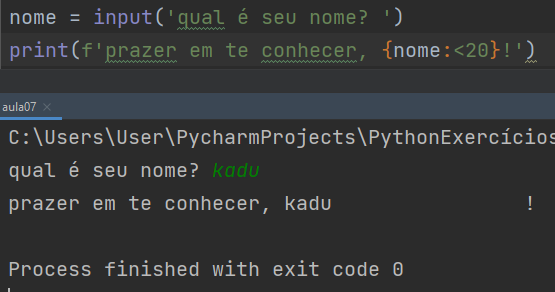
Alinhado ao centro (^):



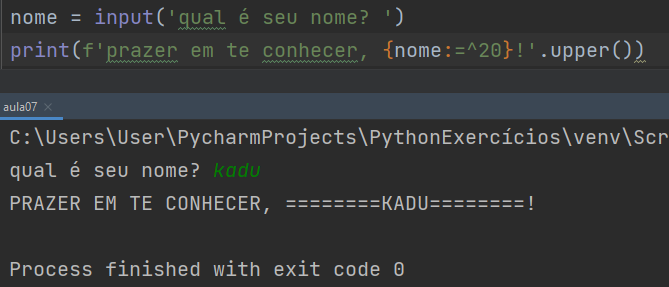
Alinhado à direita (>):



Alinhado à esquerda (<):

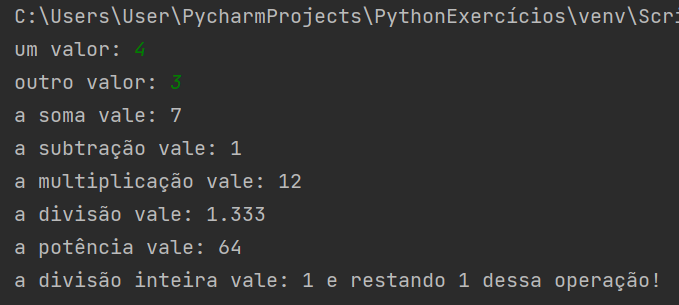


Exemplo aleatório centralizado com o sinal de igual e com a função upper():

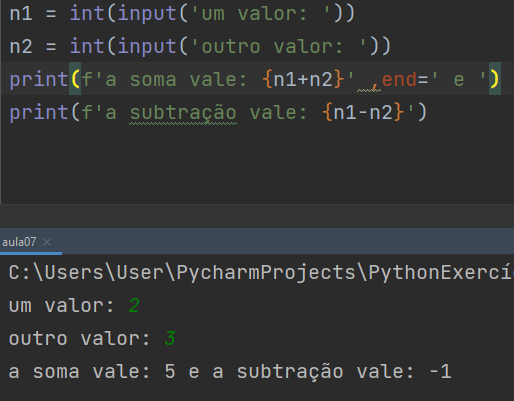


Exemplo aleatório:

