

Python

Introdução | Instalando Python

► <https://www.python.org/>

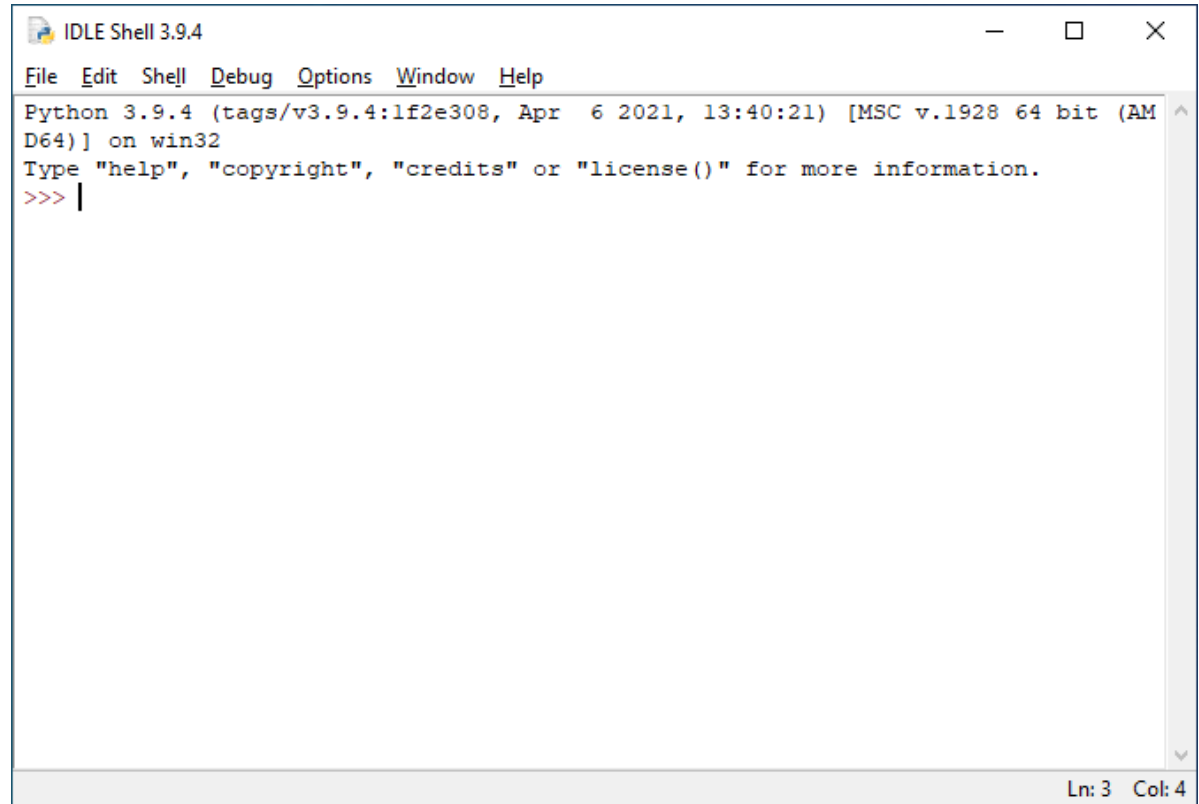


► <https://replit.com/>

**Code, create, and
learn together**

Use our free, collaborative, in-browser IDE to code in 50+ languages — without spending a second on setup.

[<> Start coding](#)

A screenshot of the IDLE Shell 3.9.4 window. The title bar says "IDLE Shell 3.9.4". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the Python 3.9.4 startup message: "Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit (AMD64)] on win32" followed by "Type 'help', 'copyright', 'credits' or 'license()' for more information." and a prompt ">>> |". The status bar at the bottom right shows "Ln: 3 Col: 4".

```
IDLE Shell 3.9.4
File Edit Shell Debug Options Window Help
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
Ln: 3 Col: 4
```

Python

Python x C

- ▶ Procurar por interpreted:

- ▶ <https://docs.python.org/3/glossary.html>

- ▶ Stack Overflow:

- ▶ <https://stackoverflow.com/questions/6889747/is-python-interpreted-or-compiled-or-both>

Python

Introdução | Primeiros Comandos

► help()

- <https://docs.python.org/3.9/tutorial/>
- <https://docs.python.org/pt-br/3.9/index.html>

```
>>> help()
```

```
Welcome to Python 3.9's help utility!
```

```
If this is your first time using Python, you should definitely check out the tutorial on the Internet at https://docs.python.org/3.9/tutorial/.
```

```
Enter the name of any module, keyword, or topic to get help on writing Python programs and using Python modules. To quit this help utility and return to the interpreter, just type "quit".
```

```
To get a list of available modules, keywords, symbols, or topics, type "modules", "keywords", "symbols", or "topics". Each module also comes with a one-line summary of what it does; to list the modules whose name or summary contain a given string such as "spam", type "modules spam".
```

```
help> print
```

```
Help on built-in function print in module builtins:
```

```
print(...)  
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

