

# PYTHON PARA JORNALISTAS DE DADOS

## Introdução e lógica de programação

Rodolfo Viana  
MBA em Jornalismo de Dados  
10 de julho de 2021



# Programação | Por que um jornalista usaria Python?

Python é uma linguagem extremamente versátil, com uma sintaxe de fácil compreensão.



Dropbox usa para o aplicativo instalado no computador



Instagram usa no backend da plataforma



Spotify usa para análises de dados e machine learning

# Programação | Por que um jornalista usaria Python?

Devido a essa versatilidade e simplicidade de escrita de códigos, é cada vez mais usada em redações ao redor do mundo para a execução de pautas em que se fazem necessárias investigações de grandes volumes de dados.

## theguardian

The Guardian usou para coletar dados de anúncios do Facebook sobre as eleições dos EUA [[Fonte](#)]

## Los Angeles Times

LA Times usou para analisar dados da covid-19 no estado da Califórnia [[Fonte](#)]

## ESTADÃO

Estadão usou para observar 15 mil tweets de candidatos à presidência em 2018 [[Fonte](#)]



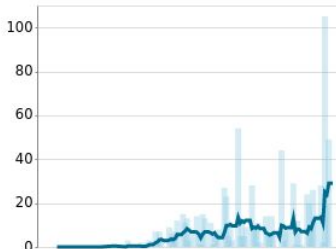
Globo usou para estudar a impunidade na Guarda Civil Metropolitana de São Paulo [[Fonte](#)]

# Programação | Por que um jornalista usaria Python?

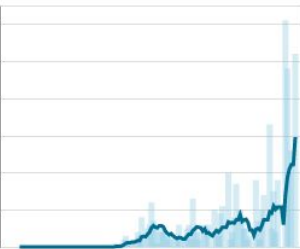
```
In [111]: chart_list = []
          for county in list(biggest_county_jumps.head(12).county):
              this_df = chronological_county_df[chronological_county_df.county == county]
              chart = alt.Chart(this_df, title=county).encode(
                  x=alt.X("date:T", title=None, axis=None),
              )
              lines = chart.mark_line(color=lat.palette["accent"]).encode(
                  y=alt.Y("new_confirmed_cases_rolling_average:Q", title=None),
              )
              bars = chart.mark_bar(color=lat.palette["default"], opacity=0.33).encode(
                  y=alt.Y("new_confirmed_cases:Q", title="New confirmed cases"),
              )
              chart_list.append((bars + lines).properties(height=200, width=250))
          facet_wrap(chart_list, plots_per_row=4)
```

Out[111]:

Solano



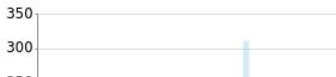
Merced



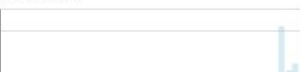
Marin



Santa Barbara



Stanislaus



Sacramento



Trecho de pauta orientada por dados no LA Times [\[Íntegra\]](#)

