

## Estruturas de repetição

Estruturas ou laços de repetição são construções da linguagem que permitem repetir um bloco de código um número determinado ou indeterminado de vezes. Python possui duas construções, uma adequada para cada caso.

### For

O laço for é adequado para o primeiro caso, quando se sabe quantas vezes irá repetir um bloco de código.

Exemplo:

```
1 for i in range(1, 10):
2     print "i =", i
```

A função range chamada na definição do for, retorna uma lista contendo uma faixa de valores, no caso, de 1 a 10. A variável i é utilizada para iterar na lista retornada por range, a cada iteração i recebe um valor, primeiro 1, depois 2, e assim por diante até o 10. A função range recebe três argumentos – o primeiro é o início do intervalo, o segundo é fim, e o terceiro é o incremento. O incremento padrão é 1, bem como o início, portanto, chamar range(10), range(1, 10) ou range(1, 10, 1) resulta a mesma lista. Veremos mais sobre argumentos padrão ou default na parte de Funções, mais adiante na apostila. Qualquer objeto iterável pode ser utilizado em um for, como uma string, lista ou tupla. Veremos mais sobre iteradores no final da apostila, em Programação Funcional.

Exemplo:

```
1 for s in "Python":
2     print s
```

No exemplo anterior os caracteres da string "Python" são impressos um por linha com a função print.

### While

O laço while é utilizado quando não sabemos exatamente qual o número de repetições será realizada.

Exemplos:

```
1 i = 0
2 while i < 10:
3     print i
4     i = i + 1
```

Linha 1: Variável i utilizada no while é inicializada com 0;

Linha 2: while testa se i é menor do que 10;

Linha 3: Caso o teste da linha 2 seja verdadeiro, escreve o valor de i na tela;

Após a linha 4 ser executada o foco (cursor) volta para o while (linha 1), que testa i sucessivamente até que o valor não seja menor do que 10. Repare que o while testa uma expressão lógica, vista anteriormente na apostila. Caso esta expressão retorne sempre True, o while será repetido infinitamente, gerando um loop infinito.

## Controle de laços

Algumas vezes você vai enfrentar uma situação em que precisa sair de um laço completamente quando uma condição externa é acionada ou quando quer “pular” uma parte do laço e iterar novamente. Python fornece dois comandos para essas situações, o break e o continue.

### Break

Pode ser utilizado tanto em um laço for quanto while. O comando break, assim como na linguagem C e derivadas, “quebra” um laço.

Exemplo:

```
1 i = 0
2 while i < 10:
3     if i == 5:
4         break
5     else:
6         print i
7         i += 1
```

No exemplo anterior, quando i é igual a 5 o laço while termina, assim, só os números 1, 2, 3 e 4 são impressos na tela.

## Continue

Também pode ser utilizado tanto no laço for quanto while. O continue faz com que controle passe para o início do laço.

Exemplo:

```
1  for letra in 'Python':
2      if letra == 'h':
3          continue
4      print letra
```

Nesse caso, as letras da string 'Python' são impressas, exceto pela letra 'h'.

## Pass

O pass, na verdade, não é um comando de controle de laços (você deve ter percebido, já que dissemos que Python possui dois comandos de controle de laços e não três). O pass não faz exatamente nada, e é utilizado quando uma expressão é sintaticamente necessária, mas o programa não precisa de nenhuma ação.

Exemplos:

```
1  def teste(*args):
2      pass                # Lembrar de implementar isso depois!!!
3
4  class ClasseVazia:
5      pass
6
7  while True:
8      pass                # Aguarda interrupção por teclado (Ctrl+C)
```

O exemplo da função teste mostra que o pass pode ser utilizado como um placeholder, quando se está trabalhando em um novo código, permitindo assim continuar pensando em um nível mais abstrato e implementar a função depois.

O segundo exemplo demonstra a definição de uma classe vazia. Classes serão explicadas no final da apostila, em Programação Orientada a Objetos

O terceiro exemplo é um while infinito que é executado até que o programa seja interrompido por um Ctrl + C.