



HASHIRA DO PYTHON

A RESPIRAÇÃO DO CÓDIGO



**6 Formas básicas de começar
com Python**

João Vitor

Python: A Forma da Simplicidade

Introdução

Python é uma linguagem de programação que conquistou o mundo graças à sua clareza, simplicidade e versatilidade. Criada por Guido van Rossum nos anos 90, ela é usada em diversas áreas, desde sites e aplicativos até inteligência artificial e análises de dados. Seu objetivo sempre foi ser uma linguagem que “todo mundo consegue ler e entender”. Por isso, sua sintaxe é mais próxima do inglês do que a maioria das outras linguagens.

Aprender a manipular dados em Python passa por dominar os **seletores** — formas de acessar e extrair informações de listas, strings, dicionários e outras estruturas. Esses recursos são fundamentais porque permitem trabalhar de forma rápida e eficiente com qualquer tipo de dado.

A seguir, você verá os principais seletores do Python, explicados de maneira simples e com exemplos em contextos reais.



01

SELECIONANDO ITENS POR ÍNDICE

Selecionando Itens por Índice



O índice é a forma mais básica de selecionar elementos em Python. Listas, strings e tuplas são sequências indexadas, onde cada item ocupa uma posição. O índice sempre começa no 0.

```
Hashira do Python - João Vitor.py

# Lista de frutas
frutas = ["maçã", "banana", "laranja", "uva"]

# Selecionando pelo índice
print(frutas[0]) # maçã
print(frutas[2]) # laranja

# Em strings também funciona
texto = "Python"
print(texto[1]) # y
```



Quando usar:

Quando você já sabe exatamente a posição do item e precisa acessá-lo rapidamente.

02

SELECIONANDO FAIXAS COM SLICING



Selecionando Faixas com Slicing



O slicing é um dos recursos mais poderosos do Python. Ele permite “fatiar” listas, tuplas ou strings usando a notação início:fim. Importante: o índice final não é incluído no resultado.

```
Hashira do Python - João Vitor.py

# Lista de notas
notas = [8.5, 9.0, 7.2, 6.8, 10]

# Pegando da posição 1 até a 3 (não inclui o 3)
print(notas[1:3]) # [9.0, 7.2]

# Do início até o índice 2
print(notas[:2]) # [8.5, 9.0]

# Do índice 2 até o final
print(notas[2:]) # [7.2, 6.8, 10]
```



Quando usar:

Quando você precisa de subconjuntos de dados, como pegar as três primeiras notas de um aluno ou as últimas letras de uma palavra.

03

SELECIONANDO COM PASSO

Selecionando com Passo



Além de início e fim, o slicing aceita um terceiro parâmetro, chamado passo. Ele define de quantos em quantos elementos a seleção será feita.

```
Hashira do Python - João Vitor.py

numeros = list(range(10))

print(numeros[::2])    # [0, 2, 4, 6, 8] → pares
print(numeros[1::2])   # [1, 3, 5, 7, 9] → ímpares

# Revertendo uma lista
print(numeros[::-1])   # [9, 8, 7, ..., 0]
```



Quando usar:

- Para selecionar apenas números pares ou ímpares.
- Para inverter uma lista ou string rapidamente.
- Para criar amostras em intervalos regulares.

04

Selecionando com Condições (List Comprehension)

Selecionando com Condições (List Comprehension)



Nem sempre basta pegar um item pelo índice — muitas vezes queremos filtrar elementos de acordo com uma condição. A forma mais elegante de fazer isso é usando list comprehension.

```
Hashira do Python - João Vitor.py

idades = [12, 18, 25, 30, 15, 40]

# Selecionando apenas maiores de idade
maiores = [idade for idade in idades if idade >= 18]
print(maiores) # [18, 25, 30, 40]
```



Quando usar:

Quando você precisa criar uma nova lista filtrada de maneira rápida e clara, sem escrever muitos loops.

05

SELECIONANDO COM FILTER

Selecionando com filter()



Outra forma de filtrar dados é com a função `filter()`, que aplica uma condição a cada item da sequência. Diferente da list comprehension, o `filter()` retorna um objeto iterável, que normalmente é convertido em lista.

```
Hashira do Python - João Vitor.py

produtos = ["caderno", "caneta", "lapis", "celular"]

# Selecionando apenas itens com mais de 6 letras
resultado = filter(lambda p: len(p) > 6, produtos)
print(list(resultado)) # ['caderno', 'celular']
```



Quando usar:

Quando você prefere uma abordagem mais funcional, deixando claro que está filtrando uma lista de acordo com uma regra.

06

SELECIONANDO EM DICIONÁRIOS

Selecionando em Dicionários



Em dicionários, os itens são acessados pelas chaves. Essa é a forma mais natural de selecionar informações que têm um rótulo associado, como em um registro de aluno ou ficha de produto.

```
Hashira do Python - João Vitor.py

aluno = {
    "nome": "Carlos",
    "idade": 20,
    "curso": "Python"
}

print(aluno["nome"])    # Carlos
print(aluno.get("curso")) # Python
```



Quando usar:

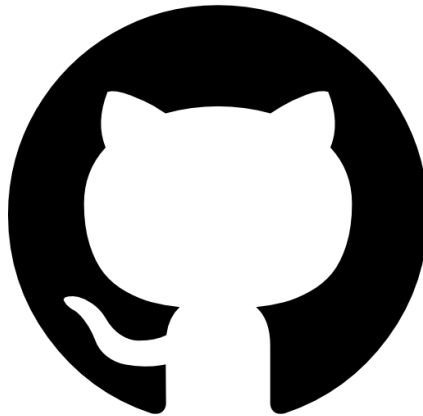
- Quando você precisa acessar valores por nome, e não por posição.
- Use `get()` quando não tiver certeza se a chave existe, pois evita erros e pode até definir um valor padrão.

Agradecimentos


Encerramento e um obrigado!



Esse Ebook foi gerado por IA, e diagramado por um humano.







<https://github.com/joaovitorsrp/prompts-recipe-to-create-a-ebook-curse>

 **Autor**

João Vitor

[GitHub](#) | [LinkedIn](#) | [Instagram](#) |

 com  por [João Vitor](#)

 com  por [João Vitor](#)

