

















FIAP

Aula 3

COMPUTATIONAL THINKING USING PYTHON





Introdução ao Python



AGENDA

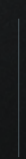


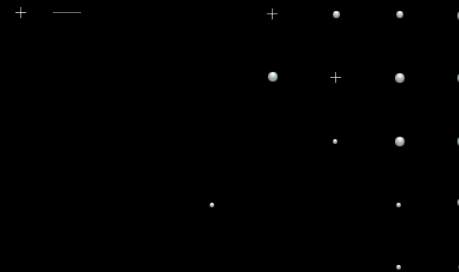
1. Apresentação do ambiente de desenvolvimento (IDLE, REPL, vscode)
2. O que é Python?
3. Variáveis e tipos de dados
4. Lendo entrada do teclado



Introdução

+





“Python é uma das linguagens de programação mais populares e em crescimento mais rápido do mundo... utilizada para diversas tarefas, incluindo programação Web e Análise de Dados

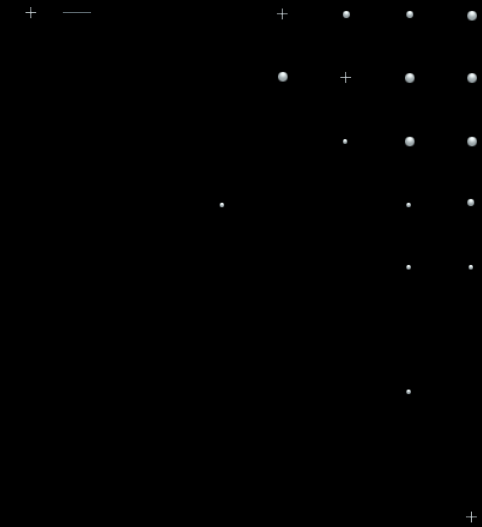


O que é Python?



LINGUAGEM PYTHON

- Linguagem de programação
- Criada no início da década de 1990
- Ampla variedade de uso:
 - automatização de tarefas
 - aplicativos Web
 - criação de modelos de *Machine Learning*
 - redes neurais
 - ciencia de dados
- Possui sintaxe avançada e de fácil entendimento
- Variedade de pacotes (bibliotecas) *open-source*
- Programas escritos em Python executam em diferentes plataformas (Windows, macOS e distribuições de Linux)

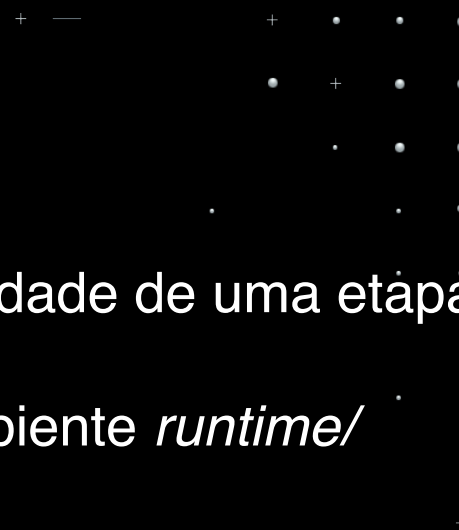




Executando um código em Python



LINGUAGEM PYTHON



- Python é uma linguagem **interpretada**
- Reduz o ciclo de edição/teste/depuração - não há necessidade de uma etapa de **compilação**
- Para executar um código em Python, é preciso de um ambiente *runtime/* *intepretador*
- Ambientes *runtime*:
 - modo **interativo**: cada comando é interpretado imediatamente
 - modo de **script**: arquivo texto com uma extensão .py

O programa é executado linha a linha e a saída é exibida

+ -
• • • •
• + • •



https://www.python.org/



Donate



Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

```
# For loop on range
>>> numbers = range(10)
>>> product = 1
>>> for number in numbers:
...     product *= number
...
>>> print('The product is', product)
The product is 3628800
```

All releases

Source code

Windows

macOS

Other Platforms

License

Alternative Implementations

Download for macOS

Python 3.10.4

Not the OS you are looking for? Python can be used on many operating systems and environments.