```
Prints the values to a stream, or to sys.stdout by default.
```

```
Optional keyword arguments:
```

```
file: a file-like object (stream); defaults to the current sys.stdout.
```

```
sep: string inserted between values, default a space.
```

```
end: string appended after the last value, default a newline.
```

```
flush: whether to forcibly flush the stream.
```

```
print("Engineering", "Technical", "School", 60, "Anos", sep="-")
```

Python

Introdução | Variáveis

```
area = "ETS"  
idade = 60  
print(area, "POSSUI", idade, "ANOS DE TRADIÇÃO!")
```

```
type(area)  
type(idade)
```

```
preco = 99.99  
v_f = True
```

Python

Introdução | Tipagem de Dados

► Tipagem Dinâmica

```
area = "ETS"  
type(area)
```

```
<class 'str'>
```

```
pais = 644  
type(pais)
```

```
<class 'int'>
```

Python

Introdução | Snake_Case / CamelCase

- ▶ Padrão Snake_Case para definição de nomes de variáveis:

```
area_bosch = "Engineering Technical School"
```



- ▶ Padrão CamelCase para definição de nomes de variáveis:

```
areaBosch = "Engineering Technical School"
```

Python

Introdução | Pycharm



Versão: 2021.1

Build: 211.6693.115

6 de abril de 2021

[Requisitos do sistema](#)

[Instruções de instalação](#)

[Outras versões](#)

Baixar PyCharm

[Windows](#)

[macOS](#)

[Linux](#)

Professional

Para desenvolvimento Web com Python e desenvolvimento científico. Com suporte para HTML, JS e SQL.

Baixar

Avaliação gratuita

Community

Para o autêntico desenvolvimento Python

Baixar

Open source gratuito

Python

Introdução | Comando Se / Senão / Senão Se

O código funciona?

```
def main():  
    idade = input("ENTRE COM SUA IDADE: ")  
    if (idade == 18):  
        print("PODE TIRAR SUA CNH!")  
    else:  
        print("NÃO PODE TIRAR CNH AINDA")  
  
if __name__ == "__main__":  
    main()
```


Python

Introdução | Comando Se / Senão / Senão Se

E esse?

```
def main():  
    print("ENTRE COM ANO, MÊS E DIA DE NASCIMENTO, RESPECTIVAMENTE: ")  
    ano = input("ANO: ")  
    mes = input("MÊS: ")  
    dia = input("DIA: ")  
    ano_dif = 2022 - ano  
    mes_dif = 2 - mes  
    dia_dif = 18 - dia  
    idade = 2022 - ano  
    if (idade > 18):  
        print("PODE TIRAR SUA CNH!")  
    elif (idade < 18):  
        print("NÃO PODE TIRAR CNH AINDA")  
    else:  
        print("CONTINUA....")
```

Python

Laços de Repetição | While

```
import time

def main():
    contagem_regressiva = 10
    while (contagem_regressiva > -1):
        print(contagem_regressiva)
        contagem_regressiva = contagem_regressiva - 1
        time.sleep(1)
        print("CONTAGEM FINALIZADA!!!")

if __name__ == "__main__":
    main()
```

Python

Laços de Repetição | While – Quebra de Laços

```
import time

def main():
    contagem_regressiva = 10
    while (contagem_regressiva > -1):
        print(contagem_regressiva)
        contagem_regressiva = contagem_regressiva - 1
        time.sleep(1)
        if (contagem_regressiva == 5):
            break
    print("CONTAGEM FINALIZADA!!!")

if __name__ == "__main__":
    main()
```

Python

Python | Analisando os Comandos If() e While()

O que o if() e o while() tem em comum?

- Ambos possuem uma condição de entrada.

- Ambos servem para testar uma condição e executar um bloco uma única vez.

- Ambos possuem uma condição de saída.

- Ambos servem para testar uma condição e executar um bloco enquanto a condição for verdadeira.

Python

Laço de Repetição | For

► range()

► <https://docs.python.org/3/library/stdtypes.html#typeseq-range>

```
temperaturas = [29, 29, 30, 27, 31, 30, 20, 23, 29, 30]
for pos in range(10):
    print(temperaturas[pos])
for pos in range(0, 10):
    print(temperaturas[pos])
for pos in range(0, 10, 1):
    print(temperaturas[pos])
```

► Não é necessário utilizar somente a função range():

```
for temperatura in temperaturas:
    print(temperatura)
```

Python

Laço de Repetição | For | Exercícios

- Transcreva o código abaixo utilizando um laço de repetição for():

```
contador = 1
while(contador <= 10):
    print(contador)
    contador = contador + 1
```

- E o código abaixo?

