

SSC0800 - Introdução à Ciência de Computação I

# Tipos de linguagens e introdução ao Python

Prof.: Leonardo Tórtoro Pereira

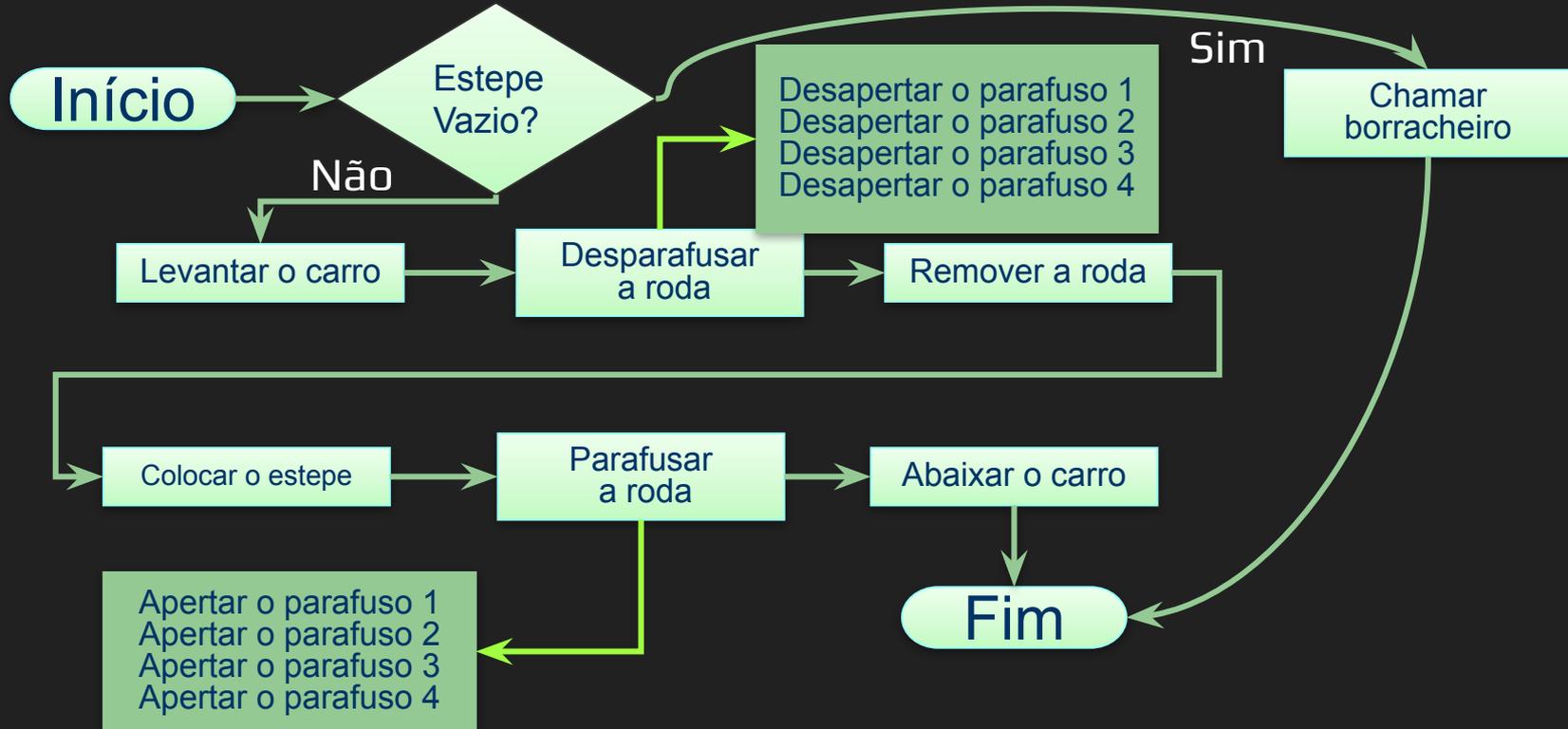
[leonardop@usp.br](mailto:leonardop@usp.br)