[View the full list of downloads.](#)

IDLE

```
IDLE Shell 3.9.4
Python 3.9.4 (v3.9.4:1f2e3088f3, Apr  4 2021, 12:32:44)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

```
Fernando 1 — Python — 107x22
Last login: Fri Mar 25 10:52:25 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[10-43-48-58:~ Fernando$ python 3
/System/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/Python: can't
open file '3': [Errno 2] No such file or directory
[10-43-48-58:~ Fernando$
[10-43-48-58:~ Fernando$
[10-43-48-58:~ Fernando$
[10-43-48-58:~ Fernando$
[10-43-48-58:~ Fernando$
[10-43-48-58:~ Fernando$ python3
Python 3.9.4 (v3.9.4:1f2e3088f3, Apr  4 2021, 12:32:44)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> 2/5
0.4
>>> █
```

Terminal



hands on!

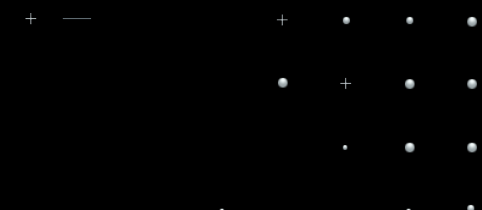




Implementações do Python



IMPLEMENTAÇÕES DO PYTHON



- Python é licenciado sob a licença *open-source* OSI e há várias implementações disponíveis
- Opções:
 - **CPython**¹ - implementação de referência (mais popular)
 - **Anaconda**² - versão especializada para tarefas de programação científicas (ciência de dados e machine learning)
 - **IronPython**³ - implementação de software livre compilada no runtime .NET³
 - **Jupyter Notebook**⁴ - ambiente interativo baseado na Web amplamente utilizados em pesquisas e meio acadêmico para modelagem matemática, aprendizado de máquina, análise estatística e aprendizado de codificação

¹ <https://www.python.org/>

² <https://www.anaconda.com/>

³ <https://ironpython.net/>

⁴ <https://jupyter.org/>



Constantes, Variáveis e tipos de dados básicos em Python



CONSTANTES E LITERAIS

- Constantes são símbolos que representam valores e não podem ser alterados
- Também conhecido como **literals**
- Geralmente utilizados em expressões
- Exemplos:
 - 5
 - -120
 - 3.1415
 - 'Introdução a Linguagem Python'
 - "Olá Pessoal"
 - True
 - False

TIPOS DE CONSTANTES

- Números **inteiros** (`int`) - valores numéricos que não possuem ponto decimal
 - 10, 123456, -35, 0b1011_(base 2), 0o7_(base 8), 0xF_(base 16)
- Números **reais** (`float`) - valores numéricos com ponto decimal
 - 3.55, .15, -23.0, 0.0
- Valores **booleanos** (`bool`) - também conhecidos como valores lógicos
 - verdadeiro (`True`) e falso (`False`)
- **Textos** (`Strings`) - são cadeia de caracteres delimitadas por apóstrofes ou aspas
 - letras, palavras ou frases



Operadores



OPERADORES

- São símbolos pré-definidos que realizam uma operação sobre **um** ou **mais** operandos
- Sempre produz um valor como resultado, dependendo dos operandos ou operadores
- Também são conhecidos como **operadores**



