

Paradigmas de programação

↳ forma de agrupar estratégias de programação com características semelhantes

* Imperativo ↳ explicita
↳ loops

```
for i in lista:
    print(math.exp(i))
```

for, while, do while

Funcional

map(math.exp, lista)
map, apply ↳ pandas
 R: apply.

↳ implícita

Matricial/vetorizado

pandas
numpy.exp(lista)

↳ R, matlab, julia
 numpy e pandas do python

group by
split-apply-combine

fácil de entender ↳ **clariza**

saber exatamente o que está sendo feito ↳ **transparência**

facilidade ↳ vai depender da experiência do programador

tarefas complexas ↔ **tarefas simples**

eficiência →

flexibilidade ↔

Popularidade ↔

Paradigma Imperativo

condições avaliadas no início

while → enquanto

mas sabemos previamente quantas vezes repetir

antes de entrar no loop

↳ loop's

Faca enquanto

do while

↳ condições avaliadas no final

↳ para toda sequência

for

↳ sabemos quantas vezes deve ser repetido.

iterador
1, 2, 3, 4

```
1 i = 1 # contador
2 N = 4 # quantidade de vezes (numero)
3 while i <= N:
4     print(i, ": Bom dia Python!")
5     i += 1 # incremento: i = i + 1
```

```
1 i = 1 # contador
2 N = 4 # quantidade de vezes
3 while True:
4     print(i, ": Bom dia Python!")
5     i += 1
6     if i > N: break
```

```
1 seq = range(1, 5)
2 for i in seq:
3     print(i, ": Bom dia Python!")
4
```

i = 1
N = 4
1 ≤ 4 V
Bom dia 1
i = 2
2 ≤ 4 V
Bom dia 2
i = 3
3 ≤ 4 V
Bom dia 3
i = 4
4 ≤ 4 V
Bom dia 4
i = 5
5 ≤ 4 F

i = 1
N = 4
V
Bom dia 1
i = i + 1 = 2
2 > 4 F
V
Bom dia 2
i = 3
3 > 4 F
V
Bom dia 3
i = 4
4 > 4 F
V
Bom dia 4
i = 5
5 > 4 V
break

i = 1
Bom dia 1
i = 2
Bom dia 2
i = 3
Bom dia 3
i = 4
Bom dia 4