# Programação | Por que um jornalista usaria Python?

## → Eu posso trocar Python por Excel, não?

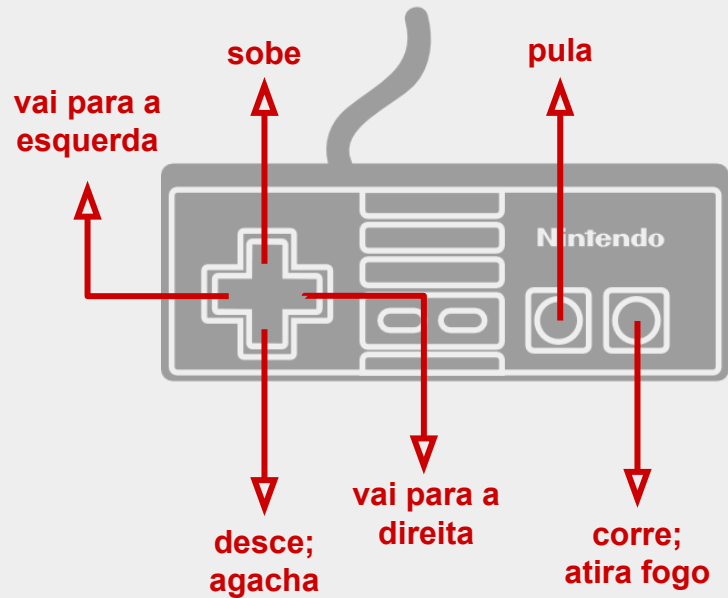
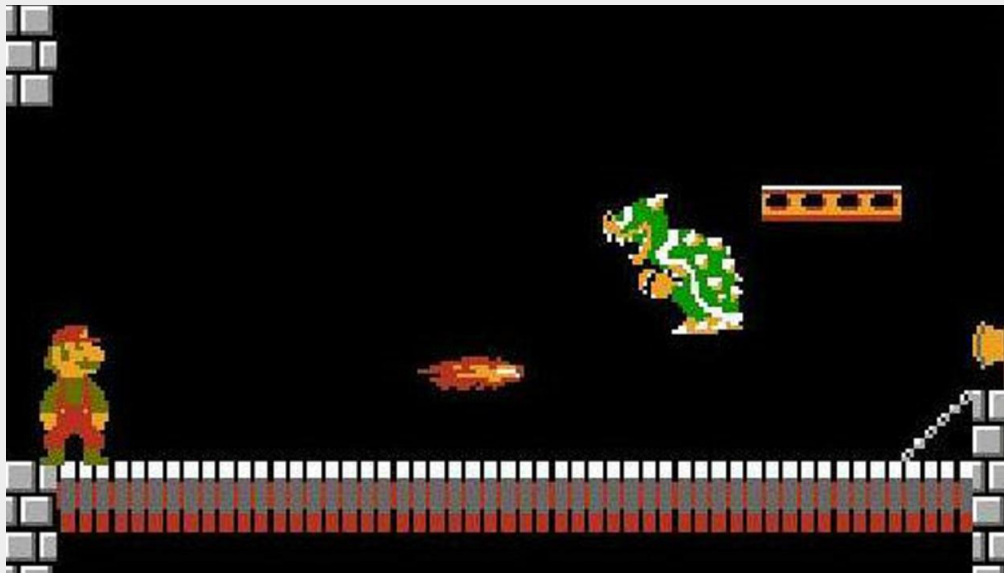
Editores de planilhas em geral têm limitações em relação à quantidade de registros. O Excel, por exemplo, suporta 1.048.576 linhas, o que torna impossível trabalhar com bases como Caged (~32 milhões de linhas/ano), Censo Escolar (~70 milhões de linhas/ano na base de matrículas) e outras. Além disso, editores de planilhas não têm funcionalidades que permitem, por exemplo, raspar dados ou mesmo fazer agregações mais elaboradas.

## → Todo jornalista de dados precisa saber Python, então?

Nem todos trabalham especificamente com Python. Há jornalistas que preferem R, uma linguagem voltada para análises estatísticas; há ainda repórteres que não trabalham com linguagem alguma e são bons jornalistas de dados. A questão, portanto, é: quanto mais ferramentas um jornalista de dados conhecer (como Python, R, SQL, Excel, visualização etc.), melhor e mais rápido ele trabalhará com dados, extraíndo pautas mais consistentes.

# Programação | Introdução

Como Mario pode derrotar Bowser?



# Programação | Conceituação

De forma simples, programar é escrever instruções que uma máquina possa interpretar e, em seguida, executar a tarefa para a qual foi instruída.

Juntas, essas instruções têm o nome de algoritmo.

```
# Instrução 1: pede ao usuário o nome
nome = input("Digite o seu nome: ")
# Instrução 2: pede ao usuário o ano
ano = input("Digite o ano de seu nascimento: ")
# Instrução 3: converte o ano em formato texto para formato numérico
ano = int(ano)
# Instrução 4: calcula a diferença entre 'ano' e 2021
anos_em_2021 = 2021 - ano
# Instrução 5: monta a frase substituindo chaves pelos valores
frase = "{} completou ou completará {} anos".format(nome, anos_em_2021)

# Instrução 6: imprime em tela a frase
print(frase)
```

Tente reproduzir este código no seu computador!

# Programação | Conceituação

Para o computador entender instruções que escrevemos, precisamos que:

1. a escrita seja compreensível ao computador

sintaxe

```
# A linha abaixo funciona
print("Vamos aprender um pouco de Python!")

# As linhas abaixo NÃO funcionam
print(Vamos aprender um pouco de Python!)
print("Vamos aprender um pouco de Python!')
print("Vamos aprender um pouco de Python!")
```

2. a escrita siga uma estrutura lógica

lógica

```
# O código abaixo funciona
a = 5
b = 3
resultado = a * b
print(resultado)


# O código abaixo NÃO funciona
a = 5
resultado = a * b
b = 3
print(resultado)
```



# Programação | Sintaxe

Para que o computador consiga interpretar o que escrevemos, é preciso que os comandos sejam escritos corretamente.