Baseado no material dos profs Fernando S. Osório e Claudio F.M. Toledo

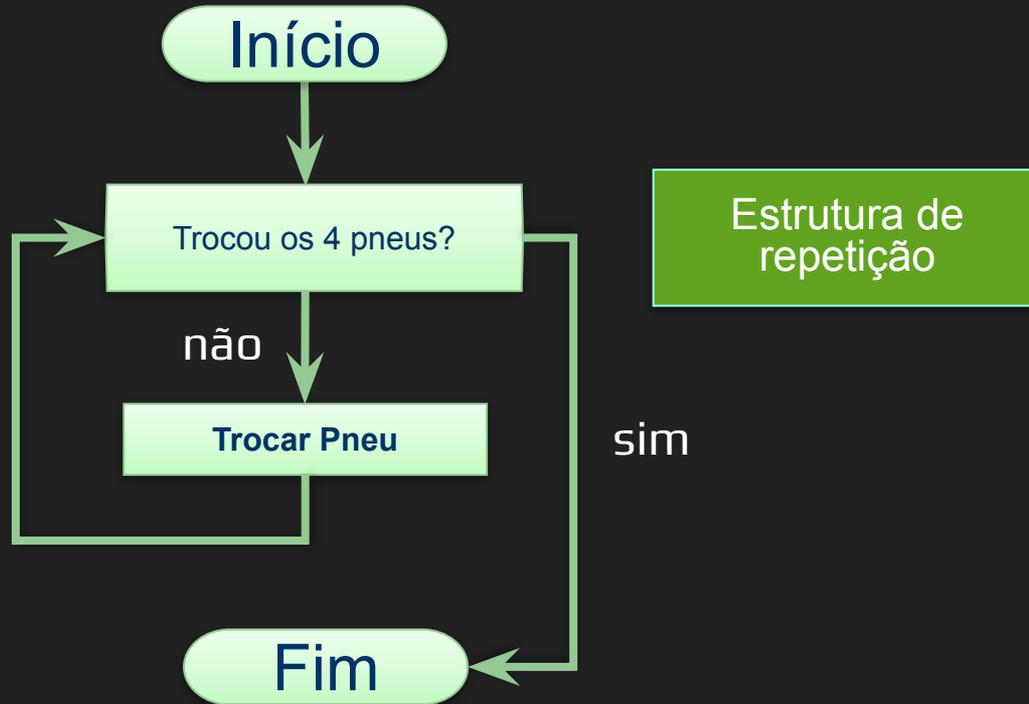
Na aula passada...

