

# Aula 2 - Setup e Python Básico

## Programação para Economistas

Prof.: Alexandre Rabelo (alexandre.rabelo@mackenzie.br)

Disciplina: Programação para Economistas

Universidade Presbiteriana Mackenzie - CCSA

2026/1

# Roteiro de Hoje

## ▶ Setup dos ambientes

- ▶ Spyder (local)
- ▶ Google Colab (online)

## ▶ Python Básico

- ▶ Variáveis e atribuição.
- ▶ Tipos de dados essenciais.
- ▶ Operações básicas.

## ▶ Prática no computador (~45 min)

- ▶ Exercícios guiados
- ▶ Primeiro código econômico

# Setup dos Ambientes

# Dois Ambientes, Sua Escolha

## Spyder

Instalado nos PCs do lab  
IDE profissional  
Scripts (.py)  
Editor + Console + Variáveis

**Bom para:** Código sequencial

## Google Colab

Online (navegador)  
Notebooks (.ipynb)  
Células interativas  
Código + texto + gráficos

**Bom para:** Exploração/análise

## Importante!

**Aulas 2-4:** Escolham qual usar (vou fornecer materiais para ambos)

**Aula 5+:** TODOS usarão Colab (bibliotecas pré-instaladas)

# Setup: Abrindo o Spyder

## ► Nos computadores do lab:

1. Abra o menu **Iniciar**
2. Busque **"Spyder"**
3. Clique para abrir
4. Aguarde carregar  
(pode demorar ~30s na 1ª vez)

## ► Primeiro teste:

1. No editor, digite:

```
print("Olá, Mackenzie!")
```

2. Aperte **F5** (rodar arquivo)  
ou **Ctrl+Enter** (rodar linha)
3. Veja o resultado no **Console!**

## Em casa: instalar Anaconda

1. Acesse `anaconda.com`
2. Baixe **Anaconda Individual**
3. Instale (inclui Python + Spyder + bibliotecas)

Tutorial completo no Moodle!

## Atalhos essenciais

**F5** — Rodar arquivo todo

**Ctrl+Enter** — Rodar linha/seleção

**Ctrl+1** — Comentar/descomentar

**Tab** — Autocompletar

# Interface do Spyder: Os 4 Painéis

## 1. Editor

- ▶ Onde você **escreve** o código.
- ▶ Salvo em arquivos `.py`.

## 2. Painéis direitos (abas)

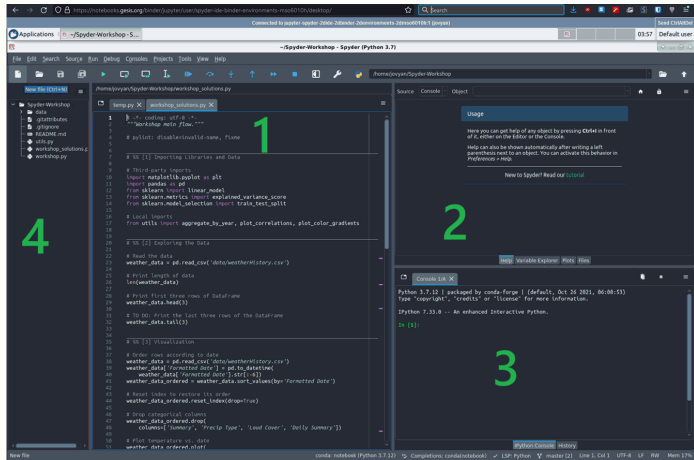
- ▶ Variáveis criadas (Variable explorer), Gráficos (Plots), manual das funções (Help).

## 3. Console IPython

- ▶ Onde o código **executa**.
- ▶ Resultados aparecem aqui.

## 4. Explorador de arquivos

- ▶ Navegue pelos seus projetos.



# Setup: Abrindo o Google Colab

## ► Acesso (sem instalar nada!):

1. Abra o navegador.
2. Acesse `colab.research.google.com`.
3. Faça login com conta **Gmail**.
4. Clique em "**Novo notebook**".

## ► Primeiro teste:

1. Clique em + **Código** para criar célula.
2. Digite:

```
print("Olá, Mackenzie!")
```

3. Aperte **Shift+Enter**.
4. Veja o resultado aparecer **abaixo da célula**!

## Vantagens do Colab

- Funciona em **qualquer computador**.
- Salva no **Google Drive** automaticamente.
- Compartilha com um **link**.
- Bibliotecas **pré-instaladas** (pandas, matplotlib, numpy...).

## Atalhos essenciais

**Shift+Enter** — Executar célula e avançar

**Ctrl+Enter** — Executar e ficar na célula

**Ctrl+MB** — Nova célula abaixo

**Ctrl+MM** — Converter para texto

# Interface do Colab: Os 4 Elementos

## 1. Célula de Código

- ▶ Onde você **escreve** o código.
- ▶ Botão ou Shift+Enter para executar.