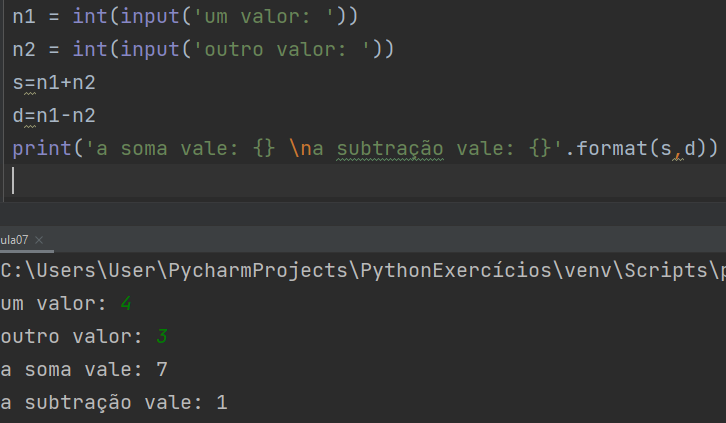


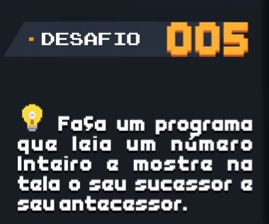


Se não quiser quebrar a linha de um print para o outro (,end=’’) no final



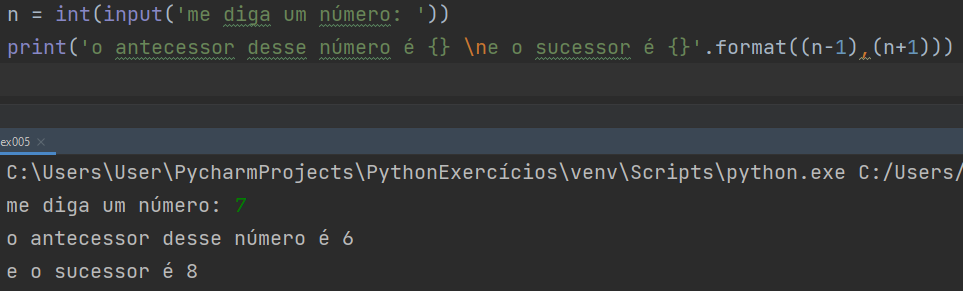
Se quiser quebrar a linha no meio do print (\n) no local onde quer

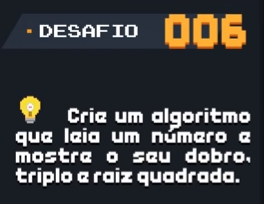


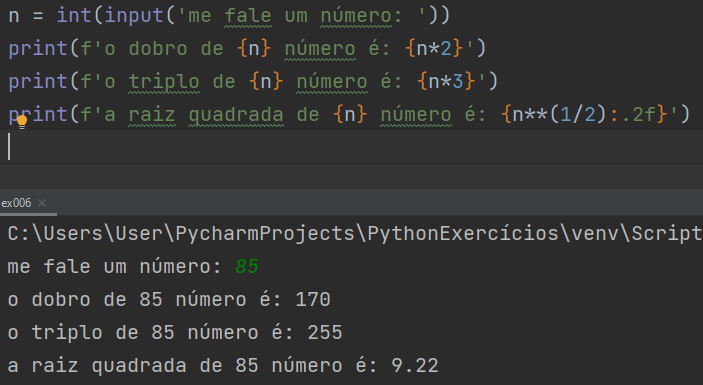


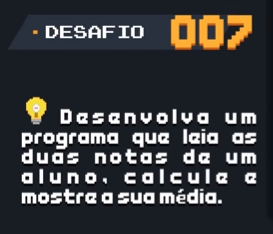


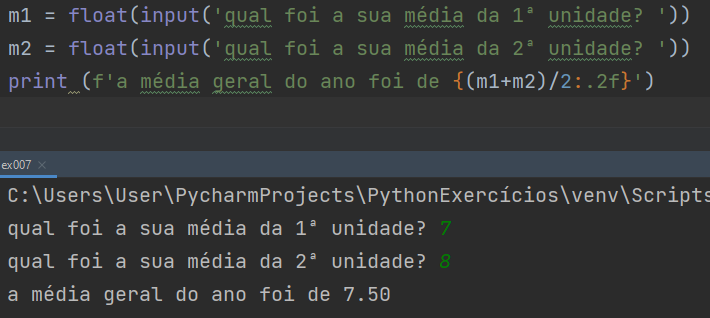
Pode-se fazer assim para economizar memória. Mas se você for precisar na frente fazer uma operação com as variáveis a ou s, melhor deixa-las.