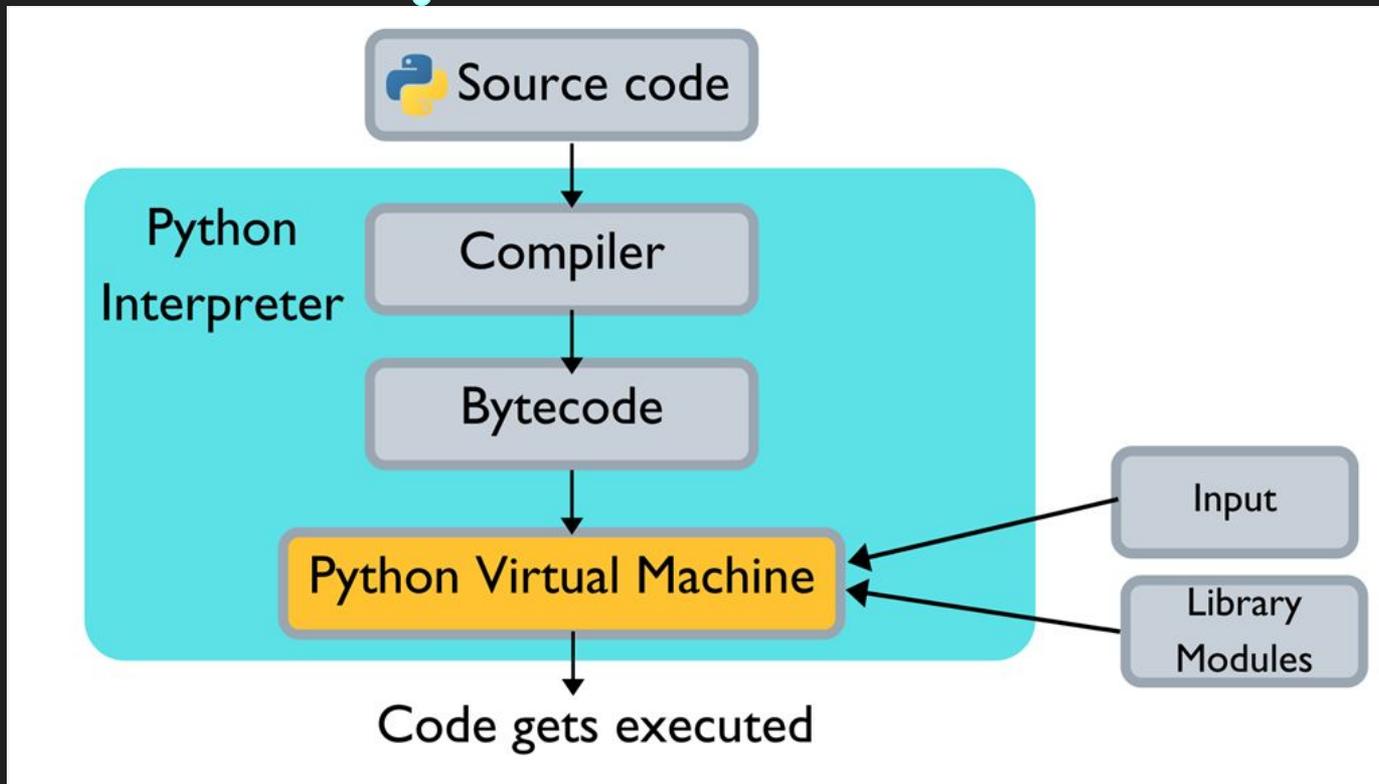


# Algoritmo para trocar um pneu



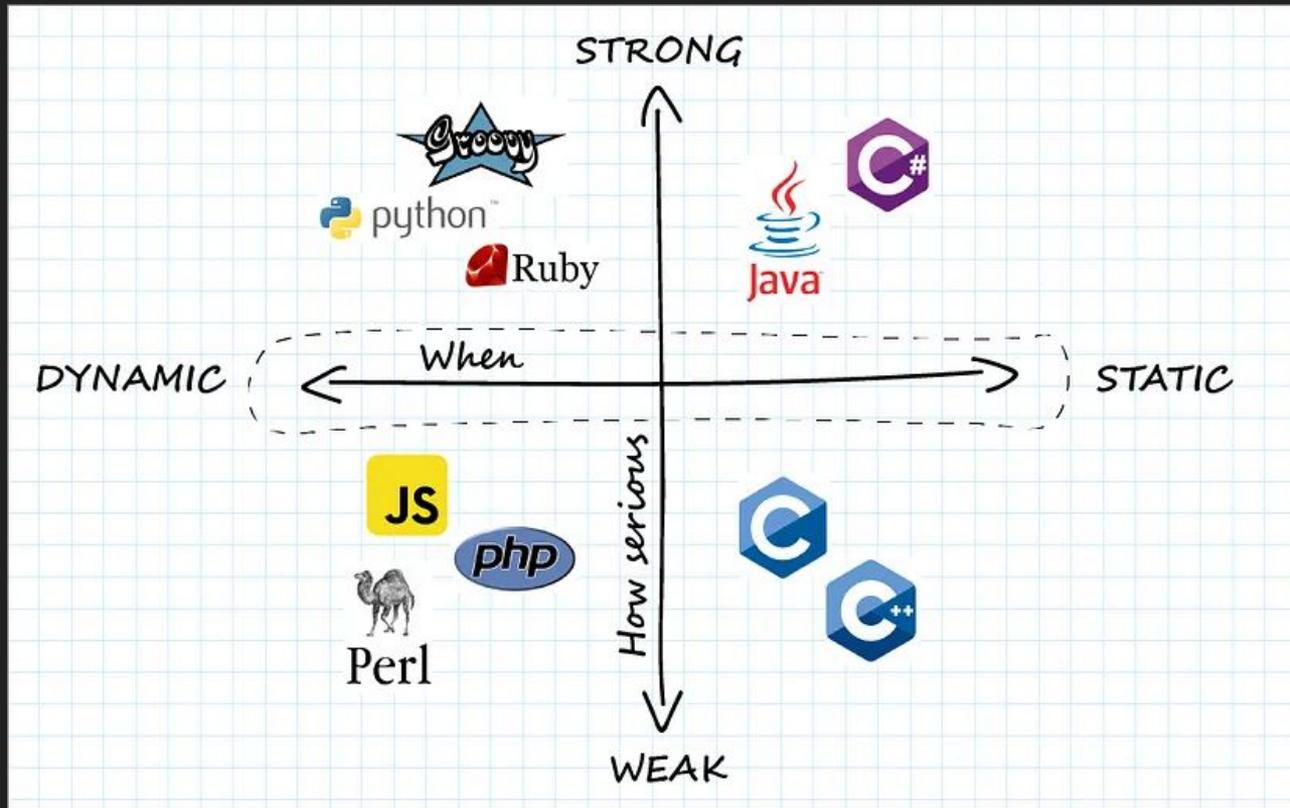
# Algoritmo para trocar um pneu





Ciclo de vida de um programa em Python

<https://www.c-sharpcorner.com/article/why-learn-python-an-introduction-to-python/>



Comparação de linguagens por tipagem. Fonte [1]

O que vamos aprender hoje?