```
contador = 1
while(contador <= 10):
    print(contador)
    contador = contador + 3
```

Python

Laço de Repetição | Parando/Continuando um Laço

- ▶ Para freiar um laço de repetição quando o usuário acertar o número utilizamos o comando break!
- ▶ Para continuar um laço de repetição para a próxima iteração, utilizamos o comando continue:

```
for temperatura in temperaturas:  
    if (temperatura == 30):  
        continue  
    print(temperatura)  
    if (temperatura == 31):  
        break
```

Python

Print | Format | Outra forma de imprimir textos

► Formato 1: `area="ETS"`
`idade=62`
`print(area, "POSSUI", idade, "ANOS DE TRADIÇÃO!")`

► Formato 2:

```
print("{} POSSUI {} ANOS DE TRADIÇÃO!".format(area, idade))
```

► <https://docs.python.org/3/library/functions.html>

► <https://docs.python.org/3.9/tutorial/inputoutput.html#fancier-output-formatting>

► Acessar:

► <https://docs.python.org/3/library/string.html#formatexamples>

Python

Print | Format | Outra forma de imprimir textos

- Como imprimir a frase abaixo utilizando o a nova configuração de print apresentada?

```
dia_inicio_ferias = 17
```

```
dia_final_ferias = 31
```

```
mes = "Dezembro"
```

```
ano = 2021
```

```
#"AS FÉRIAS SERÃO RETIRADAS ENTRE O DIA XX E O DIA YY DO MÊS DE ZZ DO ANO DE AA"
```

```
print("AS FÉRIAS SERÃO RETIRADAS ENTRE O DIA {} E O DIA {} DO MÊS DE {} DO ANO DE {}".format(dia_inicio_ferias, dia_final_ferias, mes, ano))
```

Python

Print | Format

► Acessar para seguir os exemplos:

► <https://docs.python.org/3/library/string.html#formatexamples>

► Ordem dos parâmetros:

```
numero_1 = 10
numero_2 = 20
print("ESSE É O SEGUNDO PARÂMETRO: {1}.
      ESSE É O PRIMEIRO: {0}.".format(numero_1, numero_2))
```

► Formatação de Floats:

```
peso = 70.12
print("O JOÃO PESA {:.06.2f} Kg.".format(peso))
```

Python

Print | Format

► Formatação de Inteiros:

```
dia = 1
```

```
mes = 1
```

```
ano = 2000
```

```
print("DATA DE HOJE {:02d}/{:02d}/{:d}".format(dia, mes, ano))
```

► Diferença de interpolação entre Python 2 e Python 3:

► <https://pyformat.info/>

Python

Print | Format

► Formatted String Literals

```
empresa = 'Robert Bosch Ltda.'  
print(f'TRABALHO NA {empresa}')
```

► Também pode ser utilizada com funções e métodos:

```
empresa = 'Robert Bosch Ltda.'  
print(f'TRABALHO NA {empresa.lower()}')
```

Python

Print | Format | Exercícios

- A Bosch usa um padrão de banco de dados onde o sobrenome aparece primeiro que o nome no sistema/banco de dados. Ex.: Augusto Cleber. Como adaptar um programa para que, ao exibir uma mensagem para mim, meu nome apareça primeiro que o sobrenome?

```
sobrenome = "Augusto"
nome = "Cleber"
print("OLÁ SR. {} {}".format(sobrenome, nome))
```

- O que será impresso pelos seguintes comandos?

```
print("R$ {:.1f}".format(10000.00))
print("R$ {:010.2f}".format(1.11))
```

Python

Gerando Números Aleatórios

- ▶ Não é uma função interna, precisamos importar a biblioteca:

```
import random  
  
print(random.random())
```

- ▶ <https://docs.python.org/3/library/random.html>

Python

Gerando Números Pseudo Aleatórios

► <https://docs.python.org/pt-br/3.7/library/random.html>

► Usando seed:

```
import random  
  
random.seed(123)  
  
print(random.randrange(1, 101))
```

Python

Complementando...

- ▶ Quais são as funções built-in que vimos até então?
 - ▶ <https://docs.python.org/3/library/functions.html>
- ▶ Float Division: /
 - ▶ Ex.: 5 / 2
- ▶ Integer Division: //
 - ▶ Ex.: 5 // 2

Python

Funções

► Snake_Case

```
def nome_da_funcao():  
    #TODO O CÓDIGO IDENTADO FAZ PARTE DA FUNÇÃO  
    print("EIS UMA FUNÇÃO")
```

► Receber parâmetros e retorná-los:

```
def soma(a, b):  
    return a + b
```

```
s = soma(3, 4)
```

Python

Importando Arquivos

► Arquivo a.py

```
if (__name__ == "__main__"):  
    chamar_arquivo_b()
```

► Arquivo b.py:

```
if (__name__ == "__main__"):  
    imprimir_informacoes()
```

Python

Introdução | Exercícios

- ▶ Qual será a saída do comando print?