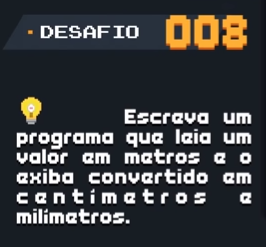


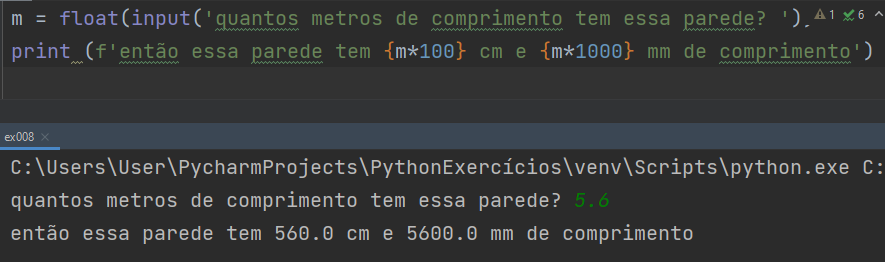




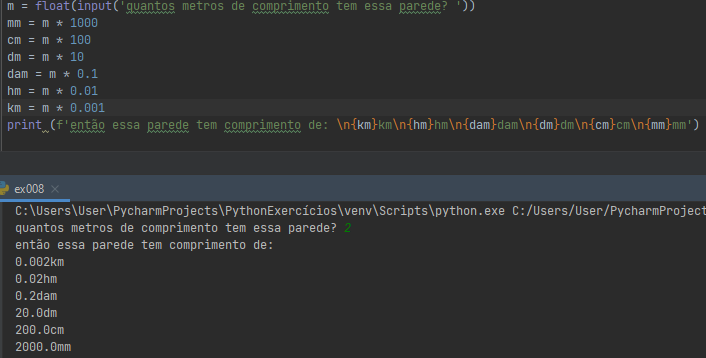


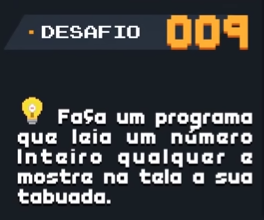


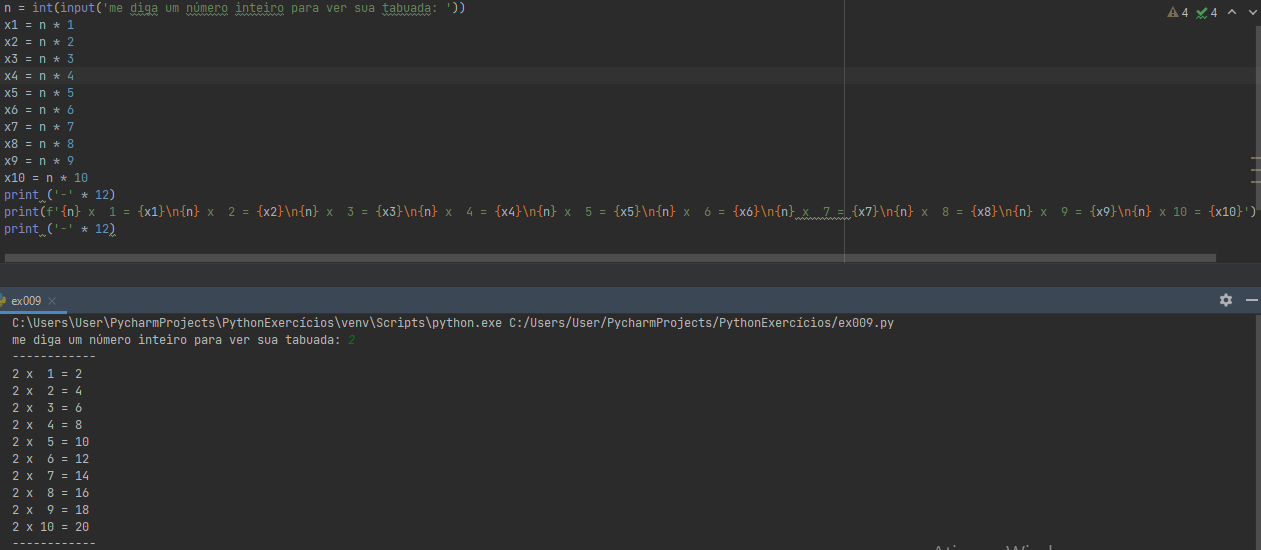


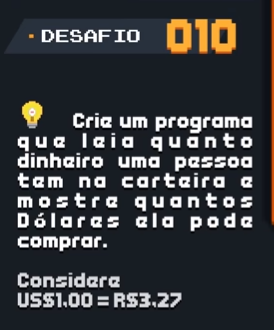


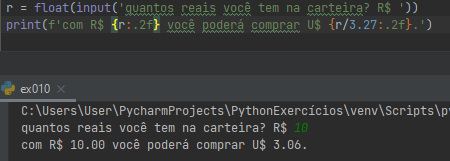
Aprofundamento do desafio feito na aula:

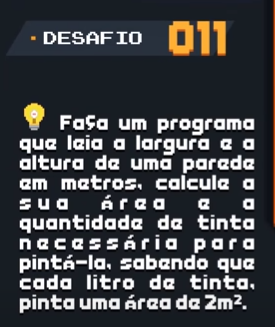


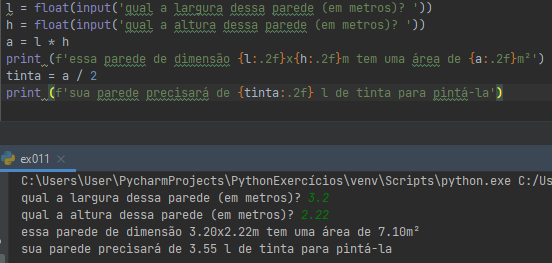


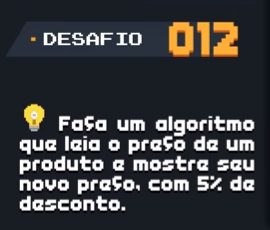


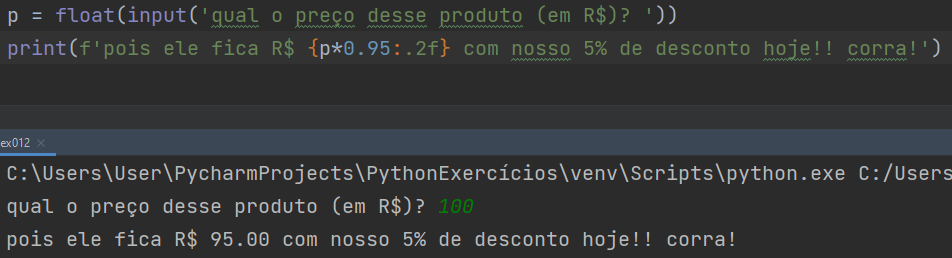




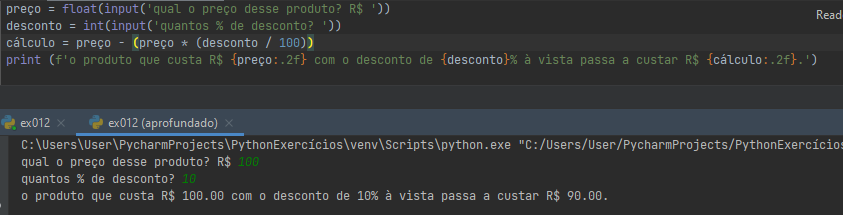




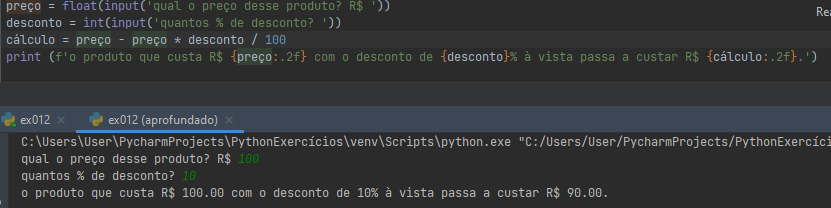


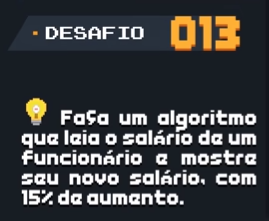


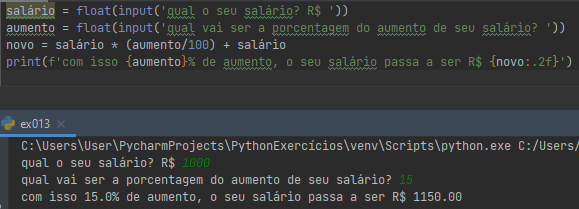
Podemos fazer também um programa que leia o desconto:

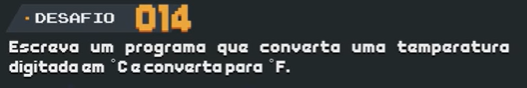


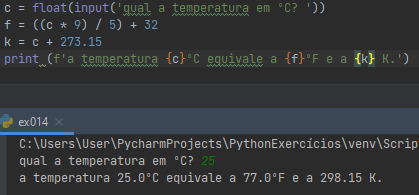
Forma alternativa e mais rápida de escrever porcentagem:

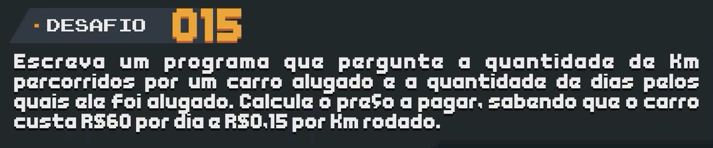


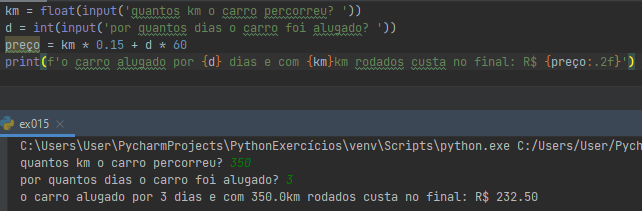












UTILIZANDO MÓDULOS

- IMPORTAR MÓDULOS



Biblioteca ‘math’

- ceil : arredonda o número pra cima

- floor : arredonda o número pra baixo

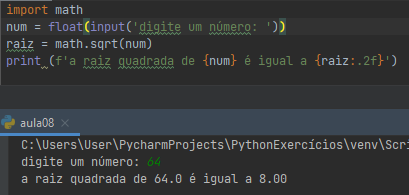
- trunc : eliminar da viga pra frente

- pow : calcula potência (power)

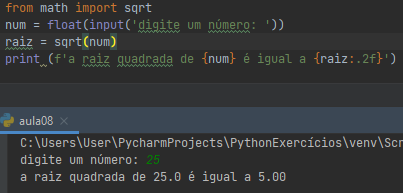
- sqrt : calcula raiz quadrada (square root)

- factorial : números fatoriais

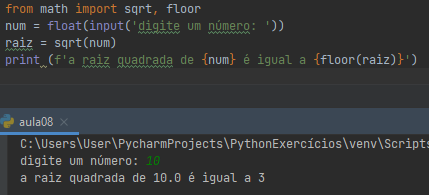
Exemplo com importação do comando ‘sqrt’ da biblioteca ‘math’:



Exemplo com importação específica do comando ‘sqrt’ da biblioteca ‘math’:



Exemplo com importação específica dos comandos ‘sqrt’ e ‘floor’ da biblioteca ‘math’:



\*Como verificar a versão do seu Python?



Vai na opção ‘Python Console’ no PyCharm

Exemplo com importação específica do comando ‘randint’ da biblioteca ‘random’:



Acessar uma biblioteca que não vem com o Python:

- Escreve o nome dela

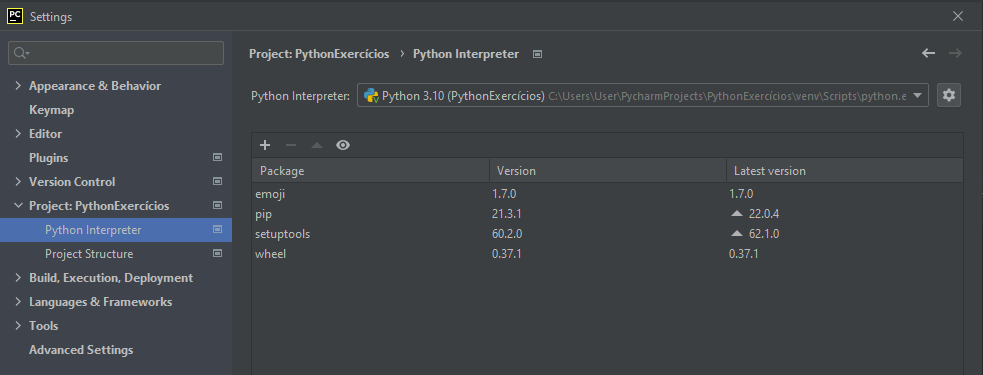
- Clica na lâmpada vermelha que aparece

- Instala



Para ver quais módulos externos foram instalados no seu arquivo:

File – Settings – Project: Python Exercícios – Python Interpretor

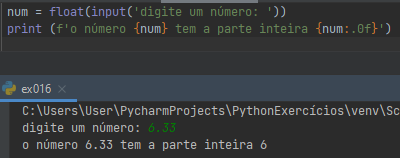


Outra forma de importar bibliotecas é escrever ‘import’ e apertar ctrl+espaço:

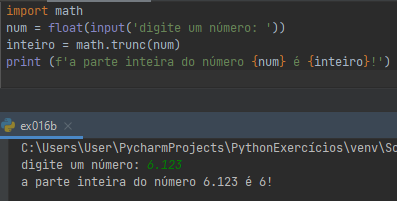




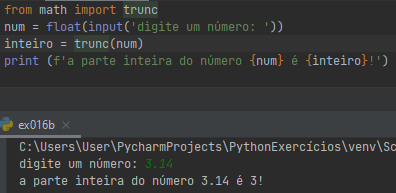
Maneira sem módulos importados:

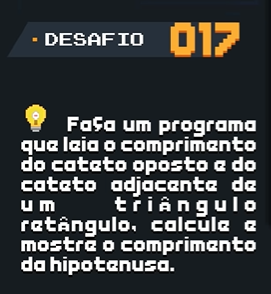


Maneira com módulo ‘math’ importado:

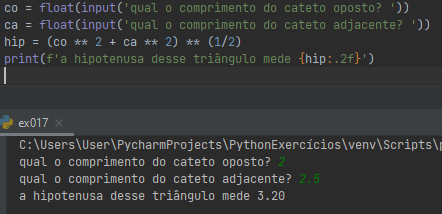


Maneira importando apenas a função ‘trunc’ do módulo ‘math’ para gastar menos memória:

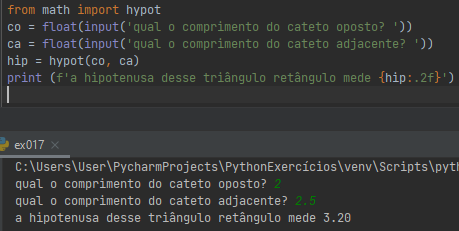


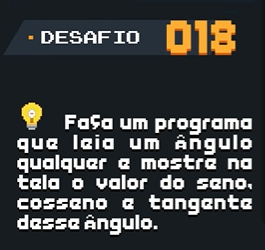


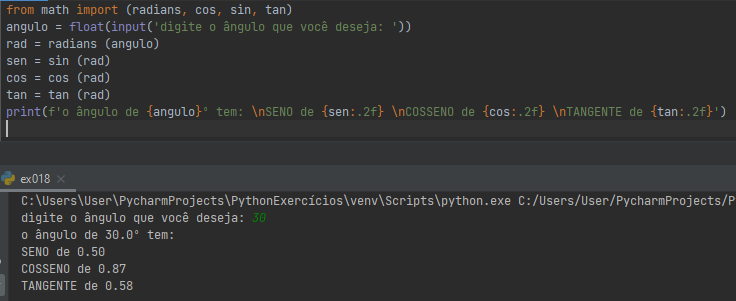
Maneira sem importar módulos:

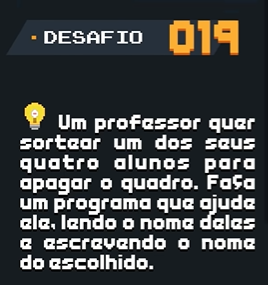


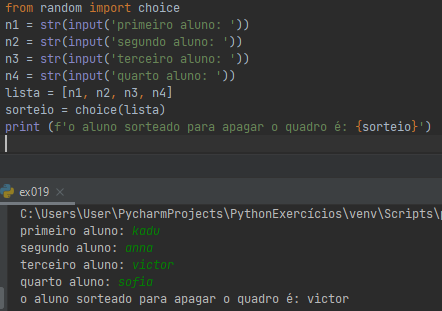
Maneira importando a função ‘hypot’ do módulo ‘math’:

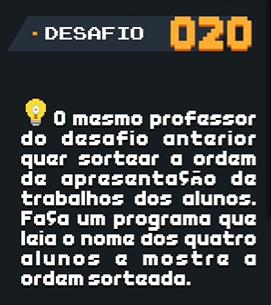




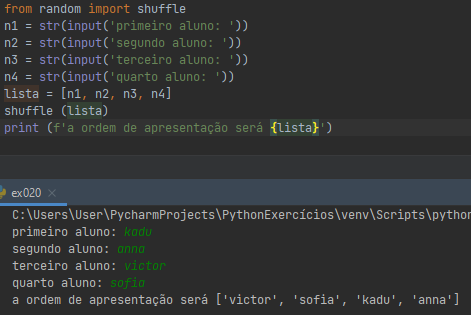




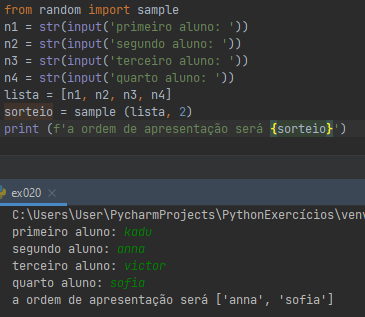




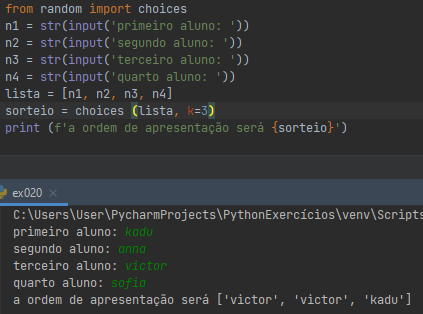
Fazendo com o comando ‘shuffle’ que mistura os nomes na lista:

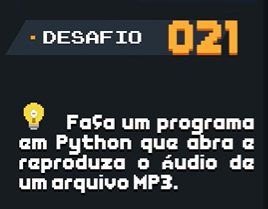


Fazendo com o comando ‘sample’ que você pode dar um limite de quantos nomes ele vai sortear:



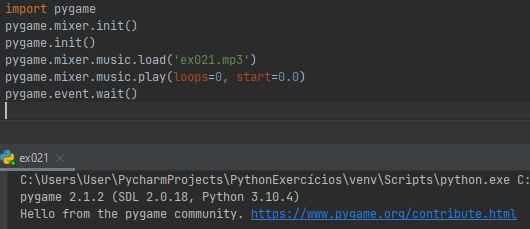
Fazendo com o comando ‘choices’ que pode dar o limite e os nomes podem ser sorteados repetidos:



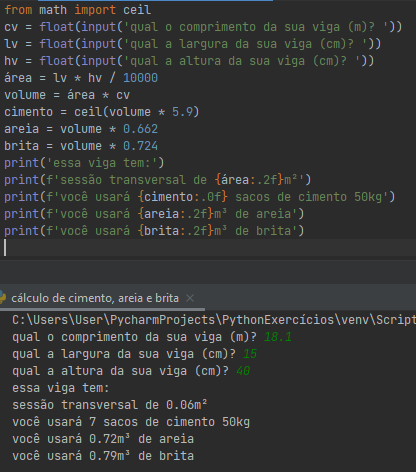


Primeiro tem que copiar e colar a música junto com os arquivos da pasta

Agora o código atual para executar através do módulo ‘pygame’ é:



Desafio que fiz que diz a quantidade de cimentos, areia e brita de determinada viga.