# Objetivos

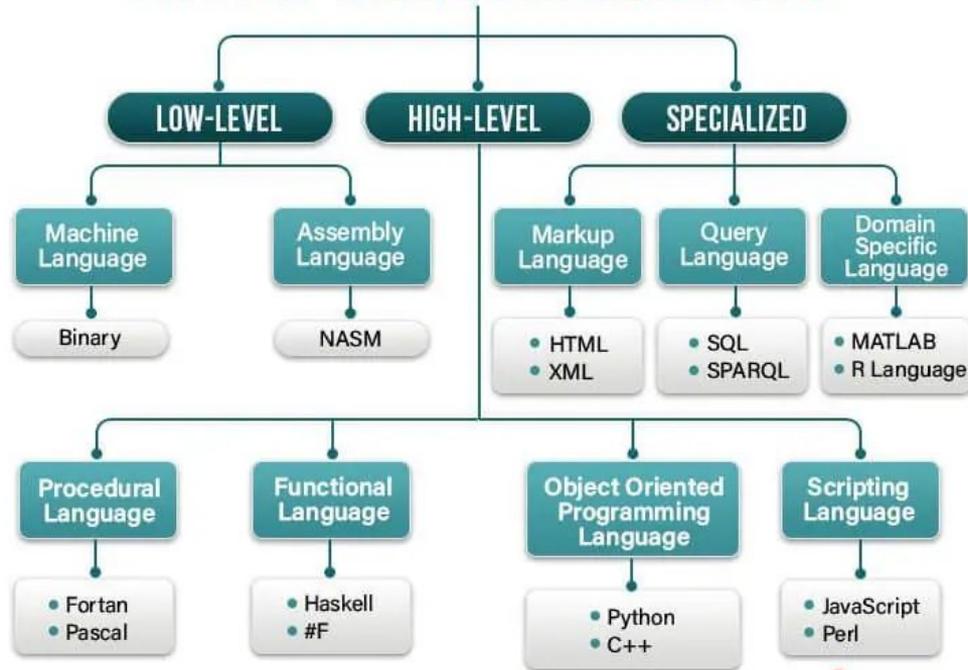
- Aprender sobre os diferentes tipos e paradigmas de programação e como Python é definido nestes termos
- Introduzir a linguagem Python
- Introduzir conceitos de operações, variáveis, e entrada e saída de dados

# Tópicos da Aula

- Tipos e paradigmas de linguagens de programação
- Variáveis, dados, entrada e saída em Python

# Tipos de linguagens

# TYPES OF COMPUTER LANGUAGES





Baixo nível

## Linguagens de baixo nível [3]

- Aquelas mais próximas ao que o computador “entende”
- Para humanos, costuma ser uma linguagem Assembly
- Para os computadores, conjuntos de bits (0 e 1)

## Baixo nível - binário [3]

01001000 -> 'H'

0110101 -> 'e'

01101100 -> 'l'

01101100 -> 'l'

01101111 -> 'o'

00100000 -> ' ' (space)

01010111 -> 'W'

01101111 -> 'o'

01110010 -> 'r'

01101100 -> 'l'

01100100 -> 'd'

# Baixo nível - Assembly [4]

```
;; Programa Hello World
section .text
global _start
_start:
    mov     edx,len           ;comprimento da mensagem
    mov     ecx,msg          ;mensagem a ser escrita
    mov     ebx,1            ;descriptor de arquivo (stdout)
    mov     eax,4            ;número da system call
; (sys_write)
    int     0x80             ;call kernel
    mov     eax,1            ;numero da syscall
; (sys_exit)
    int     0x80             ;call kernel
section .data
    msg     db 'Hello, world!',0xa    ;nossa string lindona
    len     equ $ - msg              ;comprimento da lindona
```



Alto nivel

## Linguagens de alto nível [3]

- Mais próximas das linguagens “humanas”
- Facilitam a escrita rápida de código
- São divididas em (principalmente)
  - ◆ Procedimentais (procedural)
  - ◆ Funcionais
  - ◆ Orientadas a objetos
  - ◆ De script (scripting)
  - ◆ Especializadas

## Linguagens procedurais [3]

- Criadas por procedimentos
  - ◆ Instruções em sequência com nome único
- Execução das instruções vem com um nome ou título atribuídas à instrução

# Linguagens procedurais [3]

## → Exemplos

- ◆ C
- ◆ Python
- ◆ Ada
- ◆ Lua
- ◆ JavaScript
- ◆ PHP
- ◆ ...

# Linguagens procedurais [3]

→ Exemplo em C

```
int main(){  
    printf("Hello World");  
    return 0;  
}
```

## Linguagens funcionais [3]

- Fundamentadas em funções matemáticas
- Funções equivalem a estados do sistema
  - ◆ Atribuir valores a elas
  - ◆ Usá-las como argumentos para outras funções
  - ◆ Retorná-las como valores de funções
- Concisas, limpas, e “fáceis” de entender
- Comuns para aplicações de manipulação de dados
  - ◆ Comuns para desenvolvimento web

# Linguagens funcionais [3]

## → Exemplos

- ◆ Rust
- ◆ Haskell
- ◆ Lisp
- ◆ Erlang
- ◆ F#
- ◆ Clojure
- ◆ ...

