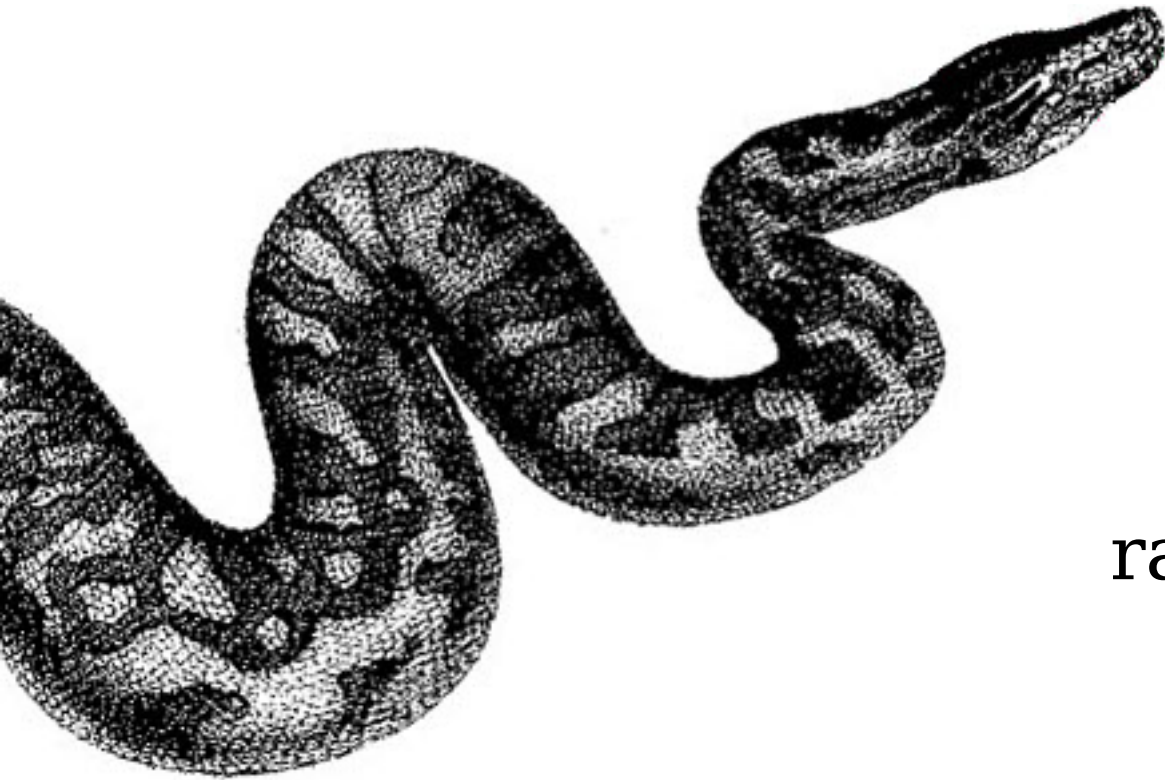


Python: sintaxe básica



Luciano Ramalho
ramalho@python.pro.br

Em vez de *Hello World...*

```
from datetime import datetime
from time import sleep

while True:      # rodar para sempre
    hora = datetime.now()
    print hora.strftime('%H:%M:%S')
    sleep(1)      # aguardar 1 segundo
```

Blocos por endentação

dois-pontos marca
o início do bloco

indentação dentro
do bloco deve ser
constante*

```
for i in range(1,11):  
    j = i*i  
    print i, j  
print 'FIM'
```

retorno ao nível anterior de
indentação marca o final do bloco

* por convenção, usa-se 4 espaços por nível (mas basta ser consistente)

Blocos

Todos os comandos que aceitam blocos:

if/elif/else	try/except
for/else	try/finally
while/else	class
	def

Se o bloco tem apenas um comando, pode-se escrever tudo em uma linha:

```
if n < 0: print 'Valor inválido'
```

Comentários

O símbolo **#** indica que o texto partir daquele ponto e até o final da linha deve ser ignorado pelo interpretador **python** exceto quando **#** aparece em uma string

Para comentários de várias linhas, usa-se três aspas simples ou duplas (isso cria uma “doc string” e não é ignorada pelo **python**, mas é usada para documentar)

```
""" Minha terra tem palmeiras,  
    Onde canta o Sabiá;  
    As aves, que aqui gorjeiam,  
    Não gorjeiam como lá. """
```

Palavras reservadas

and	elif	global	or
assert	else	if	pass
break	except	import	print
class	exec	in	raise
continue	finally	is	return
def	for	lambda	try
del	from	not	while
			yield

Execução condicional

Forma simples

```
if cond: comando
```

Forma em bloco

```
if cond:  
    comando1  
    comando2
```

Alternativas

```
if cond1: comando1  
    elif cond2: comando 2  
    else: comando 3
```

Repetições: comando for

Para percorrer sequências previamente conhecidas

```
for item in lista:  
    print item
```

Se for necessário um índice numérico:

```
for idx, item in enumerate(lista):  
    print idx, item
```

Para percorrer uma PA de 0 a 99:

```
for i in range(100):  
    print i
```


Repetições: comando while

Para repetir enquanto uma condição é verdadeira

```
""" Série de Fibonacci
    até 1.000.000
"""
a = b = 1
while a < 10**6:
    print a
    a, b = b, a + b
```

Controle de repetições

Para iniciar imediatamente a próxima volta do loop, use o comando **continue**

```
""" Ignorar linhas em branco
"""

soma = 0
for linha in file('vendas.txt'):
    if not linha.strip():
        continue
    codigo, qtd, valor = linha.split()
    soma += qtd * valor
print soma
```

Controle de repetições (2)

Para encerrar imediatamente o loop, use o comando **break**

```
total=0
while True:
    p = raw_input('+')
    if not p.strip(): break
    total += float(p)
print '-----'
print total
```

Tratamento de exceções

Comando **try/except**

```
total=0
while True:
    p = raw_input('+')
    if not p.strip(): break
    try:
        total += float(p)
    except ValueError:
        break
print '-----'
print total
```