VAMOS PRATICAR!

Resolva as operações a seguir usando apenas lápis, papel e calculadora, em seguida confira os resultados inserindo as operações na *Shell* do Python.

a) `893 // 10`

b) `893 / 10`

c) `25.0 // 2`

d) `25.0 / 2`

e) `5678 % 1`

f) `5678 % 10`

g) `5678 % 100`

h) `5678 % 1000`

i) `5678 // 1`

j) `5678 // 10`

k) `5678 // 100`

l) `5678 // 1000`

m) `123 // 1000`

n) `123 / 1000`

o) `0 / 0`

p) `0 ** 0`

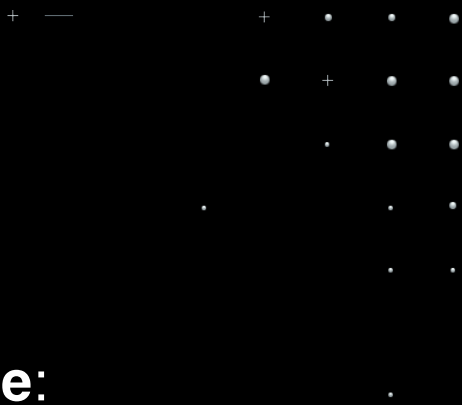
q) `1e3 + 5`

r) `0x10 - 3.5`

s) `9 ** 0.5`

t) `81 ** 0.5`

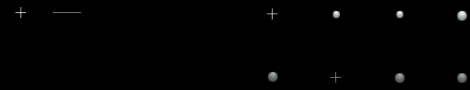
OPERADORES ARITMÉTICOS



Operador	Descrição	Exemplos
+ (binário)	Soma o primeiro operando com o segundo.	$7 + 4 \rightarrow 11$ $1 + 2.0 \rightarrow 3.0$
- (binário)	Subtrai o segundo operando do primeiro.	$15 - 5 \rightarrow 10$ $5 - 15 \rightarrow -10$
+ (unário)	Mantém o sinal do operando à direita. Observação: é a função identidade.	$+ 3 \rightarrow 3$ $+ (-3) \rightarrow -3$
- (unário)	Inverte o sinal do operando à direita.	$-(3) \rightarrow -3$ $-(-7) \rightarrow 7$
*	Multiplica o primeiro operando pelo segundo.	$3 * 8 \rightarrow 24$
/	Quociente da divisão real do primeiro operando pelo segundo.	$9 / 2 \rightarrow 4.5$ $9.0 / 2 \rightarrow 4.5$
//	Quociente da divisão inteira do primeiro operando pelo segundo.	$9 // 2 \rightarrow 4$ $9.0 // 2 \rightarrow 4.0$
%	Resto da divisão inteira do primeiro operando pelo segundo.	$9 \% 2 \rightarrow 1$ $9.0 \% 2 \rightarrow 1.0$
**	Exponenciação, eleva o primeiro operando ao segundo.	$3 ** 2 \rightarrow 9$

Importante:
Exceto pela divisão real, quando todos os operandos envolvidos na operação forem inteiros, o resultado será sempre um inteiro, porém, basta um operando real para que o resultado da operação resulte em um número real

Hands on



VAMOS PRATICAR!

Resolva as operações a seguir usando apenas lápis, papel e calculadora, em seguida confira os resultados inserindo as operações na *Shell* do Python.

- | | | | |
|------------------|-----------------|------------------|-------------------|
| a) $893 // 10$ | b) $893 / 10$ | c) $25.0 // 2$ | d) $25.0 / 2$ |
| e) $5678 \% 1$ | f) $5678 \% 10$ | g) $5678 \% 100$ | h) $5678 \% 1000$ |
| i) $5678 // 1$ | j) $5678 // 10$ | k) $5678 // 100$ | l) $5678 // 1000$ |
| m) $123 // 1000$ | n) $123 / 1000$ | o) $0 / 0$ | p) $0 ** 0$ |
| q) $1e3 + 5$ | r) $0x10 - 3.5$ | s) $9 ** 0.5$ | t) $81 ** 0.5$ |



Variáveis



VARIÁVEIS

- Variáveis são uma das bases fundamentais em programas em Python
- Contém dados na memória
- Possuem nomes e podem ser referenciados por esse nome
- Possuem tipos que especificam o tipo de dados

Python