# Linguagens funcionais [3]

→ Exemplo em Rust

```
main :: IO ()
```

```
main = putStrLn "Hello, World!"
```

# Linguagens orientadas a objetos [3]

- Envolve a criação e interação com objetos
  - ◆ Funções (módulos)
  - ◆ Estruturas de dados
- Muito boas para projetos em grande escala

# Linguagens orientadas a objetos [3]

## → Exemplos

- ◆ Java
- ◆ Python
- ◆ C++
- ◆ C#
- ◆ Ruby
- ◆ Swift
- ◆ ...

# Linguagens orientadas a objetos [3]

→ Exemplo em Java

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

## Linguagens de script [3]

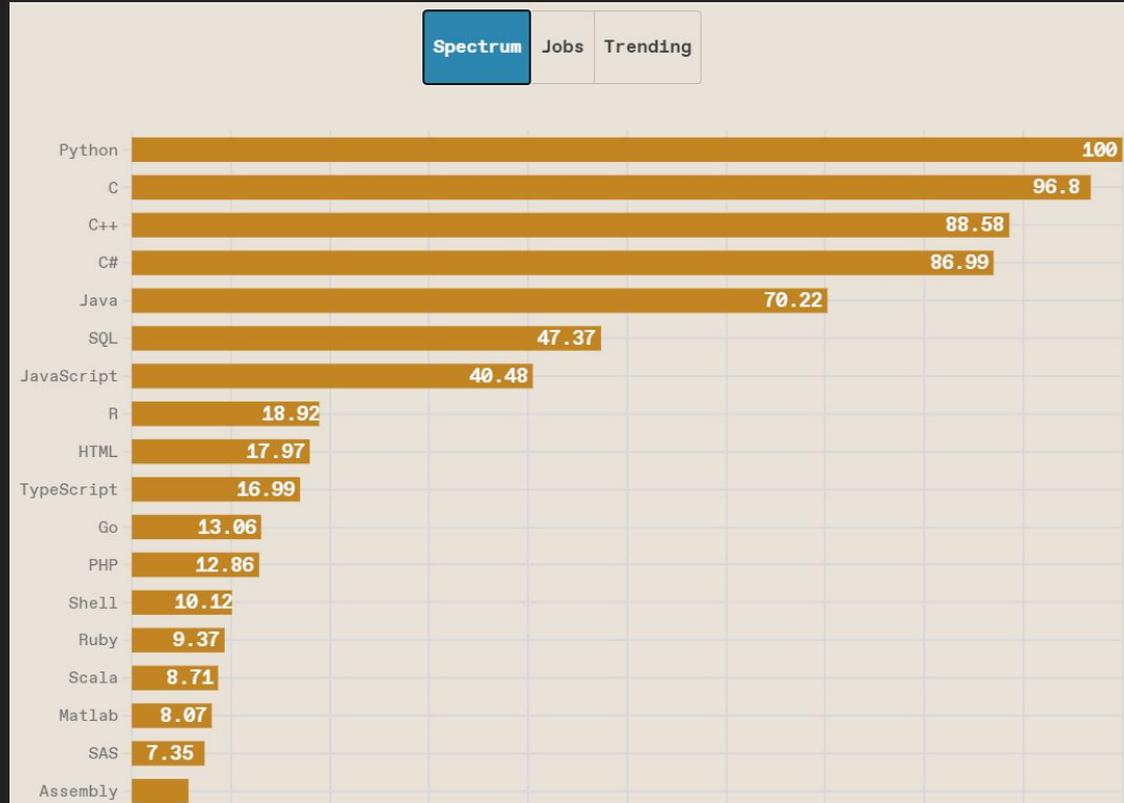
- Fáceis de aprender e bem alto-nível
- Focadas em automatização de tarefas repetitivas
- E também criação de páginas web
- Não requerem compilação antes da execução
- ◆ Facilita prototipação e teste