MANIPULAÇÃO E CADEIA DE TEXTOS

FATIAMENTO DE STRINGS

- frase[9]

\*colchete é lista.



- frase[9:13]



- frase [9:21]



- frase [9:21:2]



- frase[:5] é o mesmo que frase[0:5]



- frase[15:]



- frase[9::3]



PARA ANALISAR UM STRING

- len(frase)

Len: comprimento



- frase.count(‘o’): conta quantas vezes aparece a letra ‘o’ minúscula, por exemplo



- frase.count(‘o’,0,13): conta quantas vezes aparece a letra ‘o’ minúscula, já com um fatiamento



- frase.find(‘deo’)



- frase.find(‘android’): se colocar dentro do ‘find’ um string que não existe, aparecerá ‘-1’ e significa que ele não encontrou

- ‘curso’ in frase : ele verifica se existe, True or False

TRANSFORMAÇÃO DE STRINGS

- frase.replace(‘Python’,’Android’): substitui uma string por outra, não importando o tamanho.

- frase.upper(): transforma todas as strings em maiúsculas



- frase.lower(): transforma todas as strings em minúsculas



- frase.capitalize(): deixa somente o primeiro caractere em maiúsculo



- frase.title(): todas as primeiras letras das palavras em maiúsculo



- frase.strip(): remove todos os espaços inúteis no início e no final dos espaços das strings



- frase.rstrip(): remove somente os últimos espaços (da direita)



- frase.lstrip(): remove somente os primeiros espaços (da esquerda)



DIVISÃO DE STRINGS

- frase.split(): criar divisão onde tiver espaço, gerando uma lista para cada palavra de uma cadeia de caracteres

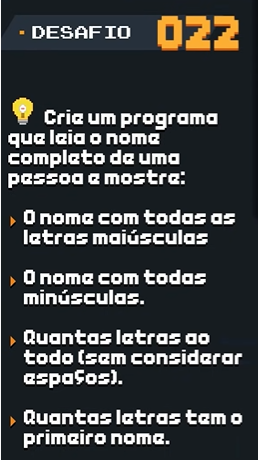
\*dar estudada nessa funcionalidade

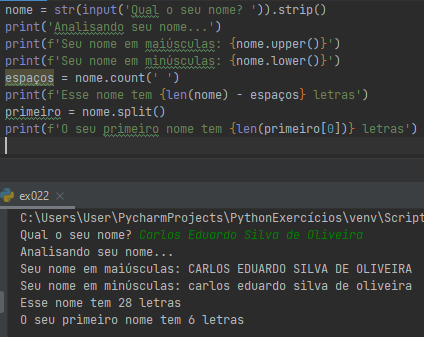


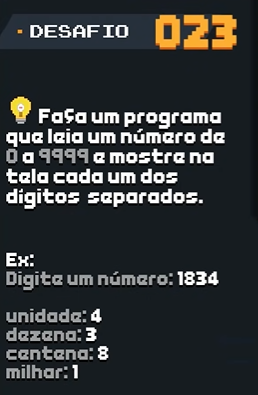


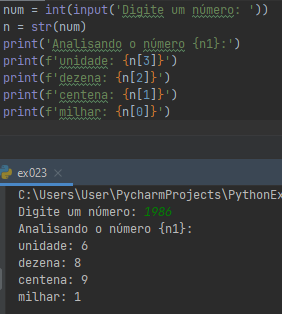
- ‘-‘.join(frase): juntar frases com esse caractere entre aspas











Mas dessa maneira daria erro se não respondêssemos um número com 4 algarismos. Na frente vai ser ensinado uma maneira de resolver isso, mas por enquanto podemos usar a matemática com o ‘//’ (divisão por inteiro) e ‘%’ resto.

