

# *Monitoria 1*

## *Macroeconomia I*

Yuri Passuelo

10 de Março de 2025

- Cronograma Monitoria
- Bibliografia
- Contexto
- Motivação
- Parte Computacional

- 1 Introdução e Cronograma / Revisão conceitos de Programação
- 2 Continuação revisão conceitos de programação
- 3 Lista 1
- 4 Revisão Função valor / Algoritmo de Iteração da Função valor
- 5 Lista 2
- 6 Problema com Escolha de Trabalho e Incerteza

## Bibliografia Base:

- Macroeconomic Theory ; Dirk Krueger (2017)
- Recursive methods in economic dynamics ; Nancy Stockey, Robert Lucas e Edward Prescott (1989)
- Recursive Macroeconomic theory; Lars Ljungqvist e Thomas Sargent (2014)

## Opcional/Curiosidade:

- Dynamic General Equilibrium Modelling: Computational Methods and Applications ; Burkhard Heer e Alfred Maussner (2005)

Disponível no Drive da Turma de 2024

Uma forma simples de se pensar na disciplina de Macroeconomia na Pós..

- Inclusão do elemento "*tempo*"
- *Microfundamentar* o que já existe na macro "Modelo de Crescimento Neoclássico"
- Introdução de "*escolhas*"

Exemplo de um problema de Microeconomia na Graduação:

$$\max (x_1^r + x_2^r)^{\frac{1}{r}}$$

Sujeito à:

$$p_1x_1 + p_2x_2 \leq R$$

$$x_1, x_2 \geq 0$$

Aqui temos que existe um único período no tempo, e dados preços  $p_1$  e  $p_2$  e a renda  $R$ , individuo deve escolher  $x_1$  e  $x_2$  que maximizem a utilidade.

**Solução:** Lagrangeano

## Contexto

Outro exemplo de otimização que é visto na graduação é uma que trabalha com dois períodos no tempo, por exemplo, em apenas dois períodos.

$$\max \log c_1 + \beta \log c_2$$

Onde:

$$\beta = \frac{1}{1+r}$$

Supondo:

$$p_1 = 1$$

Sujeito à:

$$c_1 + \frac{c_2}{1+r} = e_1 + \frac{e_2}{1+r}$$

Problema visto em aula em Macro I:

$$\max \sum_{t=0}^{\infty} \beta^t u(c_t^i)$$

Sujeito à:

$$\sum_{t=0}^{\infty} p_t c_t^i = \sum_{t=0}^{\infty} p_t e_t^i$$

$$c_t^i \geq 0$$



# Contexto

Outro problema mais complexo:

$$\max \mathbb{E}_0 \left\{ \sum_{t=0}^{\infty} \beta^t u(c_t, l_t) \right\}$$

Com:

$$u(c, l) = \frac{c^{1-\gamma}}{1-\gamma} - \frac{l^{1+\phi}}{1+\phi}$$

Sujeito à:

$$k_{t+1} = (1 - \delta)k_t + i_t$$

$$y_t = c_t - i_t$$

$$y_t = z_t k_t^\alpha l_t^{1-\alpha}$$

Porque vem se tornando cada vez mais importante?

- Mudança da natureza dos problemas
- Incremento da complexidade dos modelos
- Maioria dos modelos não terão solução analítica, apenas computacional
- Não é uma exclusividade da Macroeconomia, áreas da estatística como a própria estatística Bayesiana passam por isso
- Existência de um campo aplicado que necessitar dessas ferramentas

# Parte Computacional

- Inicialmente nossa preocupação será implementar alguns algoritmos no computador
- Dentro da Macroeconomia Moderna que aplica esses modelos teóricos há uma enorme intersecção com o campo da computação científica.
- Lida basicamente com a criação de algoritmos e programas para resolver determinados problemas, problemas esses que são complexos e computacionalmente intensivos
- Hoje existe um nicho de linguagens utilizadas nesse campo:
  - ▶ MATLAB
  - ▶ Julia
  - ▶ Fortran
  - ▶ C++
  - ▶ Em menor medida o Python com apoio de bibliotecas como:
    - ★ Numba
    - ★ Cython
    - ★ Numpy
    - ★ Dentre Outros