# Linguagens de script [3]

## → Exemplos

- ◆ JavaScript
- ◆ Python
- ◆ Perl
- ◆ Lua
- ◆ Bash

# Linguagens mais usadas



<https://spectrum.ieee.org/top-programming-languages-2022>

#1

Python

#2

JavaScript

Java

#3

HTML/CSS

JavaScript

#4

SQL

C#

#5

Python

C/C++

#6

TypeScript

PHP

# PYPL index

September 2022

#7

Java

R

#8

Bash/Shell

TypeScript

#9

C#

Go

#10

C++

Swift

PHP

# Stack Overflow

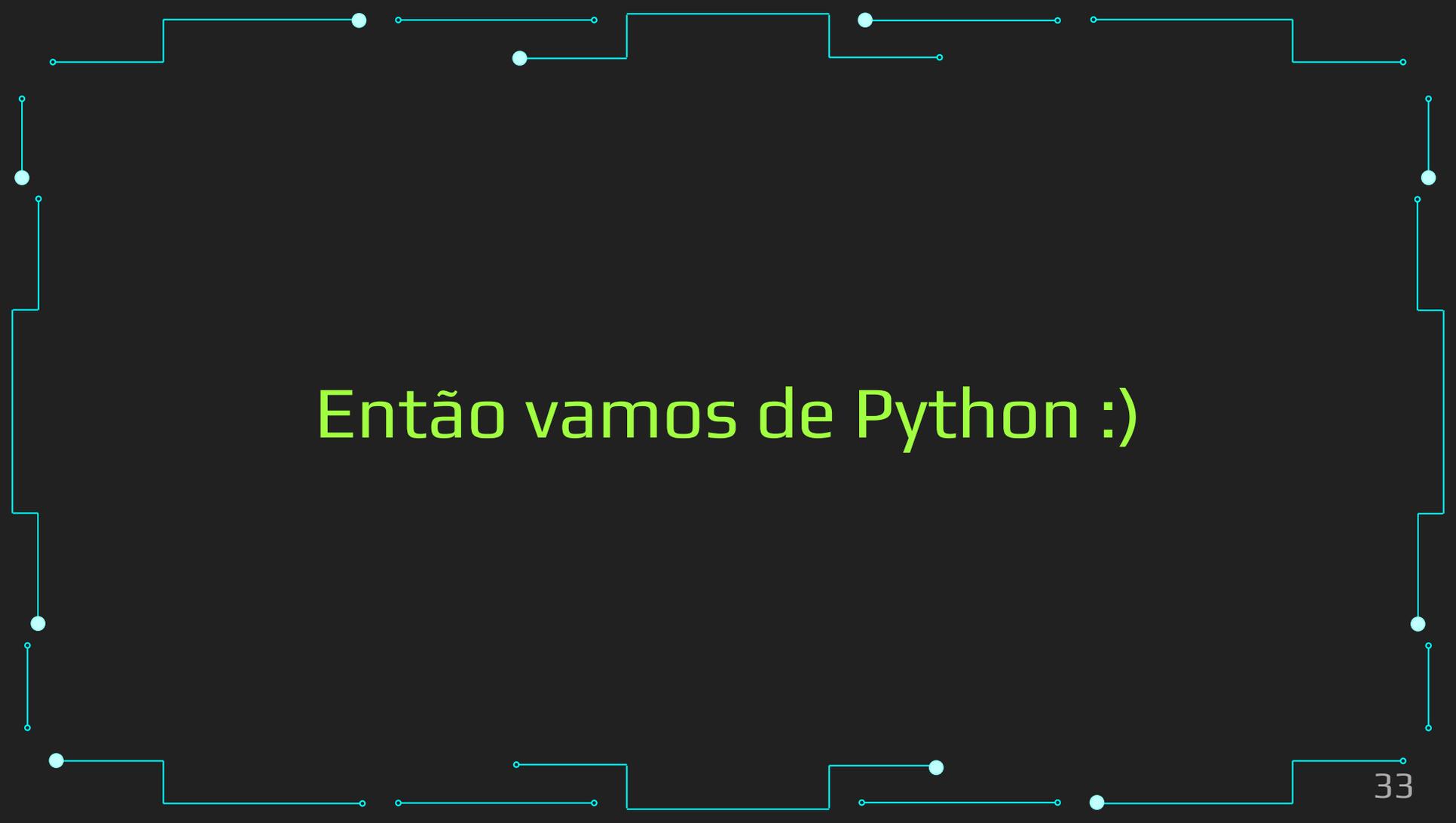
Developer Survey 2022

 Stackscale

<https://www.stackscale.com/blog/most-popular-programming-languages/>

Aug 2023	Aug 2022	Change	Programming Language	Ratings	Change
1	1		 Python	13.33%	-2.30%
2	2		 C	11.41%	-3.35%
3	4	▲	 C++	10.63%	+0.49%
4	3	▼	 Java	10.33%	-2.14%
5	5		 C#	7.04%	+1.64%
6	8	▲	 JavaScript	3.29%	+0.89%
7	6	▼	 Visual Basic	2.63%	-2.26%
8	9	▲	 SQL	1.53%	-0.14%
9	7	▼	 Assembly language	1.34%	-1.41%
10	10		 PHP	1.27%	-0.09%
11	21	▲▲	 Scratch	1.22%	+0.63%

<https://www.tiobe.com/tiobe-index/>



Então vamos de Python :)

# Python

- Lançada em 1991 por Guido van Rossum
- Multiparadigma
  - ◆ Procedural (imperativa)
  - ◆ Orientada a objetos
  - ◆ Funcional
  - ◆ De scripts

# Python

- Tipagem dinâmica\* e forte
- Grande conjunto de bibliotecas prontas
- Compilada e Interpretada

# Python

→ Muito usada em:

- ◆ Ciências de Dados
- ◆ Inteligência Artificial
- ◆ Visualização de Dados
- ◆ Web (Back-end)
- ◆ ...