```
x = 1          # assign variable x the value 1
y = x + 5      # assign variable y the value of x plus 5
z = y          # assign variable z the value of y
```

- Python é uma linguagem *dinamicamente tipada* (o tipo de variável é determinado pelos dados atribuídos a ela)
- Nomes de variáveis diferenciam **maiúsculas** de **minúsculas**

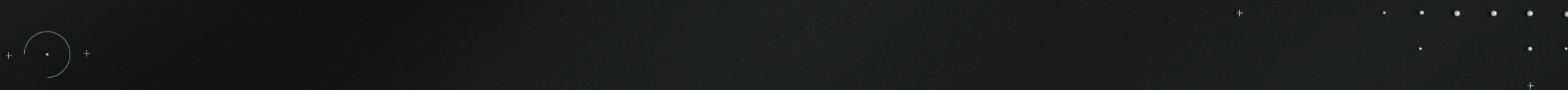
identificador da variável

valor atribuído à variável

minha_idade = 35

operador de atribuição

Lembre-se: uma variável guarda apenas um valor por vez!!!



```
[>>> a = 2
[>>> type(a)
<class 'int'>
[>>> b = 3.1415
[>>> type(b)
<class 'float'>
[>>> c = True
[>>> type(c)
<class 'bool'>
[>>> texto = 'Hello World!'
[>>> type(texto)
<class 'str'>
```

A mesma variável poderia
receber valores de tipos
diferentes?



Identificadores



IDENTIFICADORES

- Nome da variável
- Formado por uma sequência de um ou mais caracteres
- Não pode conter símbolos especiais como &, %, \$, #, @, !
- Não pode iniciar com dígito
- Não pode ser uma palavra reservada
- Python é uma linguagem *case-sensitive*

PEP 8 -- Style Guide for Python Code | [Python.org](https://www.python.org)

PALAVRAS RESERVADAS

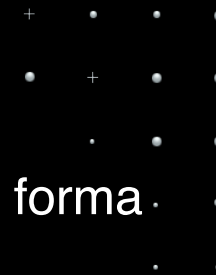


```
[>>> help('keywords')
```

```
Here is a list of the Python keywords.  Enter any keyword to get more help.
```

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

TRABALHANDO COM NÚMEROS



- A maioria dos programas manipula números
- Computadores tratam números **inteiros (int)** e números **decimais (float)** de forma diferente

Python

```
x = 1          # integer
x = 1.0        # decimal (floating point)
```

- A função `type()` interna do Python retorna um tipo de dados

Python

```
x = 1
print(type(x)) # outputs: <class 'int'>

x = 1.0
print(type(x)) # outputs: <class 'float'>
```

TRABALHANDO COM BOOLEANOS

- O tipo de dados booleano contém o valor `true` ou `false`
- Internamente, `bool` é tratado como um tipo especial de inteiro
- Tecnicamente, `true` tem um valor de 1 e `false` tem um valor de 0

Python

```
x = True  
print(type(x)) # outputs: <class 'bool'>
```

- Não são utilizados para cálculos matemáticos

TRABALHANDO COM CADEIAS DE CARACTERES

- Uma cadeia de caracteres é uma coleção de zero ou mais caracteres
- Normalmente são declaradas usando aspas simples, mas as aspas duplas podem ser usadas

Python

```
x = 'This is a string'
print(x) # outputs: This is a string
print(type(x)) # outputs: <class 'str'>
y = "This is also a string"
```

- Cadeias de caracteres podem ser adicionadas a outras cadeias de caracteres ("concatenação")

Python

```
x = 'Hello' + ' ' + 'World!'
print(x) # outputs: Hello World!
```


IMPRIMIR NO CONSOLE

- Uma cadeia de caracteres é uma coleção de zero ou mais caracteres
- Normalmente são declaradas usando aspas simples, mas as aspas duplas podem ser usadas

Python