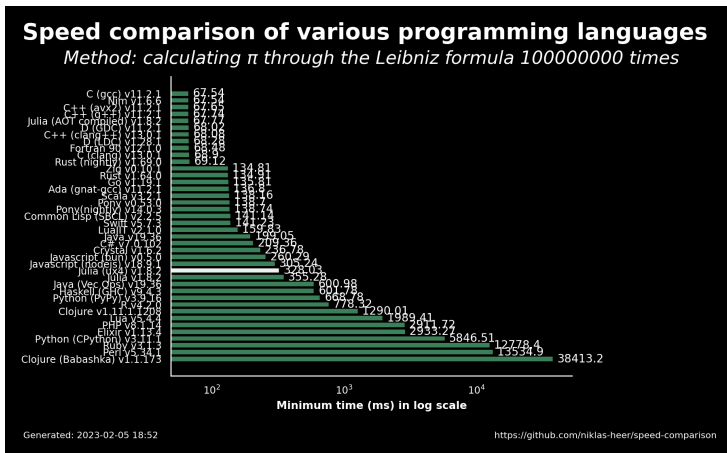


Figure: Velocidade linguagens de programação

# Porque Python?

Não é a linguagem mais rápida, mas...

- É relativamente fácil de se aprender;
- Tem uma ampla gama de bibliotecas disponíveis;
- Tem uma comunidade grande;
- Muitos materiais disponíveis.

Opções para utilização do Python:

- Instalação direta no computador
- Instalação via Anaconda.
- Utilização via Google Colab

## Instalação direta no computador

- **Vantagens:** Maior grau de personalização e integração com seu computador.
- **Desvantagens:** Maior trabalho na administração de bibliotecas, *drivers* e configurações de comando, necessita de um IDLE.

## Passo-a-passo:

- 1 Instalação do Python no [link](#)
- 2 Script `pip`
  - 1 Baixar o script `get-pip.py` no [site](#)
  - 2 Executar o script
- 3 Adicionar diretório `/Python/Python39/Scripts` na variável de ambiente Path

## Usando Anaconda

- **Vantagens:** Instalação única, usa ambiente separado do computador, contém diferentes opções de IDLE já instalados, fácil instalação de bibliotecas.
- **Desvantagens:** Toma muito espaço de armazenamento, instalação de bibliotecas fora do escopo é mais difícil.

## Passo-a-passo:

- 1 Entrar no [site](#)
- 2 Fazer o Download do Cliente
- 3 Esta pronto para uso!



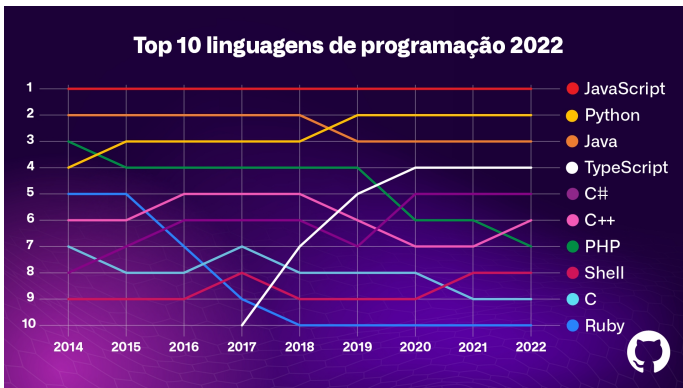
## Utilização do Google Colab.

- Vantagens: Não precisa instalar nada, ampla gama de bibliotecas disponível de forma imediata sem pagar nada.
- Desvantagens: Instabilidade do servidor, lentidão na execução de *scripts*, limitação de tipo de uso.

Bastar entrar no [site](#) com a conta google logada, scripts podem ser salvos no seu próprio google drive.

Nosso objetivo não é necessariamente dar uma aula profunda da linguagem mas abordar e passar sobre elementos essenciais da linguagem que permitam:

- Entender os principais elementos comuns entre diversas linguagens de programação;
- Saber resolver problemas simples com a utilização do computador;
- Programar e entender o que será executado;
- Ler Pseudo-Códigos e saber implementá-los.



*Figure:* Linguagens mais Usadas no Github

- Criada em 1991 Hoje é uma das linguagens mais utilizadas no mundo

- Sua disseminação se deve a sua simplicidade e versatilidade
- Linguagem Multiparadigma
  - ▶ Orientação a objetos
  - ▶ Funcional
  - ▶ Procedural
  - ▶ Estruturada
- É uma linguagem de propósito geral:
  - ▶ Desenvolvimento Web
  - ▶ Desenvolvimento de Jogos
  - ▶ *Backend* e Infraestrutura
  - ▶ Computação Científica
  - ▶ Aprendizado de Máquina

O que abordaremos?

- 1 Introdução a linguagem de programação
- 2 Controle de fluxos
- 3 Estruturas de Dados
- 4 Funções e Funções Anonimas
- 5 Um pouco da biblioteca `numpy`
- 6 Se der um pouco de `numba`

Exemplos e material baseada na documentação oficial no [site](#)