## 2. Célula de Texto

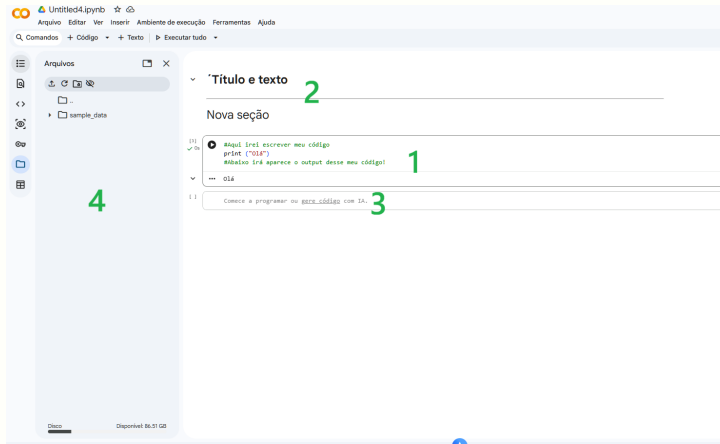
- ▶ Títulos, explicações, anotações.
- ▶ Código + texto no **mesmo** arquivo!

## 3. Célula Vazia

- ▶ + **Código** ou + **Texto** para adicionar.

## 4. Painel Lateral

- ▶ Navegação, arquivos, variáveis.





# Qual Devo Usar?

## Use Spyder se você:

- ▶ Prefere escrever tudo e rodar depois.
- ▶ Gosta de ver todas as variáveis.
- ▶ Tem Anaconda instalado em casa.
- ▶ Vem do MATLAB/RStudio.

## Use Colab se você:

- ▶ Prefere testar linha por linha.
- ▶ Quer misturar código e anotações.
- ▶ Quer trabalhar de qualquer lugar.
- ▶ Já usa Google Drive.

### Dica do Professor

Ambos são excelentes! **Não existe escolha errada..**

# Python Básico

# Primeiro Conceito: Variáveis

- ▶ **Variável** = "caixinha" que guarda um valor
- ▶ **Atribuição**: usamos o sinal =

## Exemplo

```
pib = 2.1          # Brasil (trilhões USD)
populacao = 215    # Brasil (milhões)
moeda = "Real"     # Moeda brasileira
```

## Atribuição

- ▶ Nome à **esquerda** do =
- ▶ Valor à **direita** do =
- ▶ # = comentário (Python ignora)

## Nomes de variáveis:

- ▶ **Pode**: letras, números, ...
- ▶ **Bom**: Usar letras minúsculas (pib\_pc).
- ▶ **Proibido**: começar com número, espaços e acentos.
- ▶ **Evite**: x, a, var1 (nomes descritivos são muito melhores!).

# Tipos de Dados Essenciais

- ▶ **Inteiros** (`int`): números sem vírgula.
- ▶ **Decimais** (`float`): números com vírgula.
- ▶ **Texto** (`str`): entre aspas
- ▶ **Booleanos** (`bool`): True ou False.

## Exemplos

```
ano = 2026           # int
taxa_juros = 10.75   # float
pais = "Brasil"      # str
aprovado = True       # bool
```

**Verificar tipo:** `type(variavel)`

# Operações Básicas

## Aritméticas:

```
10 + 5      # Adição (15)
10 - 5      # Subtração (5)
10 * 5      # Multiplicação (50)
10 / 5      # Divisão (2.0)
10 ** 2     # Potência (100)
```

## Com variáveis:

```
pib = 2.1
populacao = 215
pib_per_capita = pib / populacao
print(pib_per_capita)
# Resultado: 0.00976...
```

## Dica

Python respeita ordem de operações: `()` → `**` → `*` / `/` → `+` -

## Exibindo Resultados: `print()`

- ▶ `print()`: exibe valores na tela

### Uso Simples

```
pib = 2.1
print(pib)                                # Exibe: 2.1
```

### Uso com Texto (f-strings)

```
pib = 2.1
populacao = 215
print(f"PIB: ${pib} trilhões")
print(f"População: {populacao} milhões")
# Exibe: PIB: $2.1 trilhões
#           População: 215 milhões
```

**f-strings:** `f"texto {variavel}"` substitui variável pelo seu valor

## Exemplo Completo: Cálculo Econômico

Problema: Calcular PIB per capita do Brasil e exibir resultado formatado.

### Solução

```
# Dados
pib_tri = 2.1          # PIB em trilhões USD
pop_mi = 215           # População em milhões

# Converter unidades (ambos em bilhões)
pib_bi = pib_tri * 1000  # trilhões → bilhões
pop_bi = pop_mi / 1000   # milhões → bilhões

# Calcular PIB per capita
pib_per_capita = pib_bi / pop_bi

# Exibir resultado
print(f"PIB per capita: ${pib_per_capita:.2f} mil")
```

# Vamos Praticar!



## Acessar códigos via GitHub

Acesse: <https://github.com/progecono/2026.01>

# Para Casa

## ▶ **Leitura:**

- ▶ McKinney Cap. 2 (p. 17-41): Python Language Basics
- ▶ Coding for Economists: Cap. 2.1-2.3 (Basics)

## ▶ **Prática:**

- ▶ Completar exercícios da aula (se não terminou)
- ▶ Lista 1 no Moodle (prazo: 1 semana)
- ▶ Experimentar com dados de outros países

## ▶ **Próxima aula:**

- ▶ Listas e Dicionários
- ▶ Estruturas de dados fundamentais
- ▶ Mais cálculos econômicos!