Loops aninhados em Python

Em Python, quando você escreve um ou mais loops dentro de uma instrução de loop que é conhecida como **loop aninhado** . O loop principal é considerado como loop externo e os loops dentro do loop externo são conhecidos como loops internos.

A linguagem de programação Python permite usar um loop dentro de outro loop. A seção a seguir mostra alguns exemplos para ilustrar o conceito de loop for aninhado e loop while.

Python aninhado para loop

O loop for com um ou mais loops for internos é um loop for aninhado.

Python aninhado para sintaxe de loop

A sintaxe para uma instrução de loop foor aninhado Python na linguagem de programação Python é a seguinte -

```
for iterating_var in sequence:
   for iterating_var in sequence:
      statements(s)
   statements(s)
```

Exemplo de loop for aninhado em Python

O programa a seguir usa um loop for aninhado para encontrar os números primos de 2 a 100 -

```
#!/usr/bin/python

months = ["jan", "feb", "mar"]
days = ["sun", "mon", "tue"]


for x in months:
  for y in days:
    print(x, y)

print "Good bye!"
```

Quando o código acima é executado, ele produz o seguinte resultado -

```
('jan', 'sun')
('jan', 'mon')
('jan', 'tue')
('feb', 'sun')
('feb', 'mon')
('feb', 'tue')
('mar', 'sun')
('mar', 'mon')
('mar', 'tue')
Good bye!
```

Python aninhado enquanto loop

O loop while com um ou mais loops while internos é aninhado no loop while.

Python aninhado enquanto sintaxe de loop

A sintaxe para uma instrução **de loop while aninhada** na linguagem de programação Python é a seguinte -

```
while expression:
   while expression:
      statement(s)
   statement(s)
```

Uma observação final sobre o aninhamento de loop é que você pode colocar qualquer tipo de loop dentro de qualquer outro tipo de loop. Por exemplo, um loop for pode estar dentro de um loop while ou vice-versa.

Exemplo de loop while aninhado em Python

O programa a seguir usa um loop for aninhado para encontrar os números primos de 2 a 100 -

```
#!/usr/bin/python

i = 2
while(i < 100):
   j = 2
   while(j <= (i/j)):
      if not(i%j): break
      j = j + 1
   if (j > i/j) : print i, " is prime"
```

```
   i = i + 1

print "Good bye!"
```

Quando o código acima é executado, ele produz o seguinte resultado -

```
2 is prime
3 is prime
5 is prime
7 is prime
11 is prime
13 is prime
17 is prime
19 is prime
23 is prime
29 is prime
31 is prime
37 is prime
41 is prime
43 is prime
47 is prime
53 is prime
59 is prime
61 is prime
67 is prime
71 is prime
73 is prime
79 is prime
83 is prime
89 is prime
97 is prime
Good bye!
```