```
substantivo = "Python"  
verbo = "é"  
adjetivo = "fantástico"  
print(subst, verbo, adjetivo, sep="_", end="!\n")
```

- ▶ a) Python_é_fantástico!
- ▶ b) Python é fantástico!
- ▶ c) Python!é!fantástico_
- ▶ d) Python_é_fantástico

Python

Introdução | Exercícios

► E se fôssemos imprimir datas?

► Formato: DD/MM/YYYY

```
dia = 01  
mes = 01  
ano = 0001  
  
print(dia, mes, ano, sep="/")
```

Python

Introdução | Exercícios

► O código abaixo funciona?

```
num1 = 10
num2 = 10
if(num1 = num2):
    print("SÃO IGUAIS")
```

► E o seguinte código?

```
num1 = 9
num2 = "11"
print(num1 + num2)
```

```
num1 = 9
num2 = "11"
produto = num1 * num2
print(produto)
```

Python

Introdução | Exercícios

► O código abaixo funciona?

```
area = "Engineering Technical School"  
empresa = "Robert Bosch Ltda."  
print(area + empresa)
```

Python

Introdução | Exercícios

- Se eu quiser criar um sistema de acesso de usuário utilizando operadores condicionais, como devo proceder?

```
usuario = input("INFORME O NOME DE USUÁRIO QUE ACESSARÁ O SISTEMA: ")

if(usuario == "auc2ca"):
    print("BEM-VINDO Cleber!")
else(usuario == "ffr9ca"):
    print("BEM-VINDO Francis!")
else(usuario == "ian9ca"):
    print("BEM-VINDO Ianella!")
else:
    print("USUÁRIO NÃO CADASTRADO!")
```

Python

Laço de Repetição | Exercícios

- ▶ Dentro de um laço, qual é a diferença entre break e continue?
- ▶ O que o código abaixo imprime no console?

```
i = 1
while(i <= 7):
    print(i)
    i = i + 1
    if(i == 5):
        break

for i in range(1, 8):
    if(i == 5):
        continue
    print(i)
```


Python

Exercícios

- ▶ Por que importar random()?
- ▶ Qual é o menor e o maior número possível que o script abaixo consegue imprimir ?

```
import random  
  
aleatorio = random.randrange(10)  
  
print(aleatorio)
```

- ▶ Quais códigos abaixo são capazes de gerar números entre 0 e 100 incluindo o 100?

```
int(random.random() * 101);  
round(random.random() * 101);  
round(random.random() * 100);  
random.randrange(0,101)  
random.randrange(1,100)
```

Python

Exercícios

- Analise o código abaixo que define o primeiro grupo que apresentará um trabalho. Ele é justo?

```
import random
sorteado = random.randrange(0,4)
print(sorteado)
if sorteado == 1:
    print("GRUPO 1 INICIA A APRESENTAÇÃO")
elif sorteado == 2:
    print("GRUPO 2 INICIA A APRESENTAÇÃO")
else:
    print("GRUPO 3 INICIA A APRESENTAÇÃO")
```

Python

Exercícios

► Suponha três arquivos independentes. O que ocorre em cada um dos casos?

```
# arquivo a.py
print("Executando a")
```

```
# arquivo b.py
print("Executando b")
```

```
# principal.py
import a
import b
```

```
# arquivo a.py
def executa():
    print("Executando a")
```

```
# arquivo b.py
def executa():
    print("Executando b")
```

```
# principal.py
import a
import b
```

Python

Exercícios

- Suponha que você queira executar o módulo independente:

```
#ARQUIVO a.py
def executa():
    print("EXECUTA A")
```

- Como proceder?

```
def executa():
    print("Executando a")
if(__name__ == "__main__"):
    executa()
```

```
def executa():
    print("Executando a")
if(__name__ == "__main__"):
    executa()
```

```
def executa():
    print("Executando a")
if(__name__ == "__main__"):
    executa()
```

```
def executa():
    print("Executando a")
if(_name__ == "__main__"):
    executa()
```