```
x = 'This is a string'
print(x) # outputs: This is a string
print(type(x)) # outputs: <class 'str'>
y = "This is also a string"
```

- Cadeias de caracteres podem ser adicionadas a outras cadeias de caracteres ("concatenação")

Python

```
x = 'Hello' + ' ' + 'World!'
print(x) # outputs: Hello World!
```

IMPRIMIR NO CONSOLE

- A função integrada `print()`
- Imprime texto e/ou conteúdo de variável na tela

Python

```
print('Hello World!')
```

- Por padrão, a função `print()` produz um caractere de nova linha no final da linha

LENDO ENTRADA DO TECLADO

- Permite a interatividade com o usuário (ou outro sistema)
- Imprime texto e/ou conteúdo de variável na tela

Python

```
name = input('Enter your name:')  
print(name)
```

- Função `input()` sem parâmetros

Python

```
print('What is your name?')  
name = input()  
print(name)
```


LENDO NÚMEROS COMO ENTRADA

- A função `input()` sempre retorna o valor digitado como um texto

Python

```
x = input('Enter a number: ')  
print(type(x))
```

- Transformando em uma variável inteira através da função `int()`

Python

```
x = int(input('Enter a number: '))  
print(type(x))
```

CONVERTENDO NÚMEROS EM CADEIAS DE CARACTERES

- O método `str()` converte um valor inteiro ou ponto flutuante e transformará em uma cadeia de caracteres

Python

```
x = 5  
print('The number is ' + str(x))
```

- A conversão garante que o inteiro, no respectivo formulário de cadeia de caracteres, seja concatenado à cadeia de caracteres à esquerda.



Python é uma linguagem
case-sensitive





Comentários em *Python*



COMENTÁRIOS

- Comentários são úteis para ajudar os programadores que lerão o código
- Recordar a razão de algumas decisões
- São ignorados na execução do programa

```
1 a = 1045.00 # salário mínimo de 2020
2 b = 1100.00 # salário mínimo de 2021
3 c = b - a # aumento do salário mínimo
4 print('O salário mínimo aumentou:', c, 'reais')
```



Python Tutor



AMBIENTE PYTHON TUTOR

- Ambiente gratuito e online para o desenvolvimento de códigos em Python
- Ajuda a entender o que ocorre na memória e saída de dados após a execução das instruções do código-fonte
- Gera representações gráficas relacionadas ao código-fonte e são atualizadas a cada instrução executada
- Pode ser acessado através do endereço <http://pythontutor.com/>



[Get live help](#) for free in the [Python tutoring Discord](#) chat room

Write code in Python 3.6



```
1 a = 1045.00 # salário mínimo de 2020
2 b = 1100.00 # salário mínimo de 2021
3 c = b - a   # aumento do salário mínimo
4 print('O salário mínimo aumentou:', c, 'reais')
5
```

Visualize Execution

Live Programming Mode

[Get live help](#) for free in the [Python tutoring Discord](#) chat room

Python 3.6
([known limitations](#))