Daí a importância da sintaxe (que veremos ao longo de todo o módulo).



```
# Python usa "espaço", e não "tab" em comandos de múltiplas linhas
# Correto: 4 espaços
```

```
if numero > 5:
    print(numero**2)
```

```
# Errado: 1 tab
```

```
if numero > 5:
    print(numero**2)
```

```
# Python aceita aspas simples ou duplas, mas não podemos misturar
# Correto: aspas simples antes e depois
```

```
print('Introdução à linguagem Python')
```

```
# Errado: aspas simples antes e aspas duplas depois
```

```
print('Introdução à linguagem Python")
```

```
# Parênteses, colchetes e chaves, quando abertos, precisam ser fechados
# Correto: lista com sinais gráficos no começo e no fim
```

```
frutas = ["maçã", "uva", "abacate"]
```

```
# Errado: lista com sinais gráficos apenas no começo
```

```
frutas = ["maçã", "uva", "abacate"
```

```
# Variáveis não aceitam acentuação, a maioria dos sinais gráficos,
# e não podem começar com números
```

```
# Correto: sem acentos; uso de "underline"
```

```
inicio_modulo = "10/07/2021"
```

```
# Errado: com acentos; uso de hífen
```

```
início-módulo = "10/07/2021"
```

# Programação | Lógica

A máquina apenas executa instruções quando essas instruções estão encadeadas ordenadamente.

Por exemplo: se peço para ela imprimir o valor de  $n$ , é preciso ter o valor de  $n$  anteriormente estabelecido.

Isso é a lógica.



```
# Um código que segue a lógica e terá resultado
numeros = [1, 2, 3, 4] # Temos uma lista de números...
for n in numeros: # ...e para cada item (n) na lista...
    if n % 2 == 0: # ...se sobrar 0 na divisão do item (n) por 2...
        print("Este número é par") # ...imprimimos que é par...
    else: # ...caso contrário, se não sobrar 0...
        print("Este número é ímpar") # ...imprimimos que é ímpar

# Um código que não segue a lógica e quebrará
numeros = ["1", "2", "3", "4"] # Temos uma lista de números, mas em formato texto...
for n in numeros:
    if n % 2 == 0:
        print("Este número é par")
    else:
        print("Este número é ímpar")
# O código vai quebrar porque o número está em formato texto,
# e textos não podem ser usados em operações aritméticas

# Outro código que não segue a lógica e quebrará
numeros = [1, 2, 3, 4]
for n in Algarismos:
    if n % 2 == 0:
        print("Este número é par")
    else:
        print("Este número é ímpar")
# O código vai quebrar porque não temos uma lista chamada "Algarismos"
```

# Programação | Material complementar

Para ler

- Capítulo “O caminho do programa” de *Como pensar como um cientista da computação*, em IME-USP [[link](#)]

Para assistir

- *Lógica de programação e algoritmos: introdução*, em Programador BR [[link](#)]
- *Lógica de programação e algoritmos: como o computador entende as coisas?*, em Programador BR [[link](#)]

Para exercitar

- Sudoku, em Tagesspiegel [[link](#)]
- Atividades, em Toda Matéria [[link](#)]

## Programação | Atividades para casa

Há dois arquivos .py que acompanham esta aula e que devem ser abertos no VS Code:

- 01\_01.py: algoritmo de cálculo de área e perímetro de um círculo
- 01-02.py: algoritmo de cálculo de proporção de doses de vacina contra covid-19 aplicadas na região sudeste

Atividade	Entrega	Nota
01_01.py	15/07	0,1
01_02.py		0,1
Enviar para eu@rodolfoviana.com.br		

O trabalho em ambos não é escrever códigos, mas organizar as linhas para que rodem adequadamente. Como as linhas estão comentadas, não é preciso saber Python: basta usar a lógica.

Ao fim da organização das linhas, se você rodar o código e o resultado for igual ao *output* que está nos arquivos, a ordenação está correta.