# Python

## → IDEs e editores

- ◆ [Google Colab](#)
- ◆ [Jupyter](#)
- ◆ [PyCharm](#)
- ◆ [Spyder](#)
- ◆ [VSCoDe](#)
- ◆ ...

# Comandos

- Ordens para que o computador manipule dados em memória
- Exemplo clássico:
  - ◆ Olá mundo!
  - ◆ `print("Hello World!")`

# Comandos

- Como Python é interpretada, pode ser usada interativamente, como uma calculadora
- Vamos a mais um [exemplo](#)

# Tipos e Variáveis

# Tipos e Variáveis

- Como armazenar os dados de entrada, fornecidos pelo usuário?
- O que fazer com os resultados das operações?
- Variáveis são elementos que estão associados a posições de memória, cujo objetivo é o armazenamento informações.
  - ◆ ...por tempo suficiente ao seu processamento

# Identificadores

- Nome que fazem referência a elementos tais como as variáveis
- Nomes com minúsculas e maiúsculas SÃO diferenciados!
  - ◆ Ano é diferente de ano

# Identificadores

→ Regras de nomes de variáveis em Python:

- ◆ Sequência de letras (a → z, A → Z) e números (0 → 9), que devem sempre começar com uma letra.
- ◆ Apenas letras comuns são permitidas.
  - Letras acentuadas, cedilhas, espaços, caracteres especiais como \$, #, @, etc são proibidos...
  - O caractere '\_' (sublinhado/underline) é permitido!

# Identificadores

→ Em Python não precisamos dizer qual é o TIPO da variável “antecipadamente”

- ◆ Ano = 2023
- ◆ pi = 3.14159
- ◆ Nome = “Aluno”
- ◆ chuva = False

→ Não pode ser empregada qualquer uma das palavras reservadas à linguagem Python como identificadores

# Identificadores

→ Corretos:

- ◆ Contador
- ◆ Teste23
- ◆ Caixa\_Alta
- ◆ inteiro

→ Incorretos:

- ◆ 1contador
- ◆ oi!gente
- ◆ Caixa.Alta
- ◆ if

## Palavras Reservadas

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

# Tipos de Dados

- O tipo de uma variável define os valores que ela pode assumir e as operações que podem ser realizadas com ela
- Descreve a natureza da informação
- Ex:
  - ◆ Variáveis tipo ***int*** recebem apenas valores inteiros
  - ◆ Variáveis tipo ***float*** armazenam apenas valores reais
  - ◆ Variáveis tipo ***str*** armazenam textos (strings)

# Tipos de Dados

int	Número inteiro	43241423	primitivo
float	Número ponto flutuante	234234.1413	primitivo
bool	Booleano	True, False	primitivo
str	String (cadeia de caracteres)	"Bola", "Batata"	primitivo
complex	Número complexo	3+2j	não-primitivo
list	Lista heterogênea	[1, 2, 3, "eggs", "foo"]	não-primitivo
dict	Array associativo (hash table)	{"nota": 10.1}	não-primitivo
set	Conjunto ordenado único	{1, 2, 3, 4, 6, 10}	não-primitivo
tuple	Lista imutável, tupla.	(1, 2, 3, 4, "eggs")	não-primitivo

# Tipos de Dados

- Python tem tipagem dinâmica e forte
- Como saber o tipo?
- Pergunte ao python
  - ◆ *type(X)*
- Exemplos

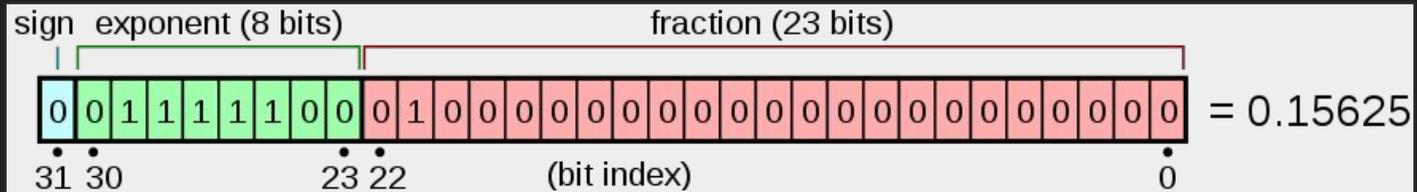
# Tamanho e precisão de dados

- Cada tipo tem seus limites
- Inteiro, por exemplo, tem 64 bits em Python
  - ◆ Valor máximo:
    - 9223372036854775807
    - -9223372036854775808
  - ◆  $2^{64} = 18.446.744.073.709.551.616$