```
1 a = 1045.00 # salário mínimo de 2020
2 b = 1100.00 # salário mínimo de 2021
→ 3 c = b - a # aumento do salário mínimo
→ 4 print('O salário mínimo aumentou:', c, 'reais')
```

[Edit this code](#)

→ line that just executed

→ next line to execute

<< First

< Prev

Next >

Last >>

[Customize visualization](#) (NEW!)

Print output (drag lower right corner to resize)

Frames

Objects

Global frame

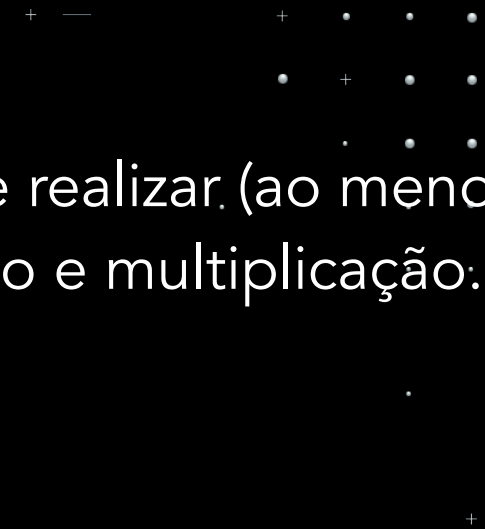
a	1045.0
b	1100.0
c	55.0



Exercícios



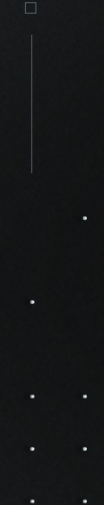
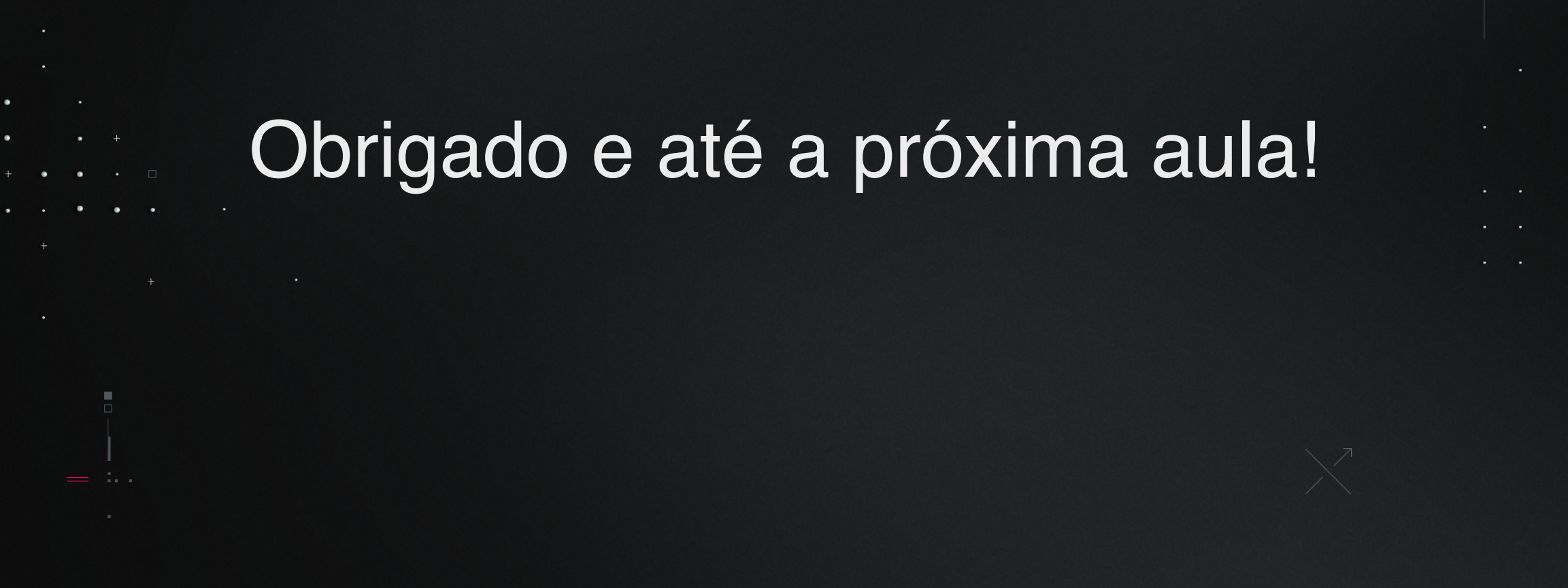
EXERCÍCIO



1. Criar uma calculadora em Python que seja capaz de realizar (ao menos) as quatro operações básicas - adição, subtração, divisão e multiplicação.



FIAP



Obrigado e até a próxima aula!