# Float

→ *float*

- ◆ Armazena números decimais com precisão única e necessita de 4 bytes

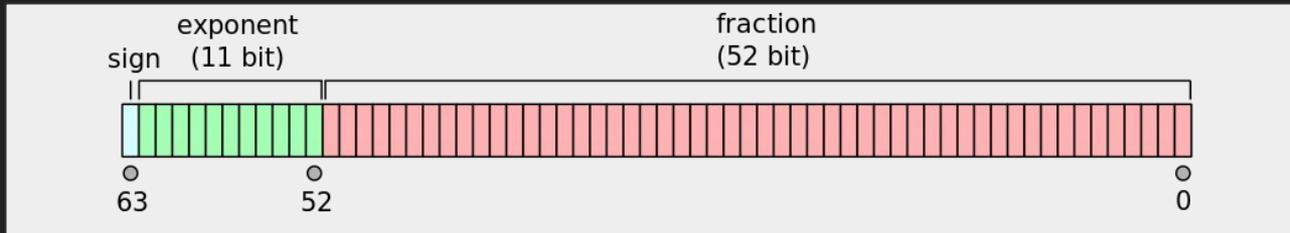


$$(-1)^{b_{31}} \times 2^{(b_{30}b_{29}\dots b_{23})_2 - 127} \times (1.b_{22}b_{21}\dots b_0)_2,$$

# Double

→ *double*

- ◆ Armazena números decimais com precisão dupla e necessita de 8 bytes



$$(-1)^{\text{sign}} (1.b_{51}b_{50}\dots b_0)_2 \times 2^{e-1023}$$

# Float em Python

→ O float em Python é um double!

- ◆ 8 bytes
- ◆ 11 bits de expoente
- ◆ 52 bits para dígitos significativos

# Conversão de Tipos

→ Tipos podem ser convertidos entre si de maneira automática (ou implícita) ou explícita (*casting*)

# Conversão Implícita

- A conversão implícita é feita pelo compilador, geralmente quando mais de um tipo está presente em uma expressão
  - ◆ Tem como objetivo evitar a perda de dados
- Dá um *upgrade* do tipo de dado para o maior tipo de dado entre as variáveis presentes
- Pode haver perda de dados mesmo assim
  - ◆ Perda de sinal e *overflow*

# Conversão Explícita

- É feita pelo programador. O usuário pode mudar o tipo de uma variável para o outro que desejar
- Sintaxe: `tipo(expressão)`
- Usado para computar expressões com variáveis de diferentes tipos de dados
- Exemplos

# Precisão

- Cuidado com a precisão dos dados, especialmente após conversões
- operações com ponto flutuante são muito sensíveis
  - ◆ <https://tecnoblog.net/noticias/2022/02/16/nubank-t-em-bug-que-nao-deixa-transferir-r-1799-e-mais-3-valores-via-pix/>
  - ◆ [https://en.wikipedia.org/wiki/Ariane\\_flight\\_V88](https://en.wikipedia.org/wiki/Ariane_flight_V88)

# Estudios Sugeridos

- [https://silentteacher.toxicode.fr/hour\\_of\\_code.html?theme=basic\\_python](https://silentteacher.toxicode.fr/hour_of_code.html?theme=basic_python)
- <https://codecombat.com/play>
- <https://www.learnpython.org/>
- <https://www.w3schools.com/python/>
- [https://www.youtube.com/watch?v=WT\\_zCgSHSTQ&list=PLcoJJSvnDgcKpOi\\_UeneTNTIVOigRQwcn](https://www.youtube.com/watch?v=WT_zCgSHSTQ&list=PLcoJJSvnDgcKpOi_UeneTNTIVOigRQwcn)
- <https://www.geeksforgeeks.org/python-programming-language/>
- <https://www.hackerrank.com/domains/python>

# Referências

# Referências

- <https://www.learnpython.org/>
- <https://www.w3schools.com/python/>
- <https://panda.ime.usp.br/cc110/static/cc110/index.html>
- [https://www.youtube.com/playlist?list=PLcoJJSvnDgcKpOi\\_UeneTNTIV0igRQwcn](https://www.youtube.com/playlist?list=PLcoJJSvnDgcKpOi_UeneTNTIV0igRQwcn)